Department of Electronics and Communications Engineering

PERFORMANCE ANALYSIS OF TWO PROMINENT WIRELESS ADHOC ROUTING PROTOCOLS USING NS-2 SIMULATION

Prepared By

Shamanta Ismit

ID: 2012-2-55-049

Mehnaj Tabassum Ahmed

ID: 2012-2-55-053

Dept. of ECE

East West University

Supervised By

Md. Asif Hossain

Assistant Professor, Dept. of ECE

December 2016

Letter of Transmittal

To
Md. Asif Hossain
Assistant Professor
Department of Electronics and Communication Engineering
East West University

Subject: Submission of Project Report as (ETE-498)

Dear Sir,

We are pleased to let you know that we have completed our thesis on **"Performance Analysis of Two Prominent Wireless Adhoc Routing Protocols Using Ns-2 Simulation"**. The attachment contain of the thesis that has been prepared for your evaluation and consideration. Working on this thesis has given us some new concepts. By applying those concept we have tried to make something innovative by using our theoretical knowledge which we have acquired since last four years from you and the other honorable faculty members of EWU. This thesis would be a great help for us in future.

We are very grateful to you for your guidance, which helped us a lot to complete my project and acquire practical knowledge.

Thanking You.

Yours Sincerely

Shamanta Ismit

ID: 2012-2-55-049

Mehnaj Tabassum Ahmed

ID: 2012-2-55-053

Dept. of ECE
East West University

Declaration

This is certified that the project is done by us under the course “Thesis (ETE-498)”. The thesis of **“Performance Analysis of Two Prominent Wireless Adhoc Routing Protocols Using Ns-2 Simulation”** has not been submitted elsewhere for the requirement of any degree or any other purpose except for publication.

Shamanta Ismit

ID: 2012-2-55-049

Mehnaj Tabassum Ahmed

ID: 2012-2-55-053

Acceptance

This thesis paper is submitted to the **Department of Electronics and Communications Engineering, East West University** is submitted in partial fulfillment of the requirements for the degree of **B.Sc.** in **Electronics & Telecommunications Engineering** under complete supervision of the undersigned.

Md. Asif Hossain
Assistant Professor
Dept. of ECE
East West University

Abstract

As a new generation of wireless communication system, mobile ad-hoc network (MANET) has developed greatly during the last decade. Endowed by great mobility, dynamic topology, self-organizing and other unique features, it is widely used in military action, on-demand operations, in any disaster relief efforts. However, the fact that MANET has no central infrastructure, hence the devices can move randomly gives rise to various kind of difficulties, such as routing and security. Without a doubt, we will soon be able to observe ad-hoc network deployment everywhere in the near future. In this thesis the problem of routing is considered. This thesis addresses issues pertaining to two prominent routing protocols Destination Sequenced Distance vector (DSDV) and Dynamic Source Routing (DSR) protocols. NS2 simulation and ViSim simulation have been used to analysis the performances of these two protocols.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Mobile adhoc network (MANETs) are networks where nodes are communicating wirelessly with each other without any existing infrastructure. MANETs have the advantage of fast deployment, robustness, flexibility and intrinsic support for mobility. This flexibility of self-configuring and self- management makes it worthwhile for various applications in military operations, wireless mesh networks; wireless sensor networks etc. Due to the wireless nature of MANET, the routing protocol is a very important issue to make it more efficient and reliable. Routing protocols for MANET are classified as two main classes: on-demand routing protocols and Proactive or table Driven routing protocols.

Many paper present comparisons between the routing protocols like DSR, AODV and DSDV based on Pdf, Average End to End Delay and Routing Load [1, 2]. This project aims to provide a step by step comparative analysis of 2 popular routing protocols: DSR and DSDV.

The rest of the project paper is organized as follows: Section 2 presents an overview of the Wireless routing protocol that is analyzed and compared. Section 3 gives a brief description of the Simulation parameters, assumptions hold and description of the step by step comparing methodology used in the paper. Section 4 provides the simulation results and discusses it. Finally the conclusion is provided in section 5.

1.2 PROJECT DESCRIPTION

The ad hoc routing protocols DSDV and DSR are two of the promising routing protocols. They can be used in mobile ad hoc networks to route packets between mobile nodes. The main objectives of this thesis project are:

(1) Implementing the existing DSDV and DSR routing protocols by using ViSim1.0

(2) Comparing the performance of two protocols under following metrics

- (i) throughput vs time
- (ii) Routing load
- (iii) Goodput vs time

CHAPTER 2

MOBILE AD HOC NETWORKING

This chapter gives an overview of Mobile Ad Hoc Networking. Section 2.1 introduces the protocol stacks used in the Internet and MANET and compares them with the Open Systems Interconnection (OSI) model. Section 2.2 and 2.3 describes the proactive and reactive gateway discovery. Then section 2.4 and 2.5 describes the different routing concepts of DSDV and DSR.

2.1 THE PROTOCOL STACK

In this section, the protocol stack for MANET has been described. This gives a comprehensive picture of, and helps to better understand, mobile ad hoc networks. Figure 2.1, shows the protocol stack consisting of five layers: physical layer, data link layer, network layer, transport layer and application layer. It has similarities to the TCP/IP protocol suite. In TCP/IP suit, the application layer is the combination of the session, presentation and application layer of the OSI model.

On the left of Figure 2.1, the OSI model is shown. OSI model is a theoretical layered outline for the design of network systems that allows for communication over all types of communication systems.

In the middle of the figure, the TCP/IP model has been demonstrated. The OSI model, the layers in the TCP/IP suite do not resemble exactly to the OSI layers. The lower four layers are the same but the fifth layer in the TCP/IP suite (the application layer) is equivalent to the combination of session, presentation and application layers of the OSI model.

On the right, the MANET protocol stack has been described. It is similar to the TCP/IP suite. The main difference between these two protocols stacks are in the network layer. Mobile nodes (which are both hosts and routers) use an ad hoc routing protocol to route the packets. Mobile nodes run protocols that have been designed for wireless channels are in the physical and data link layer. In simulation tool used in this project, the standard IEEE 802.11 has been used in these layers. [7]

