ISSN:0975-9646

Simulation of MANET using GloMoSim Network Simulator

Kriti Jaiswal , Om Prakash

Department of Computer Science & Engineering,
United College of Engineering & Research
Allahabad, U.P, India

***Abstract*— Mobile ad hoc network (MANET) is a temporary self-organizing network of wireless mobile nodes without the support of any existing infrastructure that may be readily available on the conventional networks. It allows various devices to form a network in areas where no communication infrastructure exists. In order to maintain connectivity in a mobile ad hoc network all participating nodes have to perform routing of network traffic. This paper is carried out to simulate the MANET using GloMoSim. It is a popular network simulation tool, which is used in the study of the behavior of large-scale hybrid networks that include wireless, wired, and satellite based communications. With the advent of the scenarios seeking infrastructure-less network with wireless nodes, security concerns have made the required personnel to mull over the MANET configuration using protocols for various sized networks. We will also discuss various categories of protocols proposed for the MANET, applications that are found out of this network, overview of the AODV protocol and GloMoSim and most importantly the simulations that are carried out using AODV protocol.**

Keywords— Mobile ad-hoc network (MANET), AODV, GloMoSim and GnuPlot.

I. INTRODUCTION

A mobile ad hoc network (MANET) is a temporary self-organizing network of wireless mobile nodes without the support of any existing infrastructure that may be readily available on the conventional networks [1]. It allows various devices to form a network in areas where no communication infrastructure exists [9]. In order to maintain connectivity in a mobile ad hoc network all participating nodes have to perform routing of network traffic. The basic idea of the design is to operate each mobile node as a specialized router, which periodically advertises its view of the interconnection topology with other mobile nodes within the network [2]. Since there is no fixed infrastructure available for MANET and due to the dynamic nature of the nodes, routing becomes an extremely important issue.

The simulations carried out AODV routing protocol using GloMoSim network simulator could help in setting up such networks in real-life scenarios. Providing security to the MANET would help in establishing "on demand wireless networks" without the fear of any menace. The MANET could then be used in number of communication purposes.

It is said “Prevention is better than cure”. Carrying out simulations before setting up network of any type in real-life obviously discards waste of resources, labor and time. Thus keeping up with the lines, the paper is based on the simulation of MANET using GloMoSim network simulator. To overcome the security concerns that hover on this network a proposal for implementation of Diffie- Hellman Key agreement is given. This could help addressing concerns for:

- Integrity
- Authenticity
- Confidentiality

The complete paper is organized as follows. In Section 2, we describe the literature review of the existing routing protocol for MANETS, applications and security issues of MANET. In Section 3, we describe AODV routing protocol in brief which utilized in our work. In Section 4, we give a review of GloMoSim network simulator. In Section 5, the simulation is carried out and the results of the simulation and settings are explained. In Section 6, we plot a graph of the simulation by using Gnu Plot and performance analysis is done in this section. In Section 7, we provide our concluding remarks and future scope.

II. LITERATURE REVIEW

This section describes the categories of existing routing protocol for MANETs, application & its security issues.

A. *MANET Protocols*

Many protocols have been proposed for MANETs. These protocols can be divided into three categories: *proactive, reactive, and hybrid* [3]. Proactive methods maintain routes to all nodes, including nodes to which no packets are sent. Such methods react to topology changes, even if no traffic is affected by the changes. They are also called table-driven methods. Reactive methods are based on demand for data transmission. Routes between hosts are determined only when they are explicitly needed to forward packets. Reactive methods are also called on-demand methods. They can significantly reduce routing overhead when the traffic is lightweight and the topology changes less dramatically, since they do not need to update route information periodically and do not need to find and maintain routes on which there is no traffic. Hybrid methods combine proactive and reactive methods to find efficient routes, without much control overhead [10].

1) *Proactive Routing Protocols:* In order to maintain correct route information, a node must periodically send control messages. Therefore, proactive routing protocols may waste bandwidth since control messages are sent out unnecessarily when there is no data traffic. E.g.: *Global*

State Routing (GSR) is based on the Link State (LS) routing method.

