Global Mobile Information Simulator in Windows XP to Analysis the Performance of Routing Protocols in MANET

R. P. Mahapatra, Manish Bhardwaj, and Sachi pandey

***Abstract*—Mobile Ad hoc Network (MANET) is an autonomous system of mobile nodes connected by wireless link. It can operate without fixed infrastructure and can survive rapid changes in the network topology .They can be studied formally as graphs in which the set of parameters varies in time. The main method for evaluating the performance of MANET is simulation. This paper is subjected to how to install global mobile information system simulator in window xp and to evaluate the performance of dominating sets in Ad hoc on demand distance vector routing algorithm (AODV) that analysis packet delivery fraction and end to end delay with varying number of nodes.**

***Index Terms*—AODV, glomosim 2.03, MANET, parsec, visualization tool.**

I. Introduction

A Mobile Ad hoc Network (MANET) is a kind of wireless ad-hoc network and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links – the union of which forms an arbitrary topology. If there are only two nodes that want to communicate with each other and are located very closely to each other, then no specific routing protocols or routing decisions are necessary. On the other hand, if there are a number of mobile hosts wishing to communicate, then the routing protocols come into play because in this case, some critical decisions have to be made such as which is the optimal route from the source to the destination which is very important because often, the mobile nodes operate on some kind of battery power. Thus it becomes necessary to transfer the data with the minimal delay so as to waste less power [1]. One of the main difficulties in MANET (Mobile Ad hoc Network) is the routing problem, which is aggravated by frequent topology changes due to node movement, radio interference and network partitions. Many Routing protocols have been proposed in past and reported in the literature. The reactive approaches only find new routes when required like AODV.

II. Description of AODV

The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for Ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes.

It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes. AODV builds routes using a route request / route reply query cycle[2]. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ[3]. If this is the case, it unicast a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it. As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route.

AODV is another variant of classical distance vector routing algorithm, based on DSDV and DSR. It shares DSR’s on demand characteristics hence discovers routes whenever it is needed via a similar route discovery process. However, AODV adopts traditional routing table; one entry per destination which is in contrast to DSR that maintains multiple route cache entries for each destination. The initial design of AODV is undertaken after the experience with DSDV routing algorithm [4].

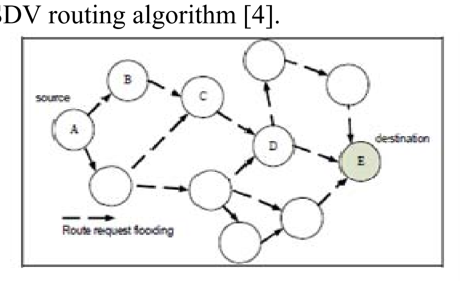

Fig. 1. Route request (RREQ) flooding

Manuscript received March 5, 2012; revised May 11, 2102.

The authors are with Dept. of Computer Science and Engineering, SRM University, India, (e-mail: mahapatra.rp@gmail.com, simenscomputer4u@yahoo.com, sachipandey_08@yahoo.co.in).

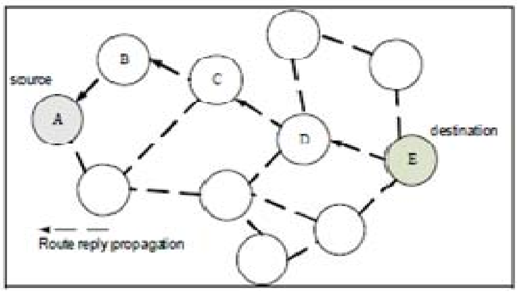

Fig. 2. Route reply (RREP) propagation

As long as the route remains active, it will continue to be maintained [5]-[7]. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery [8], [9].

III. GLOMOSIM 2.03

Global Mobile Information System Simulator (GloMoSim) simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multihop wireless communications using ad-hoc networking, and traditional Internet protocols. Developers use these simulators to model the wired or wireless network design process. It is being designed using the parallel discrete-event simulation capability provided by Parsec [10]. This makes it possible to evaluate various design alterations and configurations before even deploying the actual devices and components.

