A Simplified Methodology for Random Topology Generator builds in Ad Hoc Network Test Bed

Nitiket N. Mhala
Department of Electronics Engg., BDCOE,
Sevagram(Wardha), India.

N. K. Choudhari
Principal, Bhagwati Chadurvedi COE
Nagpur, India

Abstract- Emulation that uses iptables commands to create a desired logical topology is a technique that has been widely used in laboratory environments. The iptables commands are input directives for the iptables packet-filtering system that controls the inbound data packets in real time to emulate the connection between any two communicating computers. The packet-filtering process is based on the hardware address of the network interface card of the sender, and it can be accomplished in a wired or wireless network. Here, we effort to automatically generate arbitrary logical network topologies. This paper briefly focuses on simplified methodology we adopted for generation of Logical Topology in Adhoc Network Test Bed in our research laboratory environment without having to physically move the nodes in the adhoc network
Keyword: iptables ,adhoc ,testbed,adjacency matrix.logical topology

I. INTRODUCTION

Recent advances in wireless communication technology and portable computing devices such as wireless handhelds, Personal Digital Assistants (PDAs) and other mobile information terminals are driving a revolutionary change in our information society towards the era of mobile communications. Mobile users can utilize several electronic platforms simultaneously through which they can access all the necessary information whenever and wherever required [1]. The nature of ubiquitous devices makes wireless networks the easiest solution for their interconnection and, as a consequence, the wireless arena has been experiencing rapid growth in the past decade. Not only are the mobile devices getting smaller, cheaper, more convenient, and more powerful, they also run more applications and network services, thus fueling the explosive growth of mobile computing equipment market. The ever increasing demand for the Internet access anytime anywhere has generated vast interest and intensive works in developing and improving wireless and mobile network protocols for portable computing devices. Today, the widely deployed network type is based on a centralized approach that requires a network point of access, commonly called the Access Point (AP), to serves as a gateway between the mobile device and the Internet. While infrastructure-based networks provide a great way for mobile devices to gain network services, it takes time and potentially high cost to set up the necessary infrastructure [2]. Because of the limitation of radio range, radio must be within the vicinity of an AP to be connected. This is undesirable due to the cost of time especially in the marketing world. Additionally, it is important to note that, in the cases of natural catastrophe, war, or geographic isolation, communication infrastructure may be broken down or overloaded; thus, the network connectivity may not be always attainable [3]. In the situations where required networking connections are not available in a given geographic area, providing the needed connectivity and network services becomes a real challenge. Ad hoc technologies have emerged to solve this problem. Mobile Ad-hoc Network (MANET) is a collection of mobile computing devices that communicate via wireless links, without the aid of infrastructures such as base-station [4]. Its topology is dynamic and nodes are free to join or leave arbitrarily. Traditionally, ad hoc technology has only been exploited in tactical networks to improve battlefield communications. Early ad hoc networking applications can be traced back to the Packet Radio Network (PR Net) project of Defense Advanced Research Projects Agency (DARPA) in 1972, which was primarily inspired by the efficiency of the packet switching technology, such as bandwidth sharing and store-and-forward routing, and its possible application in mobile wireless environment [5]. In the middle of 1990, with the introduction of new enabling technologies such as IEEE 802.11 and Bluetooth, commercial radio technologies have begun to appear on the market [5]. Lately, the wireless research community became aware of the great commercial potential and advantages of MANET outside the military arena due to the wide diversity of the potential MANET applications, ranging from the instantaneous conference in classroom to vehicular services [1]. Due to the characteristics of MANET, an adaptive and distributed routing protocol is necessary to provide a communication platform that is solid and dynamic in the face of widely fluctuating wireless channel characteristics and node mobility. Many existing proposed routing protocols have been evaluated and compared on the basis of simulation results [6, 7, 8, 9, and 10]. In reality, connectivity and performance in MANETs can be affected by several factors, such as the limitations of Operating System (OS) protocol stack and slow buffer allocation, and simulations cannot account for all of them [11]