2) *Reactive Routing Protocols*: Reactive routing protocols can dramatically reduce routing overhead because they do not need to search for and maintain the routes on which there is no data traffic. This property is very appealing in the resource-limited environment. E.g.:

Dynamic Source Routing (DSR) protocol uses the source routing approach (every data packet carries the whole path information in its header) to forward packets [4]. Before a source node sends data packets, it must know the total path to the destination. Otherwise, it will initiate a route discovery phase by flooding a Route Request (RREQ) message. The RREQ message carries the sequence of hops it passed through in the message header. Any nodes that have received the same RREQ message will not broadcast it again. Once an RREQ message reaches the destination node, the destination node will reply with a Route Reply (RREP) packet to the source. The RREP packet will carry the path information obtained from the RREQ packet. When the RREP packet traverses backward to the source, the source and all traversed nodes will know the route to the destination. Each node uses a route cache to record the complete route to desired destinations.

Ad hoc On-Demand Distance Vector (AODV): Since DSR includes the entire route information in the data packet header; it may waste bandwidth and degrade performance, especially when the data contents in a packet are small. Ad hoc On-Demand Distance Vector (AODV) Routing tries to improve performance by keeping the routing information in each node. The main difference between AODV and DSR is that DSR uses source routing while AODV uses forwarding tables at each node. In AODV, the route is calculated hop by hop. Therefore, the data packet need not include the total path [5].

3) *Hybrid Routing Protocols:* A typical hybrid routing protocol is Zone Based Routing (ZBR) [6]. ZBR combines the proactive and reactive routing approaches. It divides the network into routing zones. The routing zone of a node A includes all nodes within hop distance at most d from node A. All nodes at hop distance exactly d are said to be the peripheral nodes of node A's routing zone. The parameter d is the zone radius. ZBR proactively maintains the routes within the routing zones and reactively searches for routes to destinations beyond a node's routing zone. Route discovery is similar to that in DSR with the difference that route requests are propagated only via peripheral nodes. ZBR can be dynamically configured to a particular network through adjustment of the parameter d. ZBR will be a purely reactive routing protocol when d = 0 and a purely proactive routing protocol when d is set to the diameter of the network.

B. *Application of MANET*

1) Pure general purpose MANET: The mostly discussed application scenario for pure general-purpose MANET is Battlefield or disaster-recovery networks. However, these kinds of networks have not yet achieved the envisaged impact in terms of real world implementation and industrial deployment.

2) Mesh networks: Unlike pure MANETs, a mesh network introduces a hierarchy in the network architecture by adding dedicated nodes (called mesh routers) that communicate wirelessly to construct a wireless backbone.

- Intelligent transportation systems.
- Public Safety
- Mesh community.

3) Opportunistic Networking (Delay Tolerant Networking):

- Wildlife monitoring:
- Opportunistic networks for developing

4) Vehicular ad hoc networks

- Safety Related Application
- Comfort Application
- Application for Administration

5) Wireless sensor networks:

- Habitat and Environmental Monitoring for Scientific Applications:
- Monitoring for Civilian Applications

C. *Security Issues in MANET*

Ad hoc wireless network does not have any predefined infrastructure and all network services are configured and created on the fly. Thus it is obvious that with lack of infrastructural support and susceptible wireless link attacks, security in ad hoc network becomes inherent weakness. The attacks are generally inclined to hamper the availability, integrity, confidentiality and authenticity related issues [7]. The attack on MANET can be classified as the *active and passive attacks* [8].

1) Passive attacks: A passive routing attack does not disrupt the operation of a routing protocol, but only attempts to discover valuable information by listening to the routing traffic. Hence such attacks are difficult to detect.

2) Active attacks: An active attack attempts to improperly modify data, gain authentication, or procure authorization by inserting false packets into the data steam or modifying packets transition through the network. Active attacks are of *two types: external and internal*. An *external attack* is one caused by nodes that do not belong to the network. An *internal attack* is one from compromised or hijacked nodes that belong to the network. As malicious nodes already belong to the network as authorized parties, and hence are protected with network security mechanisms and services, therefore, internal attacks are more severe.