Based on the outcome of the validation process, simulations once can again be attempted for optimization of the hardware performance. Most network systems are currently build using a layered approach that is similar to the OSI seven layer network architecture. The plan is to build GloMoSim using a similar layered approach. Standard APIs will be used between the different simulation layers. This will allow the rapid integration of models developed at different layers by different people. It usually made available on a standalone machine. The goal is to build a library of parallelized models that can be used for the evaluation of variety of wireless network protocols. The proposed protocols stack will include models for the channel, radio, MAC, network, transport, and higher layers. The simple approach to designing a network simulation would be to initialize each network node in the simulation as a Parsec entity. We can view different entity initializations as being separate logical processes in the system. Hence each entity initialization requires its own stack space in the runtime. In GloMoSim, we are trying to build a simulation that will scale to thousands of nodes. If we have to instantiate an entity for each node in the runtime, the memory requirements would increase dramatically. The performance of the system would also degrade rapidly. Since there are so many entities in the simulation, the runtime would need to constantly context switch among the different entities in the system. This will cause significant degradation in the performance of the simulation. Hence initializing each node as a separate entity will inherently limit the scalability and performance of the simulation [11], [12].

To circumvent these problems network gridding was introduced into the simulation. With network gridding, a single entity can simulate several network nodes in the system. A separate data structure representing the complete state of each node is maintained within the entity. Similarly we need to maintain the right level of abstraction. When the simulation code of a particular node is being executed it should not have access to the data structures of the other nodes in the simulation. The network gridding technique means that we can increase the number of nodes in the system while maintaining the same number of entities in the simulation. In fact, the only requirement is that we need only as many entities as the number of processors on which the simulation is being run. Hence if we are running a sequential simulation we need to initialize only one entity in the system. We also don't meet the memory or context switching problems that limit the simulation.

IV. SOFTWARE REQUIRED

GloMoSim requires the following software to run. However, you need to make sure you have enough access rights of defining the computer environmental variables.

1) Microsoft VC++ version 6.0 (Essential)
2) JAVA JRE version 1.2 or higher (For VT)
3) JAVA SDK version 1.2 or higher (For VT)
4) Parsec compiler
5) GloMoSim software

Usually GloMoSim software is come along with parsec compiler if so no need to download parsec compiler separately. Otherwise download parsec compiler separately. GloMoSim tool and Parsec compiler can be downloaded free of cost from the URL

[http://pcl.cs.ucla.edu/projects/glomosim/]

Also for online Help refer to URL

[http://pcl.cs.ucla.edu/projects/parsec/]

And URL

[http://pcl.cs.ucla.edu/projects/glomosim]

Java compiler for windows can be downloaded from the sun Microsoft site or with the software cd from software vendors.

A. Installing Glomosim 2.03

Download the glomosim2.03 software from the internet. Inside the glomosim directories which consist of two sub directories, can be found namely GloMoSim and parsec. Copy these sub directories into the c drive directly. Now set the path for environmental variables as mentioned below:

1) Install visual studio (vc++) 6.0 or above. While installation check the option Register Environment variable,

for path setup

2) Go to command prompt

c:\.....\...>cl (some message will appear)

If cl not recognized appears the path is not set.

3) Must have GloMoSim 2.03 in your system which contains directories –GloMoSim and Parsec

4) Take both the directories out of GloMoSim 2.03 and place them directly into c drive (Let both are placed in c:\)(this is not necessary but for our convenience)

5) Go to My computer----- right click---properties-----Advanced tab----environmental variables tab

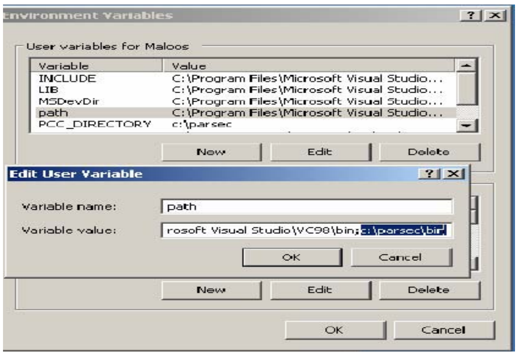

Fig. 3. Path setup in environmental variables

6) Click new tab--------------NAME PCC_Directory
VALUE c:\parsec

7) Select include and click Edit tab------

..............;c:\glomosim\include;c:\parsecinclude

Leave the already included path as it is and add above mentioned path afterwards.