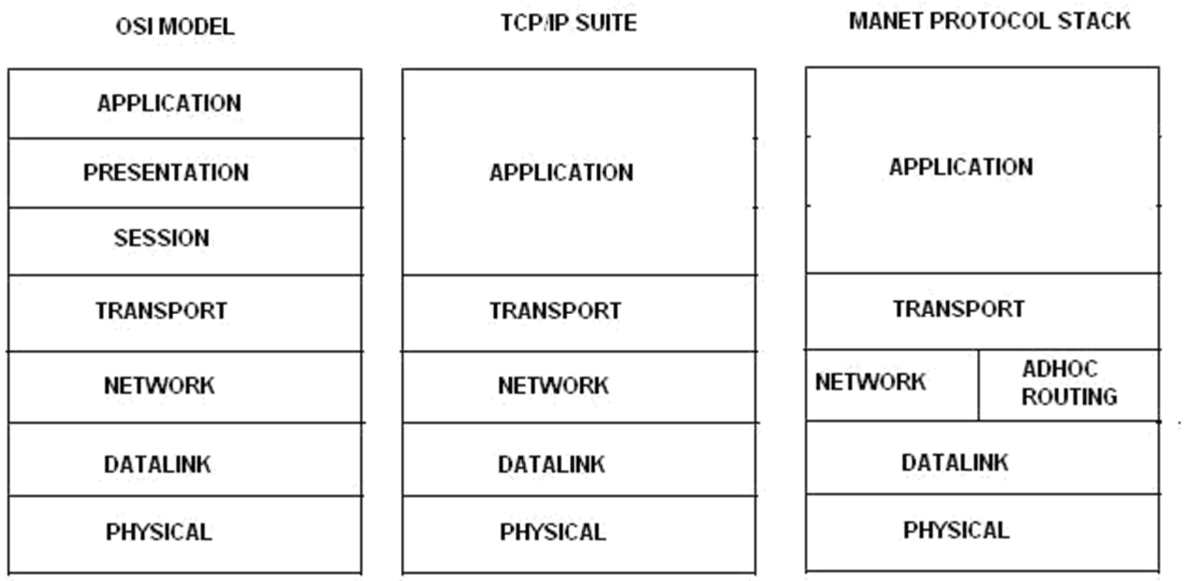

Figure 2.1: Three models

This thesis focuses on ad hoc routing which is handled by the network layer. The network layer is divided into two parts: Network and Ad Hoc Routing. The protocol used in the network part is Internet Protocol (IP) and the protocols which can be used in the ad hoc routing part are Destination Sequenced Distance Vector (DSDV), or Dynamic Source Routing (DSR), which are described in section 2.4.

MANET ROUTING PROTOCOLS:

There are 3 main categories of MANET routing protocols. They are given below:

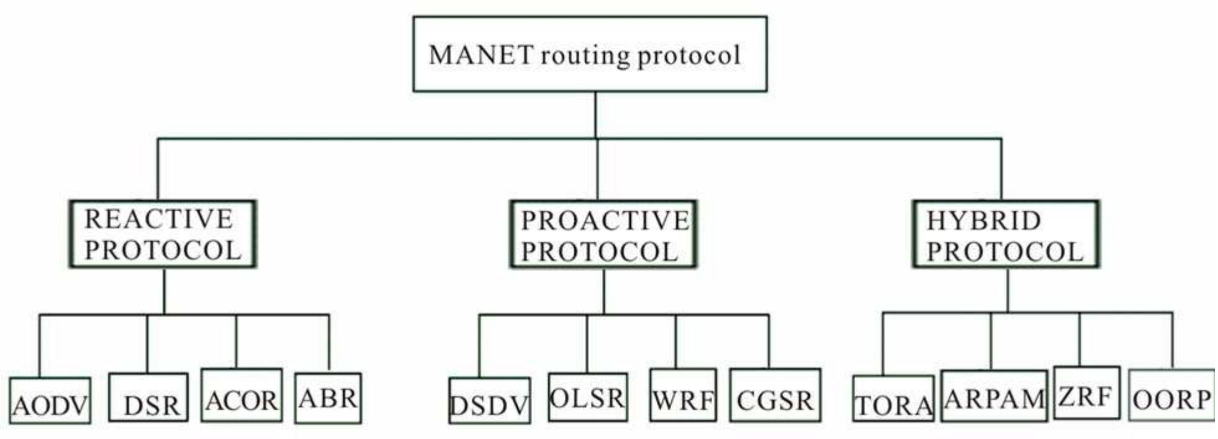

Fig. 2.1.1: Classification of MANET Routing Protocol

2.2 PROACTIVE GATEWAY DISCOVERY

All the proactive algorithms are based on traditional distance vector and link state protocols developed for use in wireless network. The primary characteristic of proactive approach is that each node in the maintenance of network is to maintain a route to every other node in the network all the times regardless of whether or not these routes are needed. In order to maintain correct routing information, a node must periodically send control messages. Updates to route table are triggered or by certain events which caused in manipulation of other nodes (neighboring) route table. Link addition and removal can trigger an event triggered updating of routing table. In proactive approach the main advantage is that the rout to each node is instantly found because the table contains all the nodal address. Source only need to check the routing table and transfer a packet. The major disadvantage of proactive approach is that each node is prone to rapid movement. So the overhead of maintaining a rout table is very high, and amount of routing state maintained at each node scales as order of o[n] where n is the number of nodes in the network. It becomes inefficient for a large network.

DSDV, OLSR, GSR etc are the examples of proactive routing protocol.

2.3 REACTIVE GATEWAY DISCOVERY

Reactive routing technique is also known as on-demand routing. It takes a different approach of routing which overcomes the disadvantages of proactive routing. In reactive approaches those nodes which require connectivity to the Internet reactively find Internet gateways by means of broadcasting some kind of solicitation within the entire ad hoc network. This approach reduces the overhead of maintaining the route table as that of proactive. The node dynamically checks the route table, and if it does not find an entry for its destination or it finds an outdated entry it performs route discovery to find the path to its destination. [5]

The signaling overhead is reduced in this method, particularly in networks with low to moderate traffic loads. However it has a drawback of route acquisition latency. That is when corresponding entry is not found the route discovery mechanism occurs which takes a very large amount of time, and for that time the packet waits for updation of the table.

AODV, DSR etc are the examples of reactive routing protocol.

2.4 DSDV

This protocol is based on classical Bellman-Ford routing algorithm designed for MANETS. It is an example of proactive routing protocols. Each node maintains a list of all destinations and number of hops to each destination. Each entry is marked with a sequence number. It uses full dump or incremental update to reduce network traffic generated by route updates. The broadcast of route updates is delayed by settling time. The only improvement made here is avoidance of routing loops in a mobile network of routers. With this improvement, routing information can always be readily available, regardless of whether the source node requires the information or not. DSDV solve the problem of routing loops and count to infinity by associating each route entry with a sequence number indicating its freshness. In DSDV, a sequence number is linked to a destination node, and usually is originated by that node (the owner). The only case that a non-owner node updates a sequence number of a route is when it detects a link break on that route. An owner node always uses even-numbers as sequence numbers, and a non-owner node always uses odd-numbers. With the addition of sequence numbers, routes for the same destination are selected based on the following rules: 1) a route with a newer sequence number is preferred; 2) in the case that two routes have a same sequence number, the one with a better cost metric is preferred. [4]

The list which is maintained is called routing table. The routing table contains the following:

(1) All available destinations' IP address

(2) Next hop IP address

(3) Number of hops to reach the destination

(4) Sequence number assigned by the destination node

(5) Install time

The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven.

As stated above one of "full dump" or an incremental update is used to send routing table updates for reducing network traffic. A full dump sends the full routing table to the neighbors and could span many packets whereas in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence number has changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental packets can grow big so full dumps will be more frequent. [4]

Each route update packet, in addition to the routing table information, also contains a unique sequence number assigned by the transmitter. The route labeled with the highest (i.e. most recent) sequence number is used. If two routes have the same sequence number then the route with the best metric (i.e. shortest route) is used. Based on the past history, the stations estimate the settling time of routes. The stations delay the transmission of a routing update by settling time so as to eliminate those updates that would occur if a better route were found very soon.

Each row of the update send is of the following form:

<Destination IP address, Destination sequence number, Hop count>

After receiving an update neighboring nodes utilizes it to compute the routing table entries.

To damp the routing fluctuations due to unsynchronized nature of periodic updates, routing updates for a given destination can propagate along different paths at different rates. To prevent a node from announcing a routing path change for a given destination while another better update for that destination is still in route, DSDV 11 requires node to wait a settling time before announcing a new route with higher metric for a destination.