II. Research Need of Building of Ad hoc Network Test Bed

The dynamical nature of an ad hoc network is very interesting. The strength of the connection can change rapidly in time or even disappear completely. Nodes can appear, disappear and re-appear as the time goes on and all the time the network connection should work between the nodes that are part of it. As one can easily imagine, the situation in adhoc networks with respect to ensuring connectivity and robustness is much more demanding than in the wired case. Generally researchers traditionally use simulations because they easily allow for a large number of nodes and reproducible environment conditions. In simulation, the developer controls the whole system, which is in effect only a single component. But on the other hand, here, our submission is for an implementation in real world, which needs to interoperate with a large, complex system. Some components of this system are operating systems, network interfaces and suitable wireless drivers. Orchestrating a live field trial of wireless mobile networking involves significant cost and logistical issues relating to mobile platforms, support personnel, network and experiment automation and support equipment. The significant cost and logistics required to execute such a field trial can also be limiting in terms of achieving meaningful test results that exercise a practical number of mobile nodes over a significant set of test conditions within a given time. There is no argument that field trials are an important component of dynamic network testing. A field test of prototype will show whether simulations were on right track or not, but that's a big leap to take; going from the simulator directly to the real thing. In conceiving our work, we envisioned a mobile network emulation system that is low cost, flexible and controllable.

1) Logical Implementation of Ad hoc Network on Each node

Our Approach [12] is based on IEEE802.11 standardized medium access protocol based on collision avoidance and tolerated hidden terminals usable for building mobile adhoc network using notebooks and 802.11 PCMCIA cards. The Basic purpose is to constitute an adhoc network under Linux in a laboratory environment for the academic research purpose.

The physical layer must adapt to rapid changes in link characteristics. The Multiple Access control (MAC) layer needs to minimize collisions, allow fair access and semi reliably transports data over the short wireless links in the presence of rapid changes and hidden or exposed terminals. The network layer needs to determine and distribute information used to calculate paths in a way that maintains efficiency when links change often and bandwidth is at premium. It also needs to integrate smoothly with traditional, non adhoc-aware internet works and perform functions such as auto configuration in this changing environment. The Transport layer must be able to handle delay and packet loss statistics that are very different than wired networks. Finally; applications need to be designed to handle frequent connection and disconnection with peer applications as well as widely varying delay and packet loss Characteristics

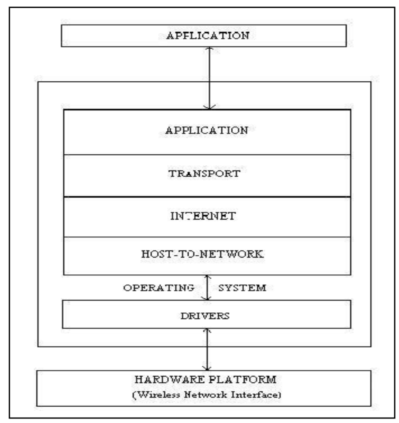

Figure 1. Logical Implementation of Ad hoc Network on each Node

2) Basic Idea of our Test bed

The basic idea of our emulation Test Bed is having a number of MANET nodes physically close to each other inside the laboratory, but forces them to 'think' that they can only communicate with a selected few of them. That way, we can emulate a logical topology. In order to work, there is the need of hardware. Our Test bed is an emulator, not a simulator. So, we have the necessary hardware equipment. Each node is a device that has a wireless (802.11 b/g) interface, so that it can communicate with other adhoc nodes and run MANET protocols. In addition, the device should also have a wired interface (Ethernet), which is used for administrative purposes. In brief, the Test Bed uses the wired interface to transfer files needed for its operation to and from the node and manipulate its networking element in such way that will create logical topology we want. That Leaves the wireless interface free of any interface and most importantly, emulates an actual MANET, which is the whole point all along. In fact, Test Bed uses ix86 architecture and Linux can run even in 80386 machines (at least requirement is Pentium II), so we can gather all those old PCs intended thrown away, adding a PCMCIA wireless card on each of them and set up a MANET test bed in a laboratory at a very low cost.