Select Path and click Edit tab-----------

...............;c:\glomosim\bin;c:\glomosim\main;c:\parsec\bin

Leave the already included path as it is and add above mentioned path after wards.

Check pcc environment in DOS prompt by

C:\.......\......>pcc (then press enter)

A Message No Input files must appear if the path is properly set

C:\glomosim\main>makent (press enter)

C:\glomosim\bin>GloMoSim config.in (press enter)

GloMoSim is now ready to use.

B. Simulate Network Using Glomosim 2.03

After successfully installing GloMoSim, a simulation can be started by executing the following command in the BIN subdirectories.

C:\glomosim\bin\glomosim config.in

The inputfile contains the configuration parameters for the simulation (an example of such file is CONFIG.IN). A file called GLOMO.STAT is produced at the end of the simulation and contains all the statistics generated [13], [14].

V. The Visualization Tool

The output data of the network is generated and stored in glomo.stat file. We can use other tools to do an analysis of glomo.stat file either by Gnuplot, Qualnet and java and by making program in c by showing result in excel.

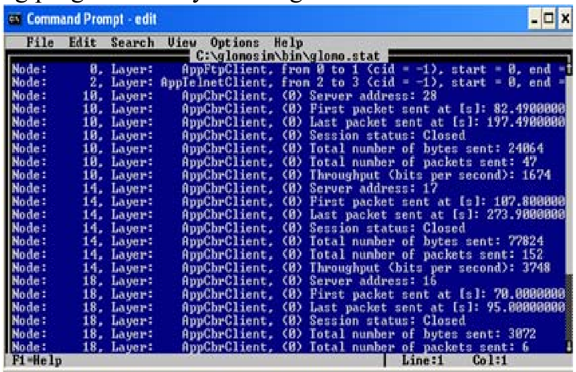

Fig. 4. Stat files to collect data

VI. Experimental Result

The network simulation is implemented using GloMoSim 2.03 simulation tool. In this scenario, AODV routing protocols are evaluated based on the two performance metrics which are

1) *Packet delivery fraction*
2) *End to End delay*

The simulation environment for this scenario is as below:

TABLE I: Parameter Evalution

Parameters	Values
Simulator	Glomosim 2.03
Protocol Studied	AODV
No. of Nodes	5,10,15,20,25,30,35
Simulation Area	2000x2000
Mobility model	Random Way point
Node speed	20m/sec

A. Packet Delivery Fraction

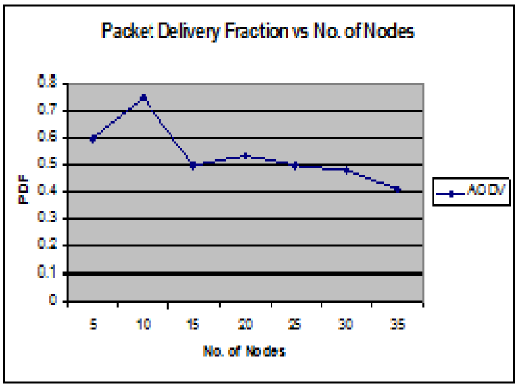

Fig. 5. Packet delivery fraction.

Based on the figure we analysis the performance of AODV on metric packet delivery fraction with varying no. of nodes.