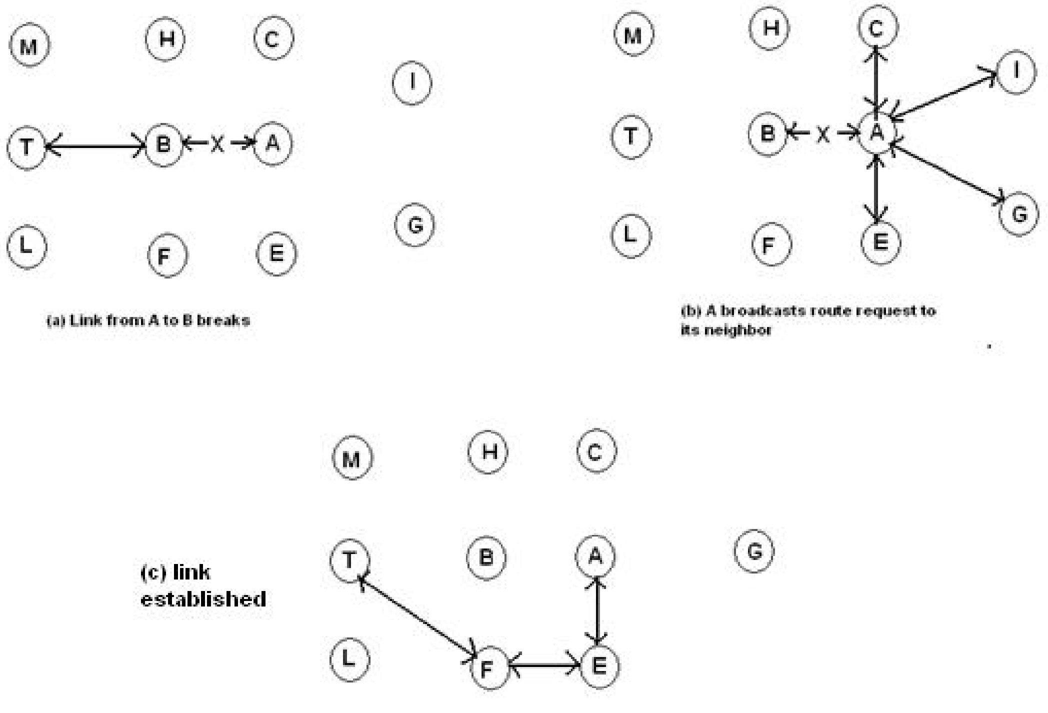

Fig. 2.2 Resolving failed links in DSDV

2.5 DSR

The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. Dynamic Source Routing, DSR, is a reactive routing protocol that uses source routing to send packets. It uses source routing which means that the source must know the complete hop sequence to the destination.

Each node maintains a route cache, where all routes it knows are stored. The route discovery process is initiated only if the desired route cannot be found in the route cache.

To limit the number of route requests propagated, a node processes the route request message only if it has not already received the message and its address is not present in the route record of the message.

As mentioned before, DSR uses source routing, i.e. the source determines the complete sequence of hops that each packet should traverse. This requires that the sequence of hops is included in each packet's header. A negative consequence of this is the routing overhead every packet has to carry. However, one big advantage is that intermediate nodes can learn routes from the source routes in the packets they receive. Since finding a route is generally a costly operation in terms of time, bandwidth and energy, this is a strong argument for using source routing. Another advantage of source routing is that it avoids the need for up-to-date routing information in the intermediate nodes through which the packets are forwarded since all necessary routing information is included in the packets. Finally, it avoids routing loops easily because the complete route is determined by a single node instead of making the decision hop-by-hop. [5][6]

The protocol is composed of the two main mechanisms of "Route Discovery" and "Route Maintenance", which work together to allow nodes to discover and maintain routes to arbitrary destinations in the ad hoc network. All aspects of the protocol operate entirely on demand, allowing the routing packet overhead of DSR to scale automatically to only what is needed to react to changes in the routes currently in use. The protocol allows multiple routes to any destination and allows each sender to select and control the routes used in routing its packets, for example, for use in load balancing or for increased robustness.

Route Discovery

Route Discovery is used whenever a source node desires a route to a destination node. First, the source node looks up its route cache to determine if it already contains a route to the destination.

If the source finds a valid route to the destination, it uses this route to send its data packets. If the node does not have a valid route to the destination, it initiates the route discovery process by broadcasting a route request message. The route request message contains the address of the source and the destination, and a unique identification number.

An intermediate node that receives a route request message searches its route cache for a route to the destination. If no route is found, it appends its address to the route record of the message and forwards the message to its neighbors. The message propagates through the network until it reaches either the destination or an intermediate node with a route to the destination. Then a route reply message, containing the proper hop sequence for reaching the destination, is generated and unicast back to the source node. [6]

Route maintenance

Route Maintenance is used to handle route breaks. When a node encounters a fatal transmission problem at its data link layer, it removes the route from its route cache and generates a route error message. The route error message is sent to each node that has sent a packet routed over the broken link. When a node receives a route error message, it removes the hop in error from its route cache.

Acknowledgment messages are used to verify the correct operation of the route links. In wireless networks acknowledgments are often provided as e.g. an existing standard part of the MAC protocol in use, such as the link-layer acknowledgment frame defined by IEEE 802.11. If a built-in acknowledgment mechanism is not available, the node transmitting the message can explicitly request a DSR-specific software acknowledgment to be returned by the next node along the route. [6]

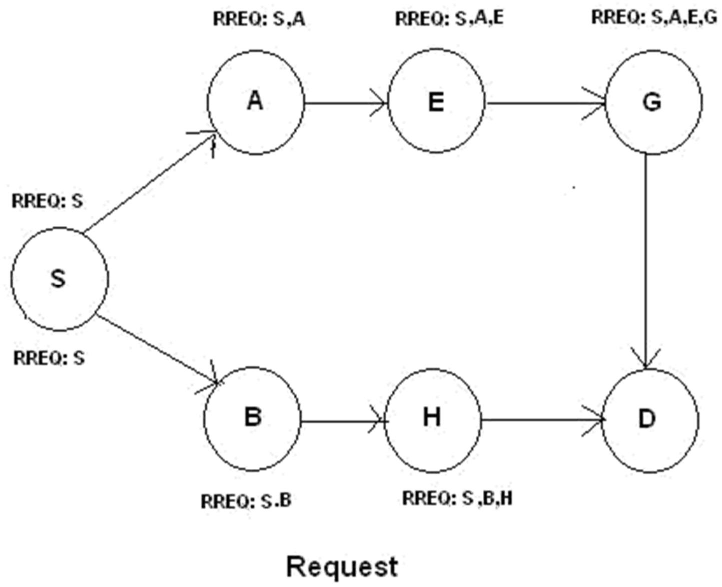

Fig. 2.3 DSR request

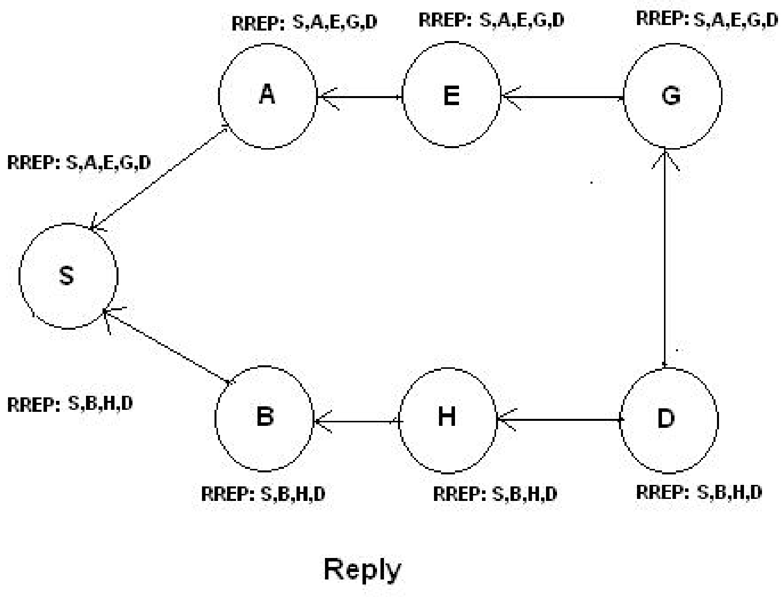

Fig. 2.4 DSR Reply

CHAPTER 3

NETWORK SIMULATOR

DSDV and DSR routing protocols can be implemented using Network Simulator 2.31. NS is a discrete event simulator targeted at networking research. It provides substantial support for TCP routing and multicast protocols over wired and wireless networks. We have used ViSim 1.0 for constructing the simulation and to create the graph.