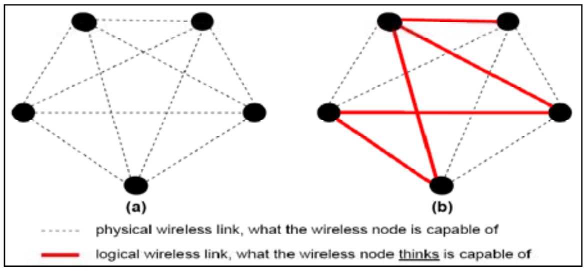

Figure2. (a) Physical links, (b) logical links between MANET nodes

III. RESEARCH MOTIVATION FOR RANDOM TOPOLOGY GENERATOR

After several years, the graph family [13] has been developed into modern era. Now they want to be part in the computer. Unfortunately, computer only understand numbers, not graph. To suit graph family into computer, they think on how to represent them into numbers that computer can understand. After long and many discussions, they finally agree to let the computer know about graph by a simple representation called *Matrix.* Matrix is a table of numbers. Each element in the matrix table can be filled by a number. One graph is represented by one matrix. Can we figure out what is the relationship between matrix and its graph? The graph family wants to be represented in the computer through matrix. However, there are many methods to represent a graph in a matrix. Each method has its own advantages. Here Adjacency matrix [14] is used to represent a graph in a matrix.

a) Adjacency Matrix*:*

Some types of graph are defined by knowing, for each pair of nodes, whether or not there is an arc connecting them. In the case of directed graphs, it matters which way round we consider the pair of nodes. There may be an arc in one direction but not in the other. Assuming a correspondence between the n nodes of a graph and the integers $0,\ldots,n-1$, the graph can be defined by an $n \times n$ matrix of boolean values. The (i,j)'th entry of the matrix is true if there is an arc from node i to node j and false otherwise. This is referred to as an **adjacency matrix** representation. For a weighted graph it will be necessary to provide storage for the weights, either in a parallel $n \times n$ numerical matrix, or by having a matrix of pairs of Booleans and numbers or, possibly, by collapsing the representation into a single matrix by using a device such as indicating non-existence of paths by ``infinite'' weights. The matrix representation can have advantages in terms of simplicity and time efficiency. The graph family argues that one of the best ways to represent them into a matrix is by counting the number of edge between two adjacent vertices. Two vertices is said to be *adjacent* or neighbor if it support at least one common edge. Graph below has three vertices. Thus, we make adjacency matrix of size 3 by 3. Then we put the name of vertices on the side of the matrix. Look at the picture and we start with an empty matrix. Only the names of vertices are there

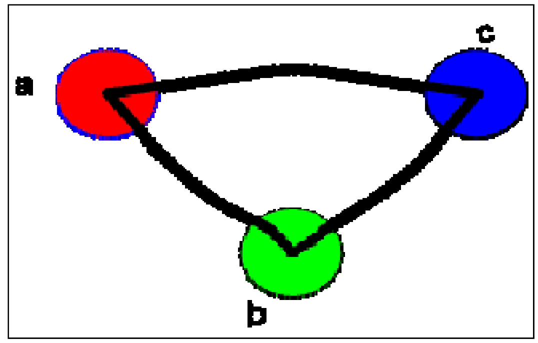

Fig 3: Graph with three Vertices

$$\begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} \_ & \_ & \_ \\ \_ & \_ & \_ \\ \_ & \_ & \_ \end{bmatrix} \end{array} \quad \begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}$$

To fill the adjacency matrix, we look at the name of the vertex in row and column. If those vertices are connected by an edge or more, we count number of edges and put this number as matrix element.

Vertex a and vertex b has one common edge, we say that Vertex a and vertex b are adjacent (neighbor). We input the number of edge in the matrix cell that correspond to vertex a and vertex b

$$\begin{array}{c} \\ a \\ b \\ c \end{array}\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} \_ & 1 & 1 \\ 1 & \_ & \_ \\ 1 & \_ & \_ \end{bmatrix} \end{array}$$

Vertex a and c is adjacent by one edge. Thus, we input the number of edge in the matrix cell that correspond to Vertex a and c.

$$\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} \_ & 1 & 1 \\ 1 & \_ & 1 \\ 1 & 1 & \_ \end{bmatrix} \end{array}$$

