Simulation of A Novel Scalable Group Key Management Protocol

Amrutasagar Kavarthapu
Gudlavalleru Engineering College
Gudlavalleru 521356
Andhra Pradesh, India

Aswini Kavarthapu
QIS College of Engineering and Tecnology
Ongole 523002
Andhra Pradesh, India

ABSTRACT
Secure and multicast group communication is an active area of research. The main problem in secure group communication is group dynamics and key management. Group key management brings challenges on scalability for multicast security. Member joining and member leaving from the group is the main challenge in designing secure and scalable group communication for dynamic update of keys. Most of the proposed solutions are not considering this parameter and so suffer from the one-affects-n scalability problem. This paper presents the simulation of our approach A Novel Scalable Group Key Management Protocol (NSGKMP) and gives the brief introduction about NetworkSimulator2 (NS2). This NSGKMP approach decreases number of rekeying operation when a member joins in to the group or leave from the group i.e. dynamic updating of keys.

General Terms
Network security, simulation.

Keywords
Group communications, group key management, key, multicast security and scalability, Simulation, trace, nam.

1. INTRODUCTION
Multicasting is nothing but delivery of messages to a group of destination computers simultaneously in a single transmission from a source. Now a day's Multicast applications have grown and greatly influenced our life along with the growth of the Internet. Examples of such applications include information services, distributed interactive simulation and teleconference. As an important and mandatory building block for multicast applications, multicast security has been extensively researched in the past decades for protecting multicast communications. The research on multicast security addresses authentication, confidentiality, and access control, among other areas, where group key management is a key component. However, scalability is still a hard problem and a sizable challenge for group key management technologies. To make sure that group communication confidentiality, a group key management protocol must create and distribute a symmetric encryption key called traffic encryption key *(TEK)* or group key. In addition to the group key secrecy, the group key management protocol must provide forward secrecy and backward secrecy. Forward secrecy prevents an accessing current communication by old member after it leaves from the group. Backward secrecy prevents an accessing of the communication sent before a new member joins to the group. To do so, a re-keying process should be performed after every join or leaving a member from the secure group. It consists in generating a new *TEK* and distributing it to all group members. The main problem with any re-keying technique is scalability: as the re-keying process should be performed after every member join or leave from the group, the computational and communication overhead induced may be important in case of frequent join and leave operation to group.

The previous approach [1] is concentrated on to decrease number of re-keying operations, i.e. from log_2^n to log_m^n when compared with [2], where n is the number of leaf nodes of tree and m is the order of the B-Tree. Here we are taken order of B-Tree is 3(i.e. m=3).

This paper presents brief introduction to NetworkSimulator2 (NS2) and simulation of our previous approach [1] with the help of NetworkSimulator2 (NS2).

The remainder of the paper is organized as follows. Section 2 present Introduction to NS2, Section 3 presents Packet Animation. Section 4 presents our protocol, Section 5 presents simulation results, Section 6 presents performance of the protocol and Section 7 gives the conclusion.

2. INTRODUCTION TO NS2
Network simulator 2 (NS2) is an open source discrete event simulation tool used for simulating Internet protocol (IP) networks. The NS2 software uses TCL (Tool Command Language) as a front-end interpreter and C++ as the back end network simulation engine. Network simulation scripts in TCL are used to create the network scenarios and upon the completion of the simulation, trace files that capture events occurring in the network are produced.

Network protocol designers countenance many complex tasks, including simultaneously observing state in a potentially large number of nodes, understanding and analyzing complex information exchanges, and characterizing dynamic communications with competing traffic. Traditionally they have used packet traces to achieve these tasks, but traces have two major drawbacks. They present an implausible amount of detail, which challenges the designer's ability to comprehend the data, and they are static, which hides an important dimension of protocol behavior. As a result, detailed analysis frequently becomes tedious and error-prone. Although some network simulators can easily generate frequent detailed traces, they provide limited help for analyzing and understanding the data. But NAM, the Network AniMator provides packet- level animation, protocol graphs, traditional time-event plots of protocol actions, and scenario editing capabilities.