3.1 ABOUT NS 2

NS is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. ns uses two languages because simulator has two different kinds of things it needs to do. On one hand, detailed simulations of protocols require a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important.

On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important.

ns meets both of these needs with two languages, C++ and OTcl .C++ is fast to run but slower to change, making it suitable for detailed protocol implementation. OTcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration.

In NS-2, the frontend of the program is written in TCL(Tool Command Language). The backend of NS-2 simulator is written in C++ and when the tcl program is compiled, a trace file and namfile are created which define the movement pattern of the nodes and keeps track of the number of packets sent, number of hops between 2 nodes, connection type etc at each instance of time. In addition to these, a scenario file defining the destination of mobile nodes along with their speeds and a connection pattern file (CBR file) defining the connection pattern, topology and packet type are 17 also used to create the trace files and nam files which are then used by the simulator to simulate the network. [1][2]

Also the network parameters can be explicitly mentioned during the creation of the scenario and connection-pattern files using the library functions of the simulator.

3.2 ViSim1.0 (Visual Simulation Tool version 1.0)

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 naive 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 associated .tcl files 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 a particular MANET scenario.

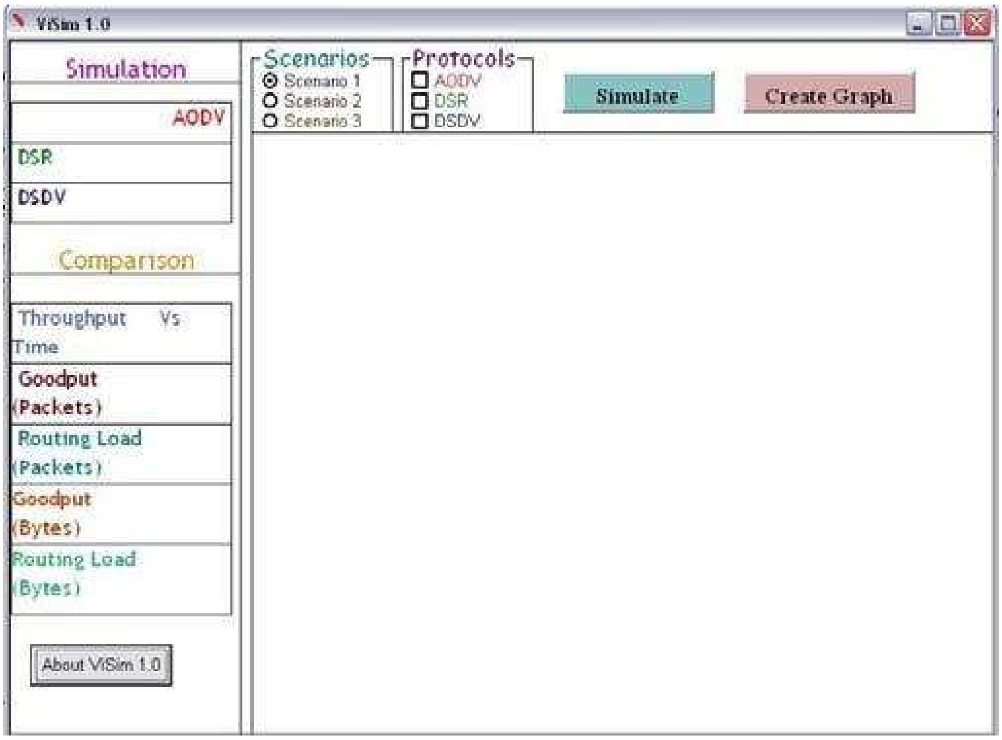

Fig. 3.1 Visual Simulation Tool Interface, ViSim 1.0

Different Working Areas

Figure 3.1 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 four 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.

(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.

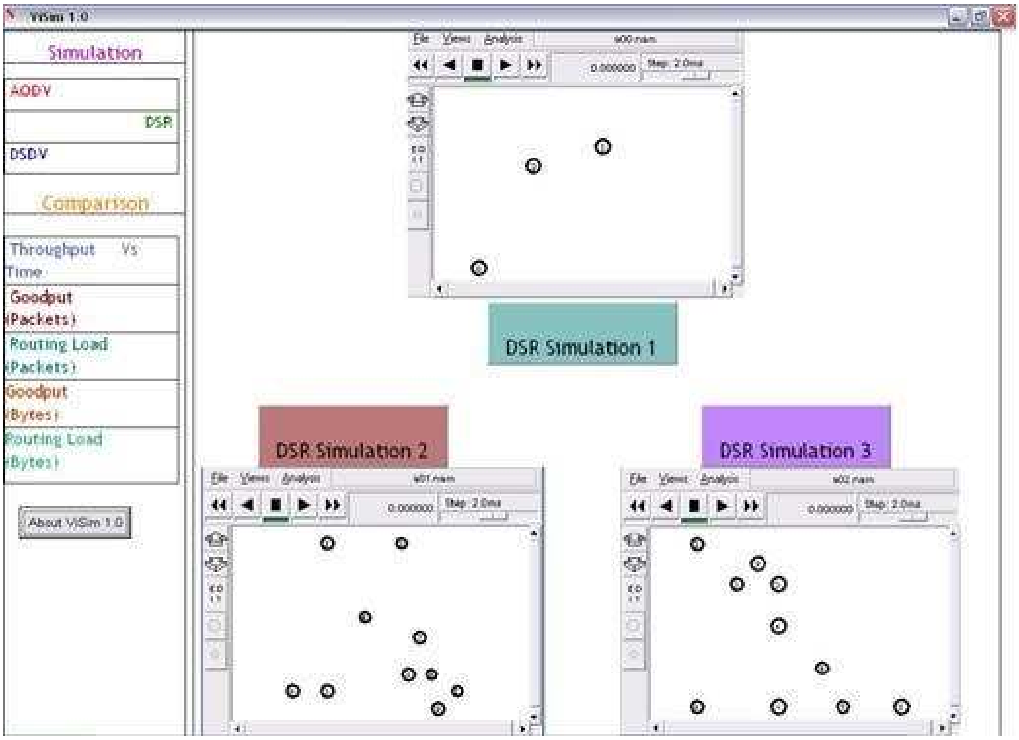

Fig. 3.2. DSR simulation options

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 3.2.

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 display for visual simulation [see Figure 3.3].

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.