There is no other edge on the graph, thus we put the rest of unfilled cells in the matrix as zero

$$\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}$$

The matrix to represent a graph in this way is called Adjacency matrix

b) IPTABLES Filtering

In the initial design, a simple propagation range model is used to emulate a nominal wireless link range, and a dynamic method for blocking packets from particular nodes out of range is used. The PREROUTING filtering chain function of IP TABLES is a dynamic way to achieve this in a low cost open design manner. IPTABLES [15] is standard system software that comes with a variety of open source operating systems (e.g., Linux) to filter network packets on a given network interface. By inserting and deleting entries in the PREROUTING chain of the IPTABLES one can emulate logical wireless connectivity or topological dynamics. Although supportable, filtering by IP address was not chosen to perform the blocking and unblocking function because the nature of MANET routing and generic IP routing is to forward IP traffic on the behalf of other source nodes. Blocking a particular IP source address is incorrect behavior and would disrupt the emulation of the IP forwarding capability.

IV. Research Methodology of Random Topology Generator build in our Ad hoc Network Test Bed

We effort to create a series of random topologies. After setting various options, several random topologies are generated by the software module developed. Its purpose is to generate random network topologies, display it to the user using Graphviz tool and to directly apply them (Through sockets at the wired interface) to simulated LAN. Thus the hardware using for the tool is a host-server machine with two network interfaces that will have direct connectivity to the LAN that will be changing the topology and all the client members to that LAN to have two network interfaces.

Therefore users can automatically generate arbitrary logical network topology in order to perform real time performance measurements of routing protocol implementations. By changing the logical topology of the network, users can conduct test on adhoc network without having to physically move the nodes in the adhoc network. Given the number of nodes in adhoc network Test Bed, each node's IP and MAC address, software module used in a test Bed creates arbitrarily connected graphs and updates each node's IP_TABLES accordingly through socket servers running on each network node in order to reflect the new logical topology. Thus arbitrary graph is represented in an adjacency matrix that is then translated into the corresponding IP_TABLES. Software module uses open source graph visualization tool Graphviz [16] to display the logical topology of the adhoc network**.**

a) Design Parameters for logical Topology creation scenario module

- **Scenario name**:

  It is the name of the scenario to be created by the user in a test bed.

- **Number of nodes:** How many nodes the generated topologies will have. Of course this should be less or any topologies this scenario will have.

- **Maximum node degree per node:**

  The degree of node is how many connections it has with other nodes. When generating random topologies a node will have a random number of connections with other nodes from zero to the value we have set here.

- **Network Density (0-100):**

  How dense the network will be. That is determined by an adjacency matrix, where element every element (i,j) determines if there is a connection between node i and node j. The number we have set here is the possibility to have a'1' in each position (i,j), therefore, has connection. And '0' i.e default value search for a node that have a neighbor according to the adjacency matrix. We take a starting node of the adjacency matrix that we have created and then based on that try to see if the adjacency matrixes that we have produced give us a connected graph. To the best of our knowledge, all currently known topology control algorithms have in common that every node establishes a connection to at least its nearest neighbor.That here we discover a starting node and then, based on the neighbours, we try to include all the nodes of the adjacency matrix.

We assume Boolean connectivity between the nodes. This means that two nodes are either connected or they are not.

We set status value as

99 -> the adjacency matrix give us an over connected or disconnected topology

100-> the adjacency matrix give us a suitable topology

Therefore the adjacency matrix is checked to meet our constraints of maximum node degree {1-4} after this point the status value should be 99 (not connected) or 100 (connected)

Thus, this module allows users to save and replay different mobility scenarios, to control the maximum and minimum node degree, produces an output in the form of adjacency matrix for further analysis and provides a framework for building additional ad hoc network testing tools.

b) Implementation and Results

After successful designing parameter for logical topology and adopting the research methodology for random topology generator developed in our test bed, we formulate the following results in brief.