Simulation is the Process of designing a model of a reality system and conducting experiments with this model for the purpose of understanding the system or evaluating various strategies for the operation of the system.

We can question our self why the Simulation? Because Real-system not available, is complex/costly or dangerous (e.g. space simulations, flight simulations) and quickly evaluate design alternatives (e.g. different system configurations)

3. PACKET ANIMATION

NAM's core is packet animation. Animation permits the user to promptly see the status of each part of the network. NAM lets users change the animation speed and play it forward or backward, fast forward, fast backward, making it easy to find and inspect appealing occurrences. The first step in a new animation is displaying the network topology. NAM has three different topology layout mechanisms to accommodate different needs, those are as follows.

3.1 Automatic

Automatic layout is a default layout. This automatic layout can produce reasonable layouts of many networks without involvements of user guidance, but it may or may not produce adequate results for complex networks. If necessary, users can manually adjust the resulting layout.

3.2 Relative

Frequently relative layout is used for smaller topologies. Here the user specifies the relative directions of links, i.e. left, up, down, left-up, left-down, right-up, right-down and NAM uses the directions to place nodes relative to each other. NAM sets link length proportional to bandwidth and delay. Relative layout works well for small topologies and has the advantage that packet movement rate is stable and consistent with link delay and bandwidth. This relative layout's have some disadvantages, those are the user must specify the directions of each link and relative layout of a topology containing different delays can result in very short links.

3.3 Wireless

Wireless layout associates each node with a physical location in a constrained area. Each node's 3D coordinate gives its position in the area and its movement position. Wireless visualizations typically lack explicit links.

4. SCALABILITY OF GROUP KEY MANAGEMENT PROTOCOL

In [1] B-Tree structure was used to form a group communication. In B-Tree of order m, each node contains at most *m-1* elements and each node contains at least $\lceil m/2 \rceil - 1$ elements. The figure1describes that a B-Tree of order 3, so each node consists of at most 2 elements. Here [1] maintaining users of the group at leaf nodes of B-Tree. So each leaf node consists of 2 users. i.e. *node21* consists of user's $u_1$and u_2, *node22* consists of user's u_3 and u_4 ……. n*ode29* consists of user's u_{17} and u_{18}.

In the figure1 user u_{18} wants to join into the group. After joining of u_{18} in to the group to provide backward secrecy we are changing the parent's key from leaf to the root node that was indicated in the figure1. After changing the parent's keys the changed keys are sent to the corresponding children's that was shown in the simulation.

5. SIMULATION RESULTS

Here the B-Tree was constructed using NS2 simulator. According to B-Tree property each node consists of two elements. In the figure 2, hexagon shape node pairs are treated as root node. i.e. (1) and (2). The square shape node pairs i.e. (4) and (5), (6) and (7), (8) and (9) are treated as intermediate nodes of tree. At leaf circle shape node pairs are (16) and (17), (18) and (19), …….. (36) and (37) are treated as users of the group.

When compare to figure1 and figure2, in figurte2 hexagon shape node pair (0) and (1) treated as root node of B-Tree in figure1. i.e. *node01*. Rectangle shape node pairs in figurte2, i.e. (4) and (5), (6) and (7), (8) and (9) are treated as *node11, node12, node13* in figure1 respectively. At leaf circle shape node pairs in figure2, i.e. (16) and (17), (18) and (19), (36) and (37) are treated as *node21,node22*, ……., *node29* in figure1 respectively.