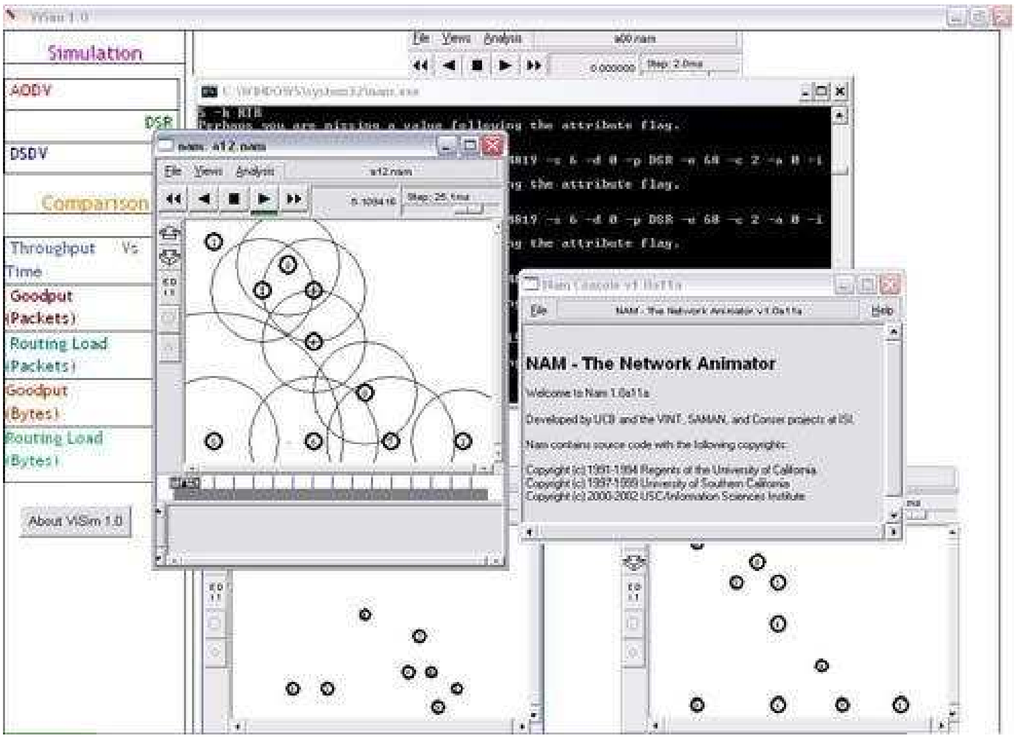

Fig 3.3. The output after choosing DSR Simulation 3

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 3.4 shows the output where we selected ‘Throughput Vs Time’.

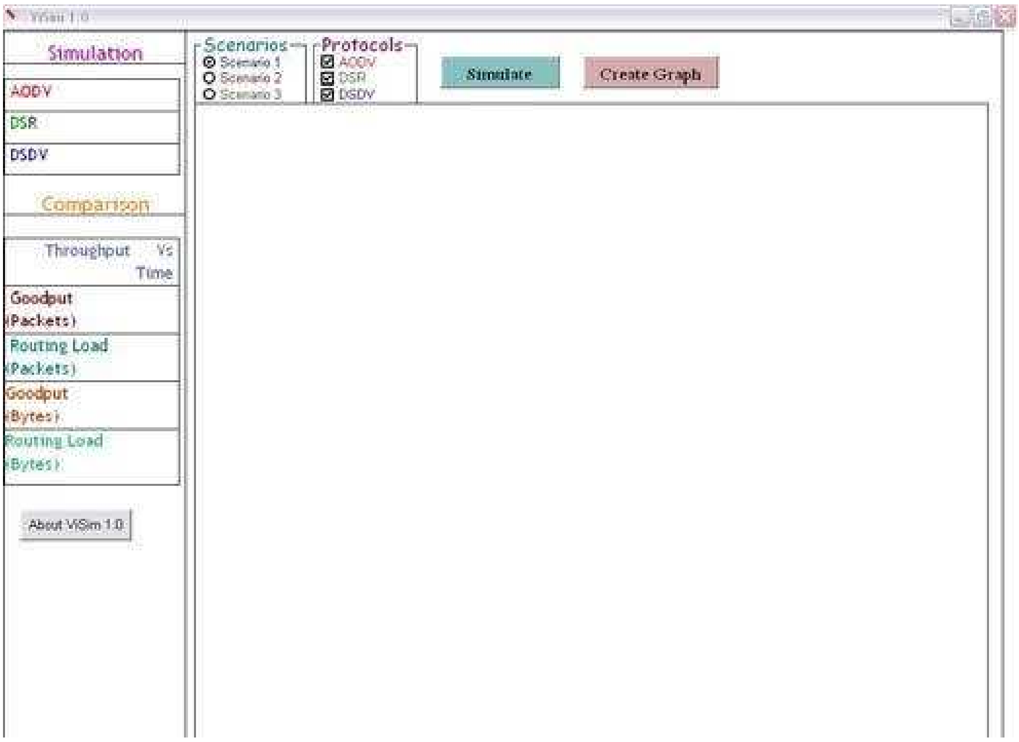

Fig. 3.4. An example where 'Throughput vs Time', Scenario 1, and three protocols are selected for running the simulations

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 3.5 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).

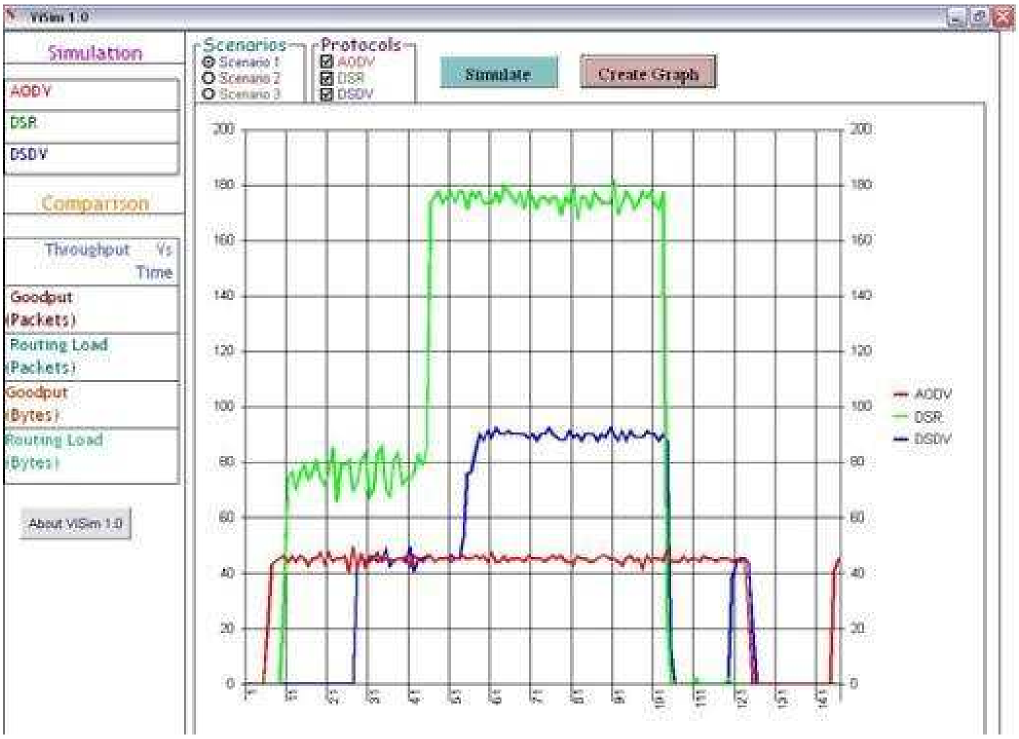

Fig. 3.5. A sample output graph (Throughput vs Time) using network scenario 1, all three protocols; DSDV, DSR, and AODV are compared

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.

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 Network, etc.) scenario.