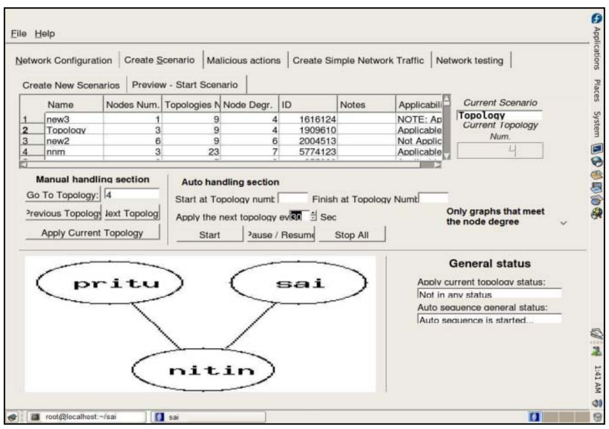

Figure 4: arbitrary creation of logical topology(Node degree = 4)

The "Fig. 4" indicates arbitrary creation of logical topology of physical three nodes in laboratory with manual handling setting to topology number 4 of current scenario 'Topology'

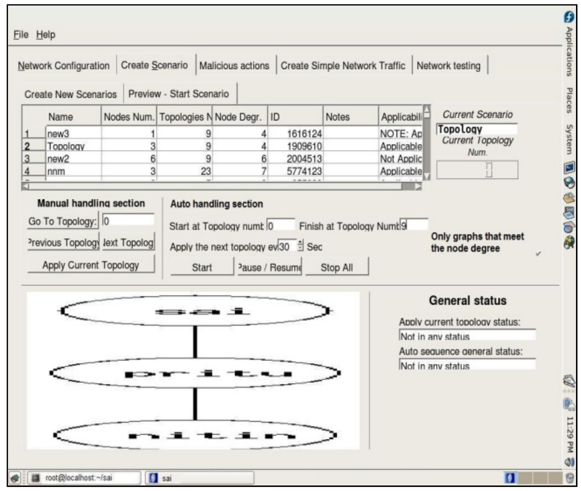

Fig 5: Logical Topology Creation (Node degree = 4)

Above indicates arbitrary creation of logical topology of three nodes automatically after every 30 seconds for setting to topology number 0 to 9 for the same current scenario 'Topology'.

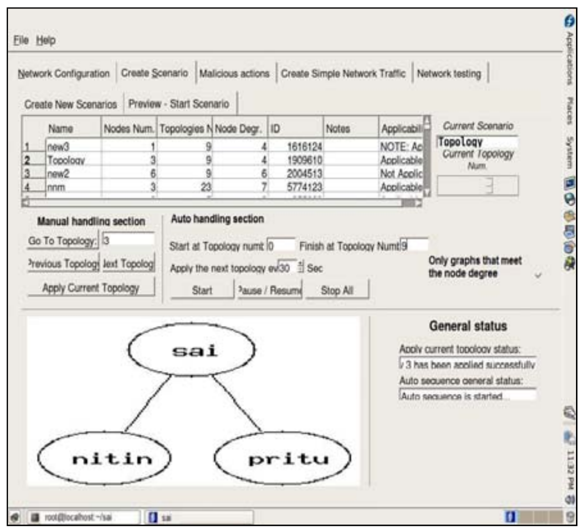

Figure 6: Logical Topology Creation (Node degree = 4)

"Fig.6" shows again arbitrary creation of logical topology of actual three nodes in every 30 seconds for setting to topology number 0 to 9 for the same current scenario 'Topology'.

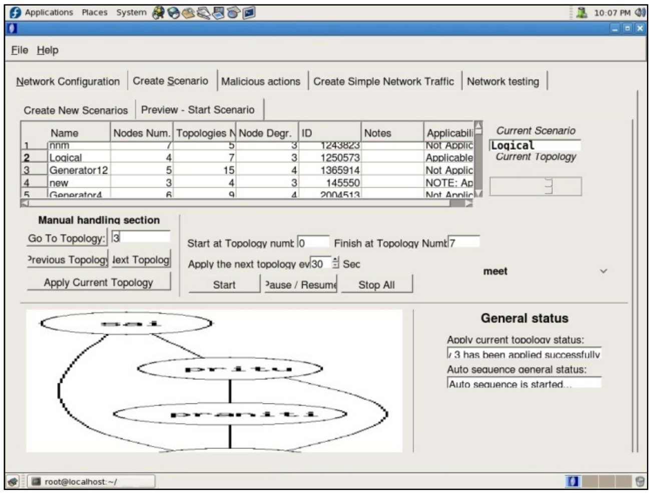

Figure 7: Logical Topology Creation (Node degree =3)