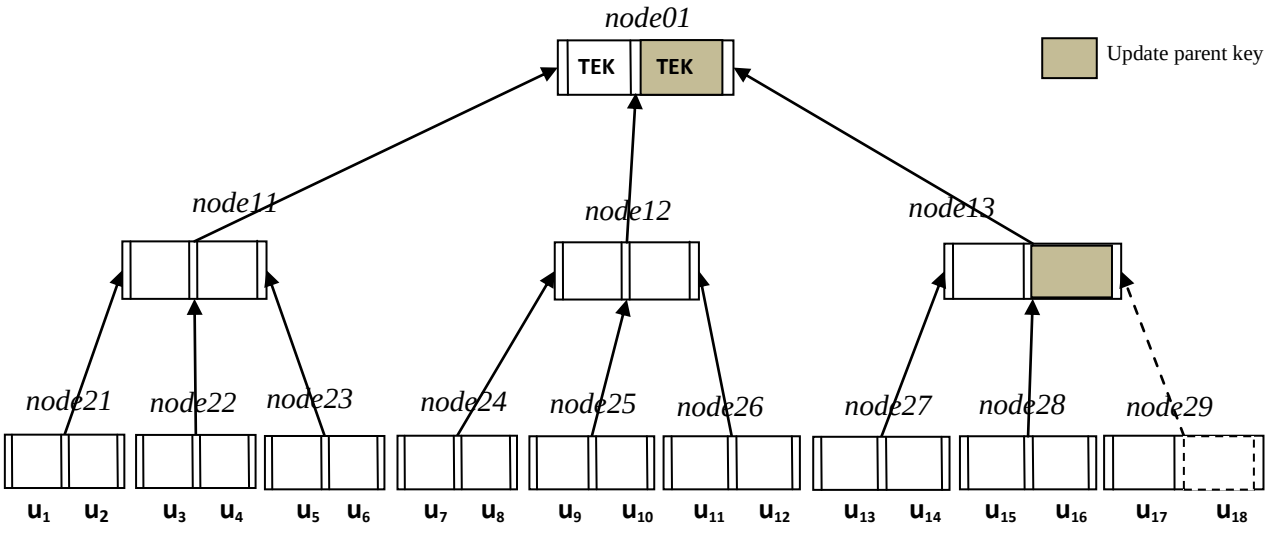

Fig 1: Group Structure in B-Tree format

The following Figure 2 showing the group in B-Tree format of order 3. According to B-Tree if *m* is the order of the B-Tree then we can place *m-1* elements at each node.