III. AD-HOC ON DEMAND DISTANCE VECTOR (AODV) ROUTING PROTOCOL

AODV enables "dynamic, self-starting, multi-hop routing between mobile nodes wishing to establish and maintain an ad hoc network. AODV allows for the construction of routes to specific destinations and does not require that nodes keep these routes when they are not in active communication. It avoids the "counting to infinity" problem by using destination sequence numbers. This makes AODV loop-free.

A. *Overview*

AODV defines 3 message types: *Route Requests (RREQs), Route Replies (RREPs), Route Errors (RERRs)*. RREQ

messages are used to initiate the route finding process. RREP messages are used to finalize the routes. RERR messages are used to notify the network of a link breakage in an active route. The AODV protocol is only used when two endpoints do not have a valid active route to each other. Nodes keep a "precursor list" that contains the IP address for each of its neighbors that are likely to use it for a next hop in their routing table. Route table information must be kept for all routes even short-lived routes. The routing table fields used by AODV are: Destination IP Address, Destination Sequence Number, Valid Destination Sequence number flag, other state and routing flags, Network Interface, Hop Count, Next Hop, List of Precursors, and Lifetime [11][12].

B. *Simple Example*

Fig. 1 A wants to communicate B

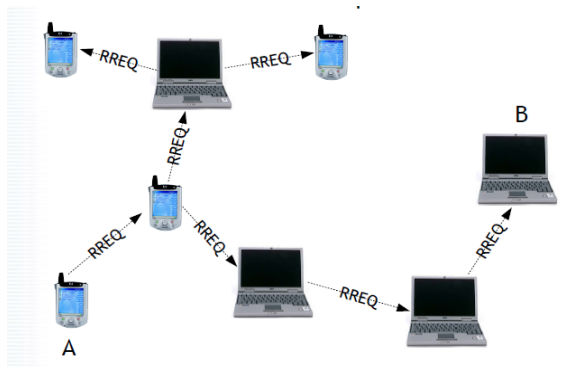

Fig. 2 A floods Route Request to all

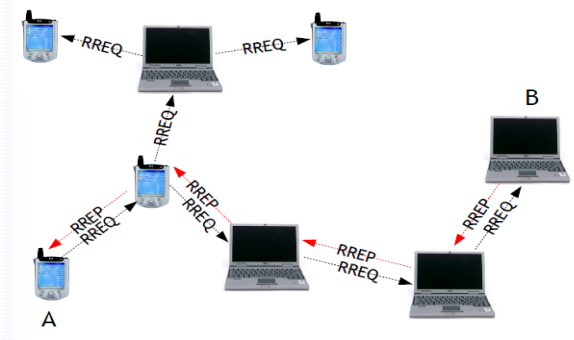

Fig. 3 B unicasted back Route Reply to A

C. *Maintaining Sequence Number*

The proper maintenance of sequence numbers is crucial to keeping AODV loop-free and thereby avoiding the "counting to infinity" problem [13]. The sequence number in the RREP message is greater than the stored number, or the sequence numbers are identical, but the route is marked as inactive, or the sequence numbers are the same, but the hop count is smaller for the RREP message. Nodes originating RREQ messages must increment their own sequence number before transmitting the RREQ. Destination nodes increment their sequence numbers when the sequence number in the RREQ is equal to their stored number. Forwarding nodes only update their stored sequence number for a given destination when forwarding RREP messages and only when: The sequence number in the routing table is invalid, or The sequence number in the RREP message is greater than the stored number, or the sequence numbers are identical, but the route is marked as inactive, or the sequence numbers are the same, but the hop count is smaller for the RREP message. Nodes originating RREQ messages must increment their own sequence number before transmitting the RREQ. Destination nodes increment their sequence numbers when the sequence number in the RREQ is equal to their stored number.