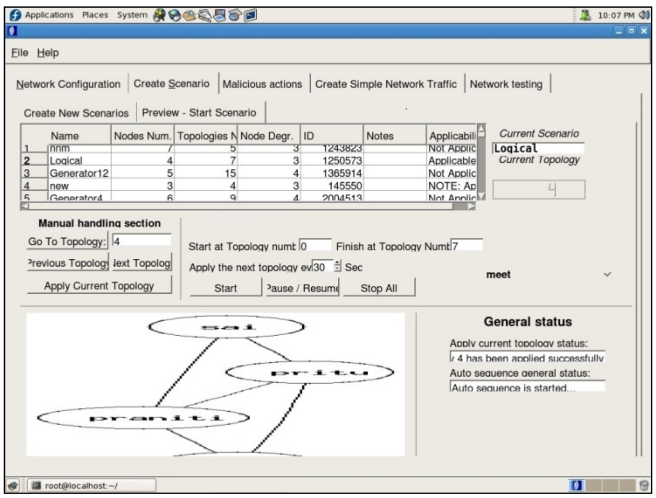

Figure 8: Logical Topology Creation (Node degree =3)

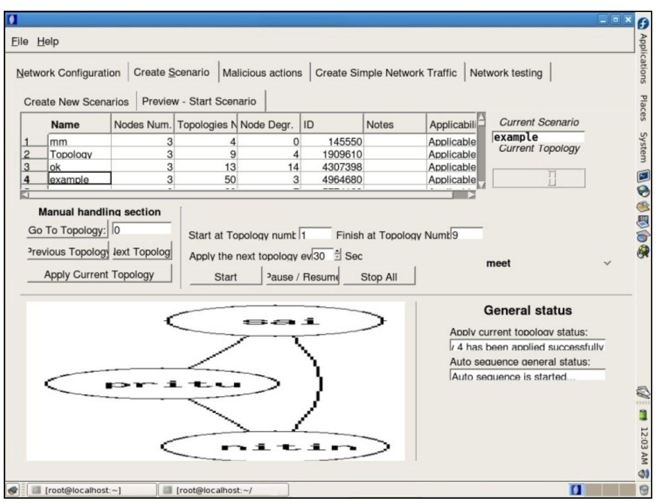

Figure 9: Logical Topology Creation (Node degree =3)

Thus, various random topologies can be generated for actual physical nodes present in a laboratory. Every random topology can be reflected with the random number we have selected in a software module. This various random topologies will produce for the maximum node degree 4. (from node degree 1 to 4) which we have set in the development of the module. Similarly, the arbitrary topology will not be created for the number of nodes more than the actual nodes physical available in. the research laboratory. Basically, we can set 150 physical nodes and theoretically, there is no limit to the number of nodes this test bed can handle.

V. CONCLUSION

Although we are able to control each node's logical perception of the network topology, we cannot control the network congestion resulting from having a large number of adhoc nodes in a close proximity in a relatively small research laboratory. But this issue will not prevent budget-conscious researchers from collecting useful results a modest size network. In fact, determining the global network topology in a mobile adhoc network given the time delays of the diagnostic packets and the mobility of the nodes make this task very difficult, but determining an approximation of this topology or subset of this topology, within certain time frame may be useful.

This paper implicates the research methodology we have adopted for random topology generator to be build in our adhoc network test bed, Here, we successfully demonstrated the design of the module which allows users to save and replay different mobility scenarios to control the maximum and minimum degree, produces an output in the form of adjacency matrix for further analysis and produces a framework for building additional adhoc network tools. . An approximation of the network topology can provide the useful information about network density, network mobility, critical path and critical nodes.Thus ,we may conclude if global topology of the adhoc network is known, researchers can benchmark the actual performance of their adhoc routing algorithms and applications against the theoretical optimal performance.