CHAPTER 4

SIMULATION

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

4.1 Simulation Parameters and Specifications

Table 1 shows the specifications and parameters that we used for our experiments:

Table 1. Simulation Parameters

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
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

4.1 SIMULATION OF DSDV AND DSR

My aim here was to implement DSDV and DSR routing protocol for 10 nodes sending cbr packets with random speed. First the cbr files and scenario files are generated and then using dsdv protocol simulation is done which gives the nam file and trace file. Then another nam and Trace files are created dsr protocol. [2]

The following figures are the execution of the nam files instances created. For each execution of the same program different nam files are created and we can view the output on the network simulator.

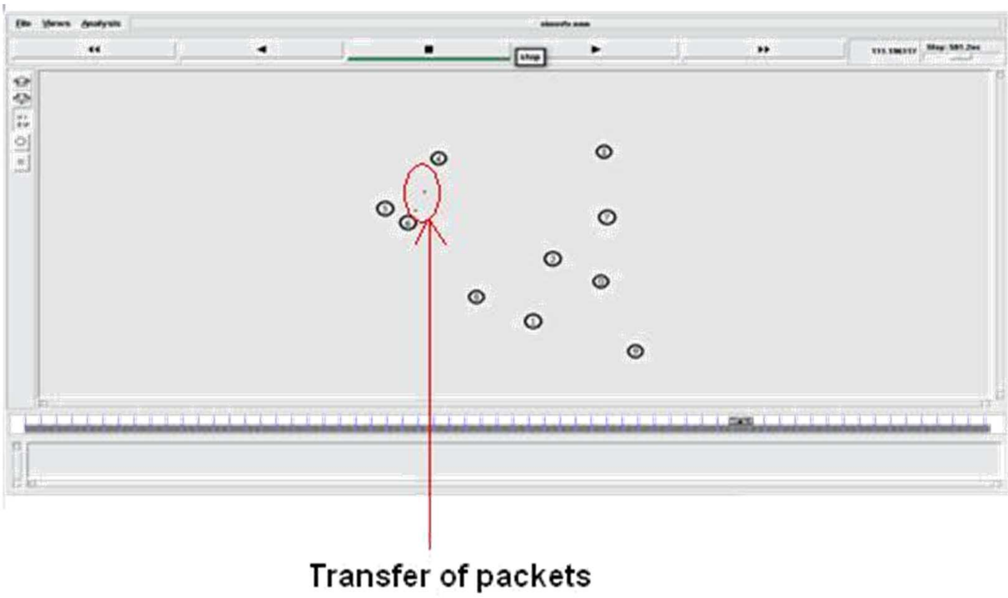

Fig 4.1 transfer of packet

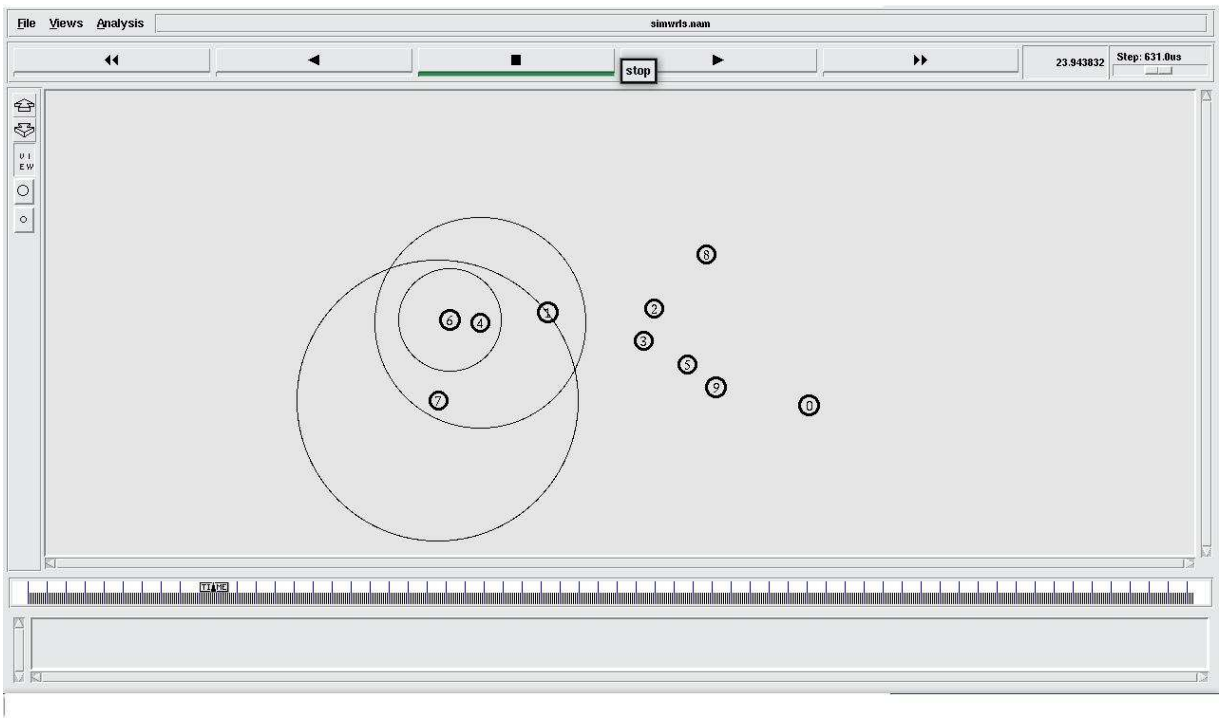

Fig 4.2 dropping of packets

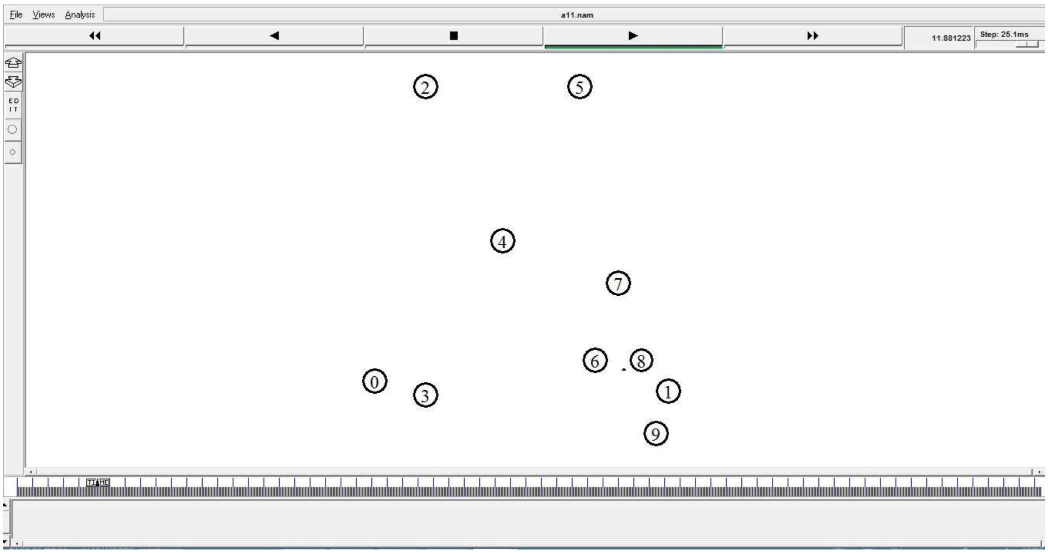

Fig 4.3 DSR Simulation

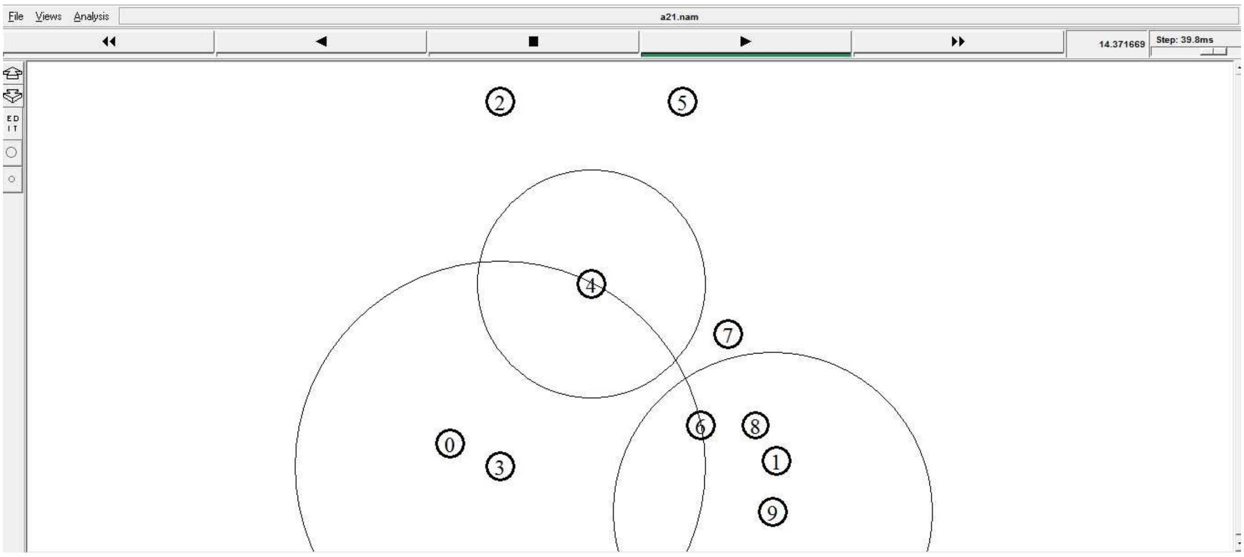

Fig 4.4 DSDV simulation

Movement Model

The mobile nodes move according to the random waypoint" model. Each mobile node begins the simulation by remaining stationary for pause time seconds. It then selects a random destination in the defined topology area and moves to that destination at a random speed. The random speed is distributed uniformly between zero (zero not included) and some maximum speed. Upon reaching the destination, the mobile node pauses again for pause time seconds, selects another destination, and proceeds there as previously described. This movement pattern is repeated for the duration of the simulation. [1]

Communication Model