D. *Links Breaks*

RERR message processing is initiated when: Node detects a link break for the next hop of an active route, or receives a data packet destined for a node for which it has no (active) route, or receives a RERR message from a neighbor for at least one active route in its routing table .Nodes must increment the destination sequence numbers of the routing entries contained in the RERR message before transmitting to nodes in precursor list. Nodes receiving RERR messages simply update their sequence numbers with those contained in the RERR message. Nodes must also mark these routing entries as invalid regardless of whether they are transmitting and/or receiving. This ensures that no predecessors may reply to a RREQ from a node on their successor path, thus providing loop-freedom. RREQ messages are ultimately forwarded back to the originator who may initiate another RREQ message.

E. *Local Repairs*

Nodes detecting a link breakage can choose to repair the link if possible. The node simply increments the destination sequence number and broadcasts a RREQ message. If it receives a RREP message then the repair was successful.

IV. Introduction of GloMoSim

GloMoSim stands for *Global Mobile Information System Simulator* and is a scalable network simulation environment for mobile ad-hoc networks, developed at UCLA Parallel Computing Laboratory. GloMoSim uses a parallel discrete-event simulation capability provided by *Parsec (Parallel Simulation Environment for Complex Systems)* which is C based simulation language. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols [14]. The TABLEI lists the GloMoSim models currently available at each of the major layers.

The *node aggregation technique* is introduced into GloMoSim to give significant benefits to the simulation performance. Initializing each node as a separate entity inherently limits the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system. Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation [15]. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined by the physical position of the nodes. Global Mobile Information System Simulator is a popular network simulation tool, which is frequently used in the study of the behaviour of large-scale hybrid networks that include wireless, wired, and satellite based communications are becoming common in both in military and commercial situations [17]. *It is available for various Linux flavours files include Freebsd-3.3, Aix, Irix, Redhat-6.0, Redhat-7.2 and Solaris. It is also available for Windows*. GloMoSim is a popular simulation tool that is freely available for education, or research, or to non-profit agencies, which means you can enhance it to suit your own requirements.

TABLE I
GLOMOSIM MODELS AT EACH LAYER

Layer	Models
Physical (Radio Propagation)	Free space, Two-Ray
Data Link (MAC)	CSMA, MACA, TSMA, 802.11
Network (Routing)	Bellman-Ford, FSR, OSPF, DSR, WRP, LAR, AODV
Transport	TCP, UDP
Application	Telnet, FTP

A. Installation of GloMoSim

To install GloMoSim Software we required following software's include (in Fedora Linux i.e. used in this paper simulation section):

- Linux of Java 1.3 or higher versions [19]
- Parsec Compiler [16]
- GloMoSim software[17]

B. Testing GloMosim

1) *Goto glomosim/bin*
2) *Type ./glomosim config.in*

C. Customizing GloMoSim

 # vi .bash_profile

// to open the .bash_profile to customize glomosim software.

As such, the path of glomosim could be anywhere on the file system. However, it is recommended you install glomosim on the file system of a user other than root. In order to allow glomosim to run from the user's file system, you may need to make a few changes to the .bash_profile file under the home directory of the user. This will also allow you to use glomosim without having to log in as the all-powered root user. The following changes then should be made to the file .bash_profile.

 # User specific environment and startup programs

PATH=$PATH:$HOME/bin:/usr/java/jdk1.7.0_03/bin:/glmosim2.03/glomosim/main:/glomosim-2.03/glomosim/include:/glomosim-2.03/glomosim/bin:/glomosim-2.03/parsec/bin:/glomosim-2.03/parsec/include

PCC_DIRECTORY=/glomosim-2.03/parsec

export PATH PCC_DIRECTORY

D. Use of GloMoSim Simulator

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

./glomosim < inputfile >

The <*input file*> 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.

E. The Visualization Tool

GloMoSim has a Visualization Tool that is *platform independent* because it is coded in Java [20]. To initialize the Visualization Tool, we must execute from the *java gui* directory the following:

java GlomoMain

This tool allows to debug and verify models and scenarios; stop, resume and step execution; show packet transmissions, show mobility groups in different colors and show statistics. The radio layer is displayed in the Visualization Tool as follows: When a node transmits a packet, a yellow link is drawn from this node to all nodes within its power range. As each node receives the packet, the link is erased and a green line is drawn for successful reception and a red line is drawn for unsuccessful reception. No distinction is made between different packet types (ie: control packets vs. regular packets, etc.)