REFERENCES

[1]. Imrich Chlamtac, Marco Conti, and Jennifer J. –N. Liu. Mobile Ad-hoc Networking: Imperatives and Challenges. *Elsevier Ad Hoc Networks Journal.* 2003. Vol. 1(1): 13-64.

[2]. Qi Xue and Aura Ganz. Adaptive Routing in Ubiquitous Mobile Access Networks. *Proceedings of IEEE VTC.* 6-9 October, 2003. Orlando, Florida: IEEE. 2003. 3070-6074.

[3.] N. Milanovic, M. Malek, A. Davidson, V. Milutinovic. Routing and Security in Mobile Ad hoc Network. *IEEE Computer Journal.* 2004. 37(2): 61-65.

[4]. Jun-Zhao Sun. Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing. *Proceedings of IEEE ICII 2001*. 29 Oct.-1 Nov 2001.Beijing: IEEE. 2001. 316-321.

[5]. Farid Dowla. Mobile Ad Hoc Networks. In: *Handbook of RF and Wireless Technologies*. Elsevier, 2004.

[6]. D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, C. Elliott. Experimental Evaluation of Wireless Simulation Assumptions. *Proceedings of 7th ACM Internaltional Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems.* Oct. 2004. Italy: ACM. 2004. 78-81.

[7]. S. R. Das, R. Castaneda, and J. Yan. Simulation Based Performance Evaluation of Mobile Ad-hoc Network Routing Protocols. *Mobile Networks and Applications, Kluwer.* 2000. 5(3):179-189.

[8.] S. R. Das, C. Perkins, and E. M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks. *Proceedings of IEEE INFOCOM 2000.* 26-30 March, 2000. Israel: IEEE. 2000. 3-12.

[9.] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson. The Effects of Ondemand Behavior in Routing Protocols for Multi-hop Wireless Ad hoc Networks. *IEEE Journal on Selected Areas in Communications Special Issue on Mobile and Wireless Networks.* 1999. 17(8): 439-1453.160

[10]. E. M. Royer and C.-K. Toh. A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks. *IEEE Personal Communications.* 1999.6(2):46-55.

[11]. Kwan-Wu Chin, John Judge, Aidan Williams and Roger Kermode.Implementation Experience with MANET Routing Protocols. *ACM SIGCOMM Computer Communications Review*. 2002. 32(5): 49-59.

[12] Nitiket N Mhala and N K Choudhari, "An Exploration of Ad-hoc Network in a Real World builds in a Laboratory Environment", *IJCNS Vol. 2, No. 8, August 2010* ,pp 61-66

[13] Graphs: Theory and Algorithm, Thuasiraman, M.N.S Swamy.

[14] The related information is available on http://xw2k.nist.gov/dads/HTML/adjacencyMatrixRep.html

[15] The related information is available on http://linux.die.net/man/8/**iptables**

[16] The opensource graph visualization software is available on http://www.graphviz.com

AUTHORS PROFILE

Mr. Nitiket N. Mhala is PhD student and also working as Associatet Professor and Head in the Department of Electronic Engineering, Sevagram, India. He received his ME Degree from RM Institute of Research and Technology, Badnera, Amravati University and BE Degree from Govt. College of Engineering, Amravati, Amravati University. He published a Book Entitled PC Architecture and Maintenance. He is a member of Institute of Electronics and Telecommunication Engineer (IETE). His area of interest spans Data communication, Computer network and Wireless Ad hoc networks. He published research papers at National and International level.

Dr. N. K. Choudhari is a Professor and completed his Ph.D degree in Electronics Engineering from J.M.I., New Delhi and received his M.Tech in Electronics Engineering from Visveswaraya regional Engineering College, Nagpur. He received his BE in Power Electronics from B.D.C.O.E., Sevagram. Presently he is Principal at Smt.Bhagwati Chaturvedi COE, Nagpur, India. He is guiding few research scholars for persuing Ph.D degree in RTM Nagpur University, Nagpur, India. He has worked as members of different advisory committees and was a member of Board of Studies Electronics Engg. of RTM Nagpur University, Nagpur, India