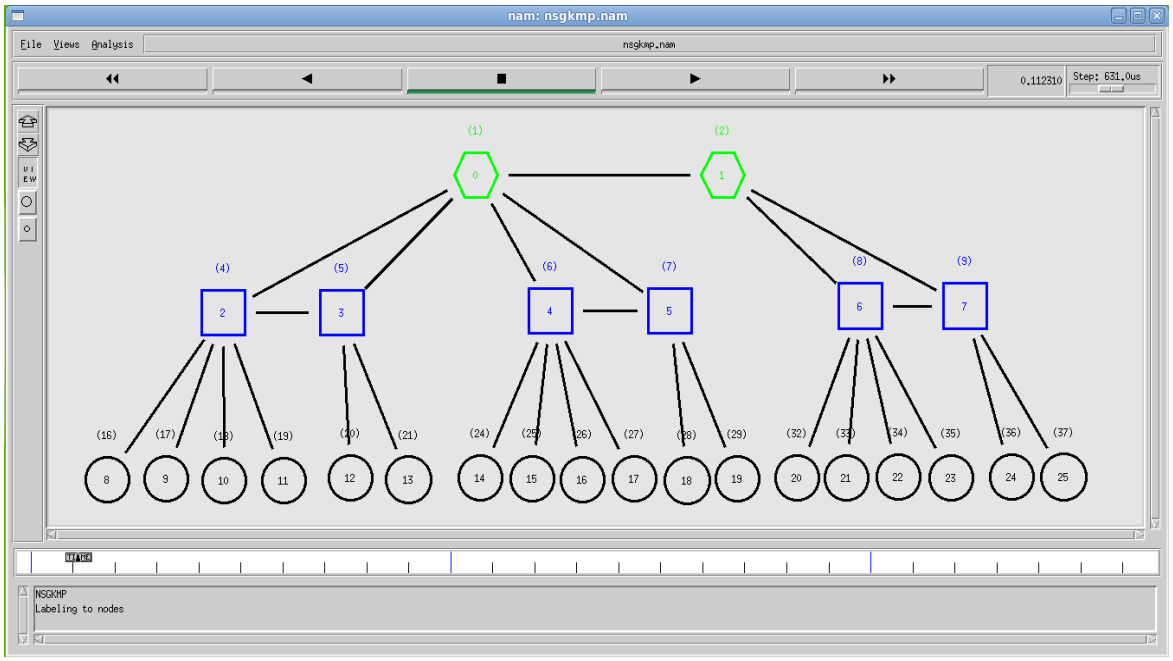

Fig 2: Group Structure in B-Tree format and Labeling to the nodes

In the following Figure3 user16 want to join to the group that is indicating by orange color.

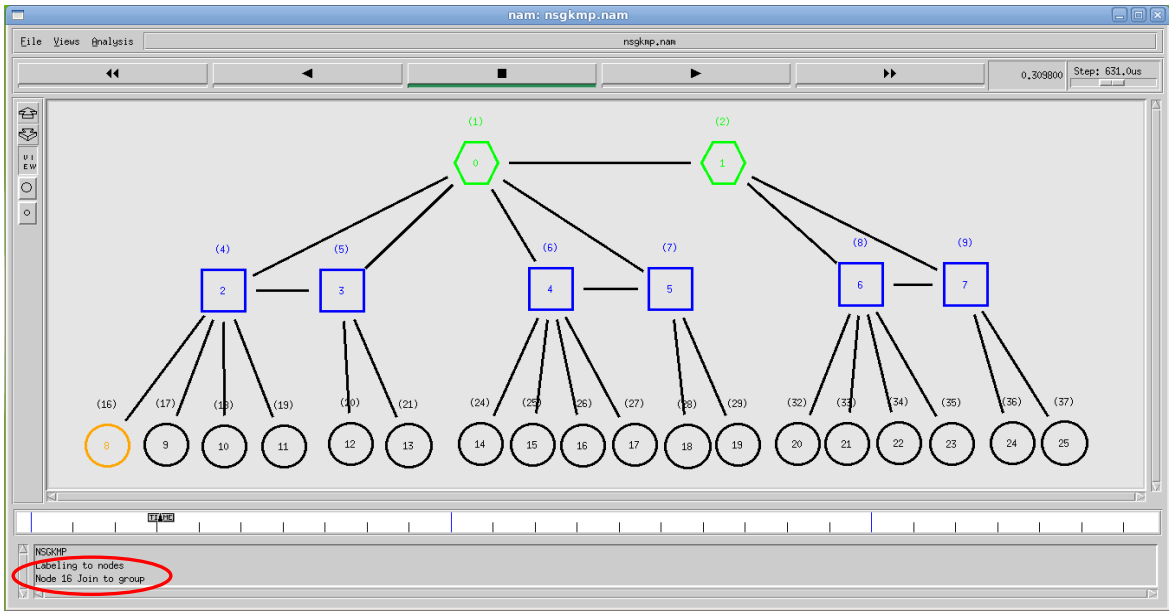

Fig 3: User 16 wants to join to the group

After user16 join in to the group his parent i.e. (4) was changed. After changing the parent's key, the changed key was sent to the corresponding children's, i.e. (16), (17), (18) and (19) as shown in the figure4.