In the scenario used in this study, five mobile nodes communicate with one of two fixed nodes (hosts) located on the Internet through a gateway. As the goal of the simulations was to compare the different approaches for gateway discovery, the 26 traffic source was chosen to be a constant bit rate (CBR) source. Each source mobile node generates packets every 0.2 seconds in this study. In other words, each source generates 5 packets per second. Since each packet contain 512 bytes of data, the amount of generated data is 5*512*8 bit/s = 20 kbit/s, for each source. The traffic connection pattern is generated by CMUs traffic generator (cbr-gen.tcl). The main parameters in cbrgen.tcl are \connections" (number of sources) and \rate" (packet rate); see Table 2 [1]

.

.

Table 2. Some parameters for the communication model

Parameter	Value
Transmission range	250 m
Simulation time	110 s
Topology size	800m x 500m
Number of mobile nodes	14
number of sources	4
Number of gateways	2
Traffic type	constant bit rate
Packet rate	5 packets/s
Packet size	512 bytes
Maximum speed	10 m/s

4.2 PERFORMANCE METRICS

The second goal of this project is to compare the performance of the two protocols under different scenario. Comparisons among different protocols were based on the aggregate of the performance metrics resulting from the simulations of 15 different scenarios that had been 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).

<u>Goodput:</u>

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.

<u>Routing Load:</u>

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.

CHAPTER 5

RESULTS

5.1 Throughput analysis

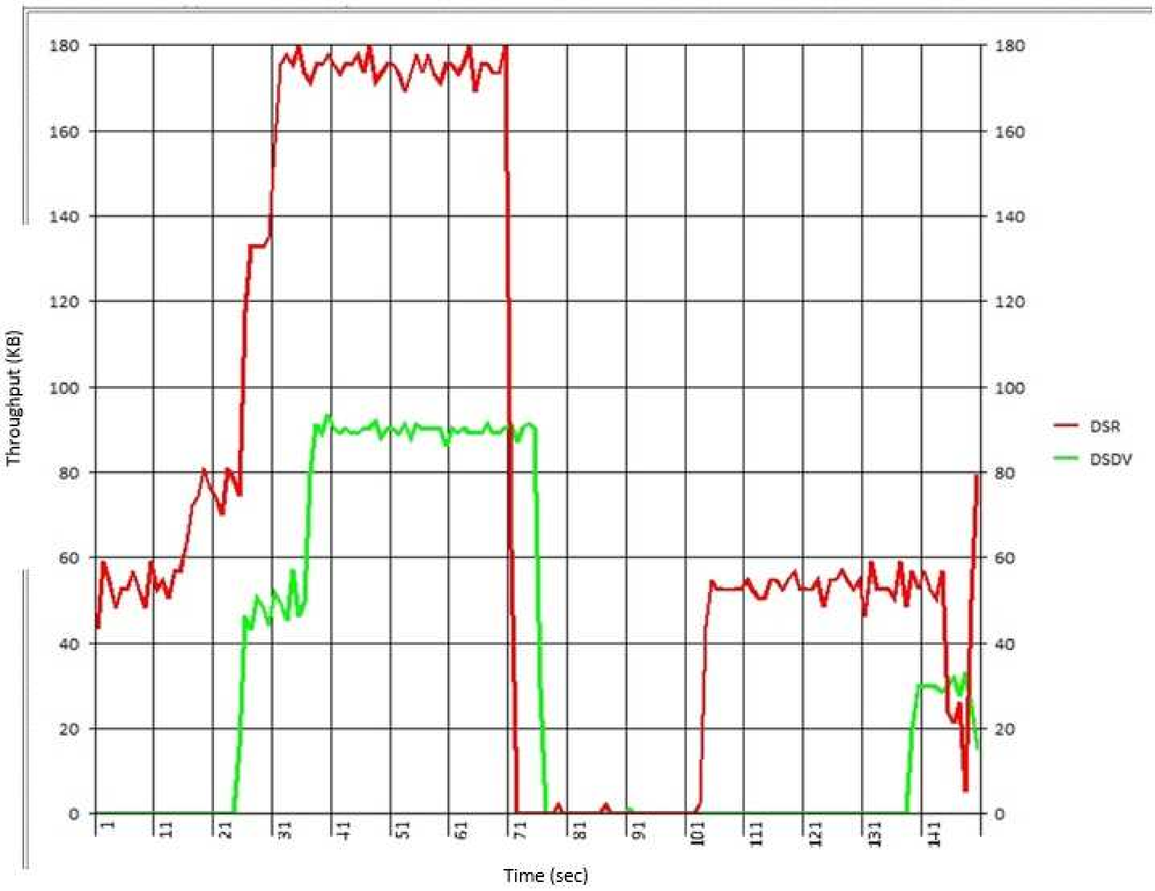

Fig 5.1 Throughput verses time

Figure 5.1 shows the aggregated result for 'Throughput vs Time' where we analyzed the total bytes received by the destination node per second (Data packets and overhead). Based on the results that we see here, the following comments could be made:

DSR starts off quickly however we can see that there are lots of fluctuations in the data rate while DSDV takes time to start off but the data rate has lesser fluctuations. DSR shows the greater throughput than the DSDV. DSR's throughput is almost double than that of the DSDV.