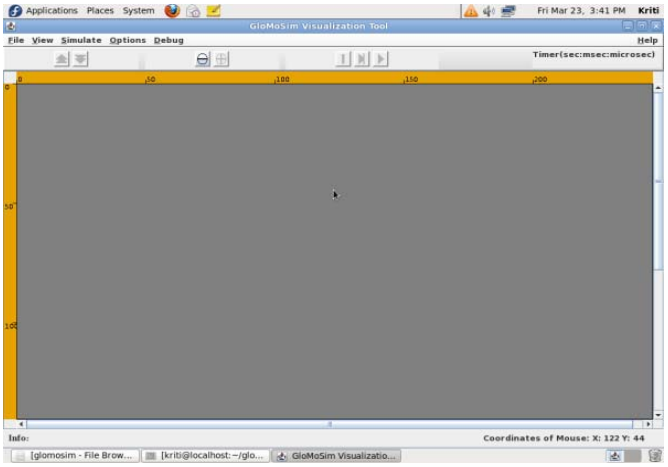

Fig. 4 the Visualization Tool

V. SIMULATION

A. *Setting up a Scenario of GloMoSim*

For this, firstly we have to install the GloMoSim properly and test this. After customize the GloMoSim before use it, stated earlier. After successfully doing this, a simulation can be started by executing the following command in the BIN subdirectory.

./glomosim inputfile > bell.trace

The input file contains the configuration parameters for the simulation (an example of such file is CONFIG.IN & APP.CONF) [14][21]. A file called GLOMO.STAT is produced at the end of the simulation and contains all the statistics generated [17].

The simulation environment for scenario is as below:

TABLE II
PARAMETER EVALUTION

Parameters	**Values**
OS	Fedora 13
Simulator	GloMoSim
Protocol Studied	AODV
Application	Telnet
No. of nodes	10, 20, 30, 40
Simulation Area	200, 300, 400, 500, 600
Mobility Model	Random Way point

After setting the files, the simulation is carried out like this:

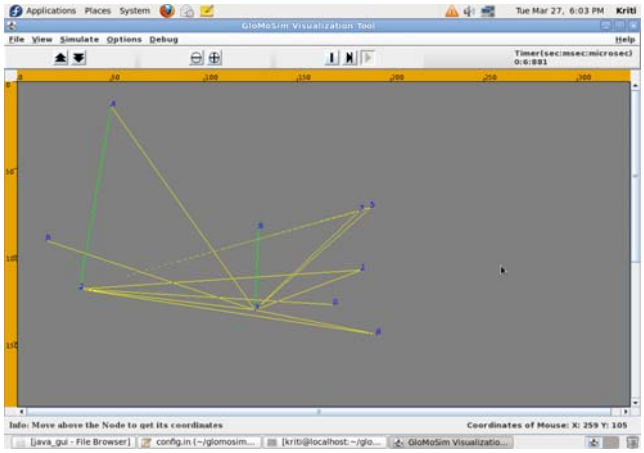

Fig. 5 Simulation

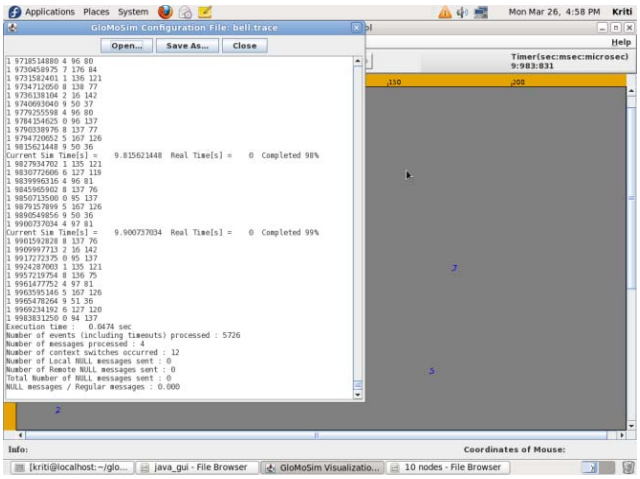

Fig. 6 bell.trace File

B. *Calculating the Radio Range*

The transmission power is expressed in dBm in the configuration file. This makes it somewhat difficult to get an idea of the transmission range in meters. Therefore a program called radio_range is shipped with the GloMoSim environment. Use the syntax below to calculate the transmission range. The configuration file contains the radio transmission power in dBm and therefore it needs to be included as an argument to the program [14].