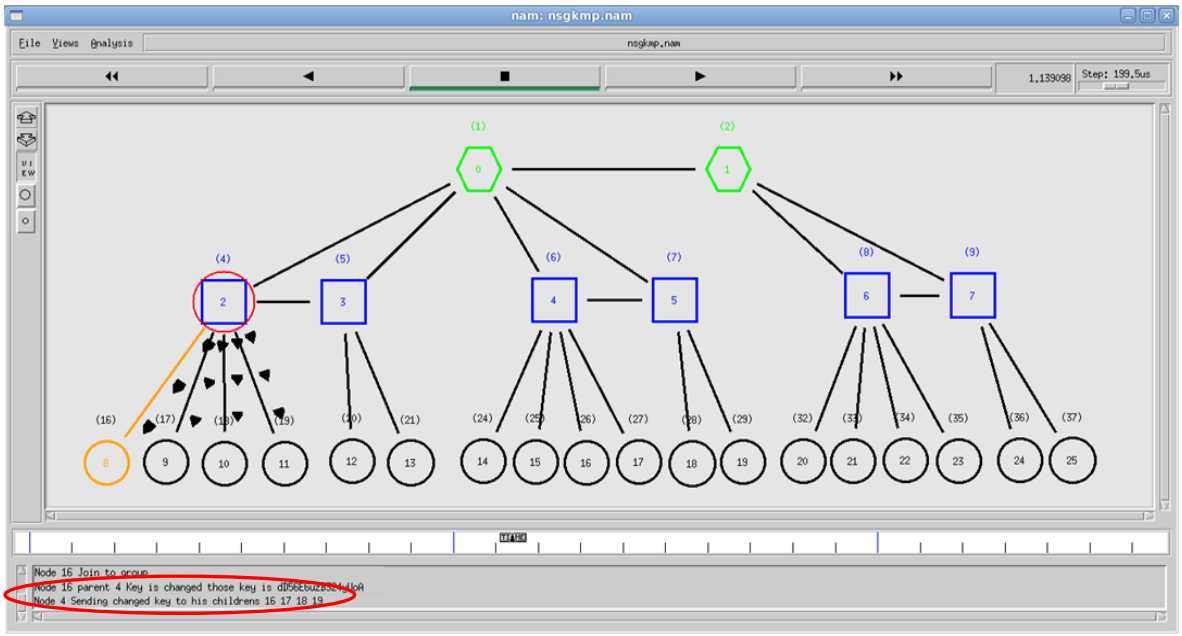

Fig 4: Node 4 sending changed key to its children's 16,17,18,19

After changing the key of (4), his parent (1) key also changed, so the changed key was sent to his children's (4), (5), (6) and (7), that was shown in the figure5.