B. *End To End Delay*

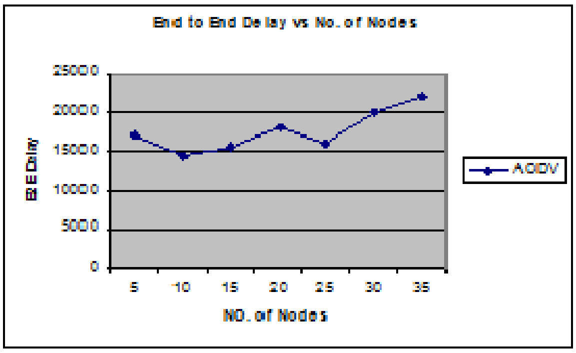

Fig. 6. End to end delay.

Based on the figure us analysis the performance of AODV on metric End to End Delay with varying no. of nodes.

VII. Conclusion

The performance of all routing protocols in MANET were measured with respect to metric like packet delivery fraction, end to end delay with varying no. of nodes by using GloMoSim simulator 2.03. This paper presents a simulation library called GloMoSim whose goal is to support accurate performance prediction of large scale network model. By the help of this simulator we can analyses the performance of any routing protocols like AODV.

References

[1] A. K. Gupta, "Performance analysis of AODV, DSR and TORA routing protocols," *IACSIT International Journal of Engineering and Technology*, vol. 2, no. 2, 2010.

[2] P. Khatri and M. Rajput, "Performance study of Ad hoc reactive routing protocols," 2010.

[3] C. E. Perkins and E. M. Royer, "Chapter ad hoc on-demand distance vector routing," Addison Wesley, 2009.

[4] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on- demand distance vector routing," *Internet Draft, draft- ietf- manet aodv-10.txt*, 2002.

[5] K. U. Rahman, R. U. Reddy, and A. Venugopal, "Performance comparison of on-demand and table driven Ad hoc routing protocols using NCTUns," *UKSIM: IEEE Tenth International Conference on Computer Modeling and Simulation,* vol. 1 , no. 3 , pp.336 – 341, april 2008.

[6] A. E. Mahmoud, R. Khalaf and A. Kayssi,"Performance comparison of the AODV and DSDV routing protocols in mobile Ad hoc networks," Lebanon, 2007.

[7] A. A. Maashri, M. O. Khaoua, "Performance Analysis of MANET Routing Protocols in the Presence of Self- Similar Traffic," *IEEE, First* published *in Proc. of the 31st IEEE Conference on Local Computer Networks*, 2006.

[8] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on- demand distance vector routing," Internet Draft, draft- ietf- manet aodv-10.txt, 2002.

[9] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps*., pp. 90–100, feb. 1999.

[10] GloMoSim: Global mobile information systems simulation library. [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[11] S. Bhatt, R. Fujimoto, A. Ogieski, and K. Permalla, "Parallel simulation techniques for large scale networks," *IEEE Communication Magazine*, pp. 42-47,1998.

[12] A. Lee, K. Xu, "GloMoSim java visualization tool. documentation version 1.1," Software Distribution.

[13] R. Bagrodia, "Readme file – GloMoSim software," University of California, Los Angeles 90095-1596.

[14] Tutorial for the network simulator "Glomosim 2.03". [Online]. Availble: http://www.isi.edu/nsnam/ns/tutorial.

Manish Bhardwaj M.Tech (Computer Science & Engineering) presently working as Assistant Professor (CSE) at Modinagar Campus of SRM University Chennai; contributing his enormous research works in the area of Computer Science, Simulations, Mobile Ad hoc network protocols.

Rajendra Prasad Mahapatra PhD (CSE) presently working as Associate Dean at Modinagar Campus of SRM University Chennai; contributing his enormous research works in the area contributing his enormous research works in the field of Computer Science, Simulations, Network Security and Artificial Intelligence and RF Technology.

Sachi Pandey M. Tech (Computer Science & Engineering) presently working as Assistant Professor (CSE) at Modinagar Campus of SRM University Chennai; contributing his enormous research works in the area of Computer Science, Simulations, Mobile Ad hoc network protocols.