%/radio_range config.in

C. Result Analysis

The output data of the network is generated and stored in glomo.stat file. In these cases we use other tools to do an analysis of the glomo.stat file. The following *shell program* and *awk programs* are used to display packet data ratio, percentage of loss rate and packets sent.

1)Example: For 10 nodes

Shell program:

File Name : result1.sh

Cat $1 | grep App | grep Total | grep packets | awk –f analysis.awk

AWK program:

File Name: analysis.awk

```
BEGIN{
sumcountsent = 26;
sumcountrcv = 26;
}
{
if($10 =="sent:") sumcountsent+=$11;
else if ($10 == "received:") sumcountrcv+=$11;
}
END{
printf("Loss Packet Percentage= %f \% \n",100-((sumcountrcv*100)/sumcountsent));
printf("Packet Delivery Ratio = %f \n\n",sumcountrcv/sumcountsent);
printf("Packet Received = %d \n", sumcountrcv);
printf("Packet Sent = %d \n\n", sumcountsent);
}
```

In command prompt:

sh result1.sh glomo.stat

Loss Packet Percentage =0.000000 %
Packet Delivery Ratio = 1.000000
Packets Received = 26
Packets Sent = 26

2) Transmission Range For 10 Nodes:

[kriti@localhost ~]$ cd glomosim-2.03/glomosim/bin
[kriti@localhost bin]$ ./radio_range config.in

radio range: 376.782m
Execution time: 0.0281 sec
Number of messages processed: 0
Number of context switches occurred: 6
Number of Local NULL messages sent: 0

Number of Remote NULL messages sent: 0
Total Number of NULL messages sent: 0
Results of simulation: We want to obtain a graph of the variance of TTL of packets delivered to final destination, with networks of different size (10,20,30 and 40 nodes) and nodes placed at different terrain dimension (200,300,400,500 and 600).

TABLE II
RESULT OF SIMULATION

Terrain Dimension	10 Nodes	20 Nodes	30 Nodes	40 Nodes
200	1638	1260	882	4032
300	1764	3276	504	2646
400	2016	504	2314	504
500	1368	2520	2268	630
600	2268	1764	2520	3402

If the data of this table is used in a *data.txt* file, then we may use graphic tools such as *GnuPlot* to obtain an appropriate graph [22].

VI. PERFORMANCE ANALYSIS

The performance measurement of this is done by obtaining the graph for different-different simulations. The graph is obtained using *GNUPLOT Graphics Tool* [23].