5.2 Goodput analysis

We calculated Goodput in terms of number of packets. Now, if we analyze the graph presented in Figure

5.2, we can see that on an average, if 100 packets are transmitted in the network, 18 packets would be data packets for DSR and 38 for DSDV. That means transmitted packets' 18% are data packet for DSR and rest of the 82% are overload (non-data packet) while in case of DSDV 38% are data packet and 62% are the overload. From these data, we could say that; though DSDV takes time to converge, it actually is sending more data packets in number than that of DSR.

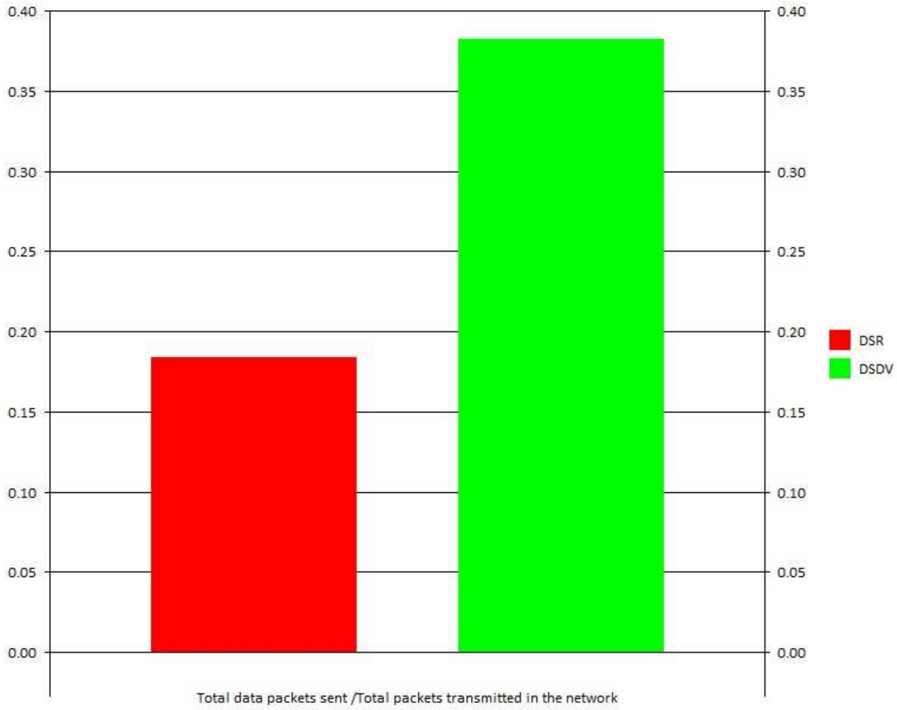

Fig 5.2 Goodput (packets)

5.3 Routing load analysis

We again calculated routing loads in terms of number of packets. The results are presented in Figure 5.3. Again we can see that; though DSR has a better throughput, it actually contains more overhead for routing packets. DSDV has a relatively lower routing load than DSR.

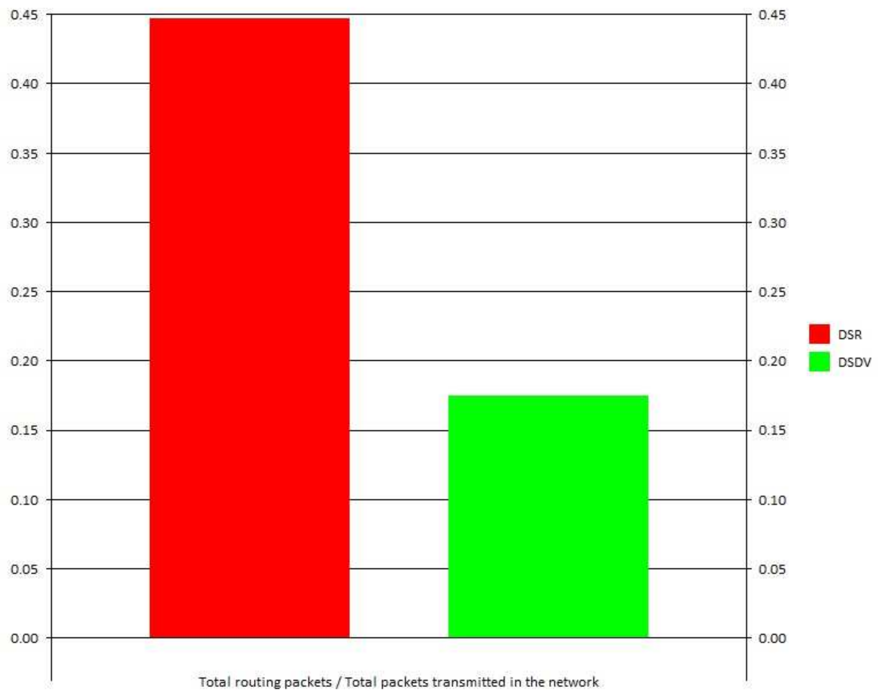

Fig. 5.3 Routing Load (packets)

CHAPTER 6

CONCLUSION

In this project, we have implemented the Destination Sequenced Distance Vector and Dynamic Source Routing protocols in Tool command language and integrated the module in the ns-2 Simulator. The performance of the protocols were measured with respect to metrics like throughput analysis, goodput analysis and routing load analysis. Simulations were carried out with identical topologies and running different protocols on the mobile node.

In our results, it has been found that DSDV turns out to have the best goodput and lesser routing load; however, it takes more time to converge. So if there is relatively less number of nodes in the network and the mobility is somewhat steady or slow, DSDV will work more efficiently. On the other hand, DSR has a very high throughput. But it actually contains less data packets and higher routing load and it has been shown that there are lots of fluctuations on the throughput curve which are not preferred in a wireless network. We can conclude that any of the single protocol does not supersede the other one. Their performances depend upon the different scenarios.

REFERENCES

[1] www.isi.edu/nsnam/ns/tutorial Marc Greis tutorial on ns2

[2] Matthias Transier “ Ns2 tutorial running simulations ”

[3] D. Kim, J. Garcia and K. Obraczka, “Routing Mechanisms for Mobile Ad Hoc Networks based on the Energy Drain Rate”, IEEE Transactions on Mobile Computing. Vol 2, no 2, 2003, pp.161-173

[4] C.E. Perkins & P. Bhagwat, “Highly Dynamic Destination Sequence-Vector Routing (DSDV) for Mobile Computers”, Computer Communication Review, vol. 24, no.4, 1994, pp. 234-244.

[5] C.E. Perkins and E.M. Royer, “Ad-Hoc on-Demand Distance Vector Routing,” Proc. Workshop Mobile Computing Systems and Applications (WMCSA ’99), Feb. 1999 pp. 90-100.

[6] David B. Johnson and David A. Maltz. “Dynamic source routing in ad hoc wireless networks”, Mobile Computing, Kluwer Academic Publishers. 1996 pp.153–181, 1996.

[7] M. S. Corson, J. P. Maker and G. H. Cirincione , "Internet-Based Mobile Ad Hoc Networking," IEEE Internet Computing, Vol. 3, no. 4, July-August 1999, pp. 63-70.

[8] Saquib, Nazmus, Md Sakib, Sabbir Rahman, and Al-Sakib Khan Pathan. "Performance Analysis of MANET Routing Protocols Using An Elegant Visual Simulation Tool." *arXiv preprint arXiv:0911.2620* (2009).