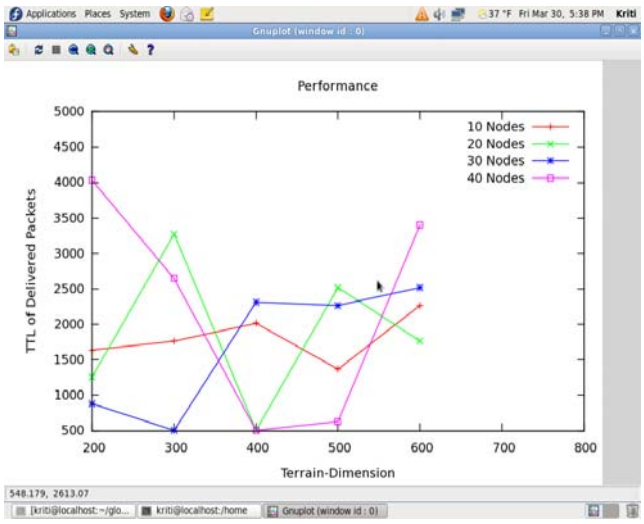

Fig. 7 Graph of GnuPlot

VII. CONCLUSION & FUTURE WORK

This paper presents a reviewed existing routing protocols, applications and security issues of MANET. This investigation enables researchers to focus on the issues surrounding MANET and its applications. This paper lays down a bridge of concepts which could be walked on to reach the final destination of MANET formation. The concept of the AODV protocol that serves in the configuration of infrastructure less and wireless network is discussed decently. This paper presented GloMoSim whose goal is to support accurate performance prediction of large-scale network model. The simulation part helps providing interferences that could be incorporated in setting up such networks in the real life scenarios.
In future, the purposes that MANET serves are likely to extend to important communications too. But AODV defines no special security mechanisms for it to be unbleached by attackers. Thus, availability, confidentiality, integrity and authenticity of the data collectively becomes a set of important issues to be dealt with a vision to enhance the quality of service being catered by MANET we would like to propose *Diffie-Key Agreement Algorithm*.

ACKNOWLEDGMENT

This work is the result of good support and guidance with facilities, which were provided to us by the Department of Computer Science and Engineering. The authors would like to express their deepest gratitude to the faculty of United College of Engineering & Research, Allahabad, India for his guidance, support, and contribution for this paper. At last but not least, I want to thank all the members of the Department who helped me towards the completion of my paper.

REFERENCES

[1] "Mobile Ad-hoc Networks (MANET) Working Group" http://www.ietf.org/html.characters/manet-charter.html.2004.
[2] Satyabrata Chakrabarti and Amitabh Mishra, "Quality of service in Mobile Ad-hoc Networks", CRC press LLC 2003.
[3] M. Abolhasn, T. Wysocki and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks", Ad Hoc Networks 2, Elsevier, pp. 1-22, 2004.
[4] D. Johnson, B.D.A. Maltz, and Y.C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad-Hoc Networks (DSR)", draft-ietf-manet-dsr-10.txt, 2004.
[5] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, N/w Working Group, 2003.
[6] Z. J. Haas, "The Zone Routing Protocol (ZRP) for ad-hoc networks", Internet Draft, Nov.1997.
[7] Preetida Vinayakray-Jani, "Security within Ad-hoc Networks", Position Paper, PAMPAS Workshop, Sept. 16/17 , London.
[8] William Stalling, "Cryptography & Network Security".
[9] Marco Conti, Silvia Giordano, "Multihop ad-hoc Networking: The Theory", IEEE Communication Magazine, April 2007.
[10] Patroklos G. Argyroudis, Donal O' Mahony, "Secure Routing for Mobile Ad-Hoc Networks".
[11] Charles E. Perkins and Elizabeth Royer, "Ad-Hoc On-Demand Distance Vector Routing", Addison Wesley, 2009.
[12] Jani Lakkakorpi, "The Ad Hoc On-Demand Distance-Vector Protocol: Quality of Service Extension".
[13] C.E. Perkins, "Ad-Hoc Networking", Chapter 6, Addison Wesley, 2001.
[14] Jorge Nuevo, "A Comprehensible GloMoSim Tutorial", INRS-Université du Québec, March 4, 2004.
[15] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Ken Tang, R. Bagrodia, Mario Gerla, "GloMoSim: A Scalable Network Simulation Environment", Technical Report 990027, University of California, 13,1999.
[16] Parsec Compiler free download: http://pcl.cs.ucla.edu/projects/parsec/
[17] "GloMoSim: Global Mobile Information Systems Simulation Library", http://pcl.cs.ucla.edu/projects/glomosim/
[18] R. Bagrodia, "Readme file- GloMoSim Software", University of California, Los Angeles, 90095-1596.
[19] Java software for Linux free download: http://www.java.com/en/download/manual.jsp
[20] Addison Lee- Kaixin Xu, "GloMoSim Java Visualization Tool" Documentation Version 1.1, Software Distribution.
[21] Thomas Nilsson, "A Tutorial On GloMoSim", Umeå University, Swedon.
[22] "Gnuplot", Documentation version 4.0, Software Distribution, www.gnuplot.info/
[23] "Gnuoplot 4.2- A Brief Maunual and Tutorial", Duke University, Durham, NC 27708-0287. www.duke.edu/~hpgavin/gnuplot.html