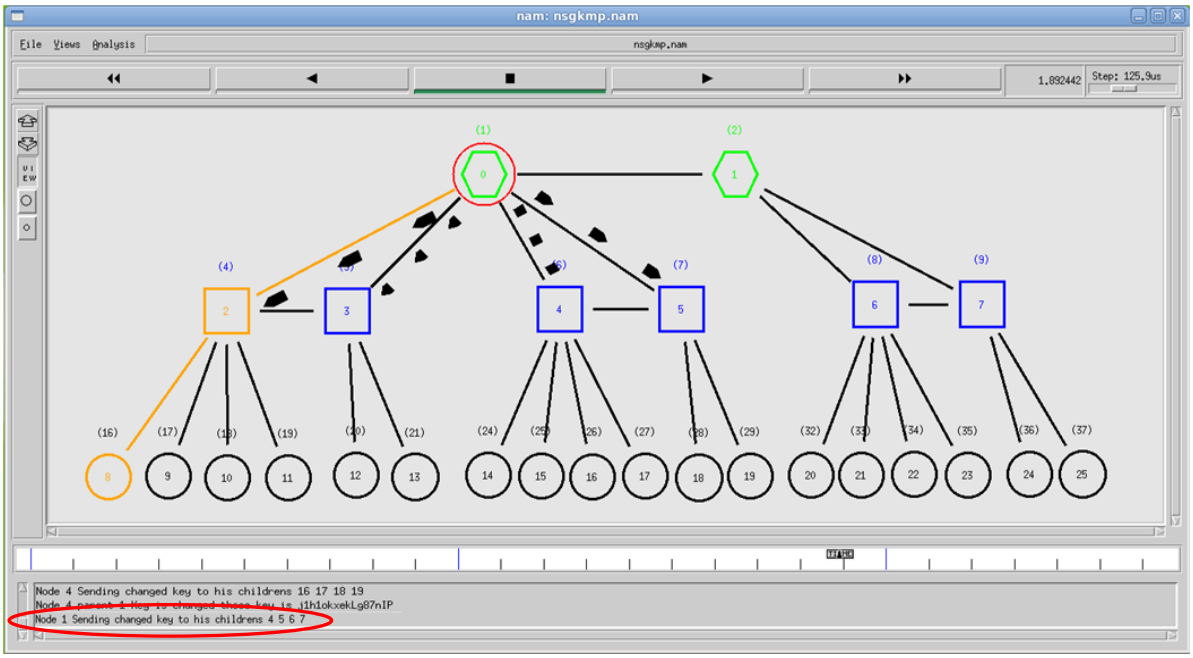

Fig 5: Node 1 sending changed key to its children's 4,5,6,7

Finally the number of re-keying operation from leaf to the root node indicated by gray color hexagon shown in the figure6.

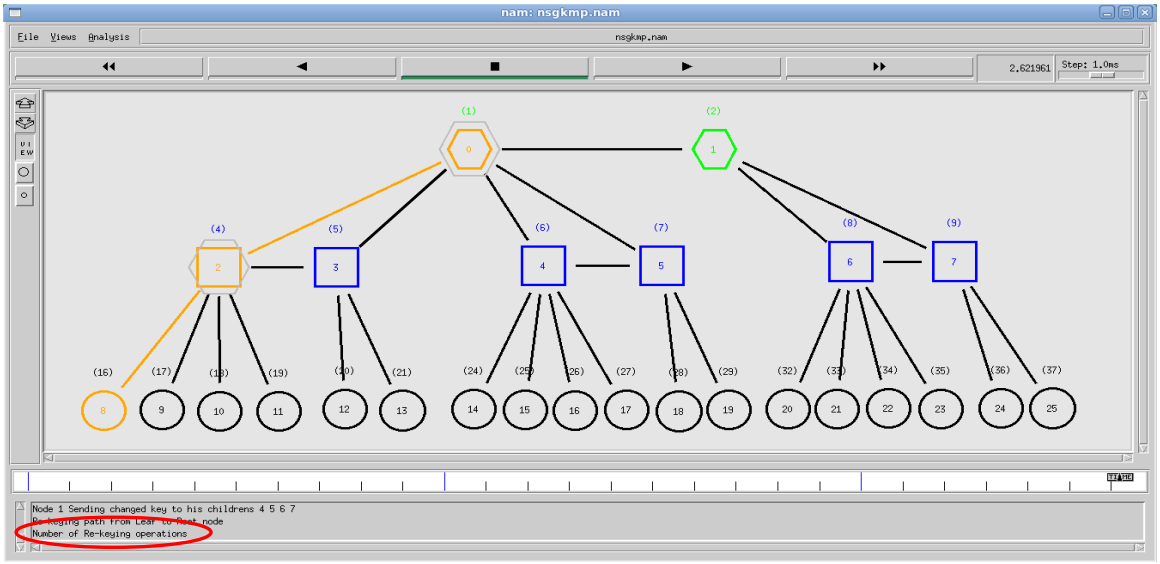

Fig 6: Re-keying operations performed at nodes 4, 1

6. PERFORMANCE OF THE NEW PROTOCOL

This session provides overview of simulation model and some of the results by comparing [1] with the protocol of [2]. Table1 give the comparison between [1] and [2]. From **Figure7** NSGKMP [2] has the less number of Re-keying operations when a member joins or leave from the group.

Table 1. A comparison of NSGKMP with SGKMP

Scalability metrics	Number of Re-keying operations	
Protocols	J	L
NSGKMP	log_3^n	
SGKMP	log_2^n	

J: Join; L: Leave

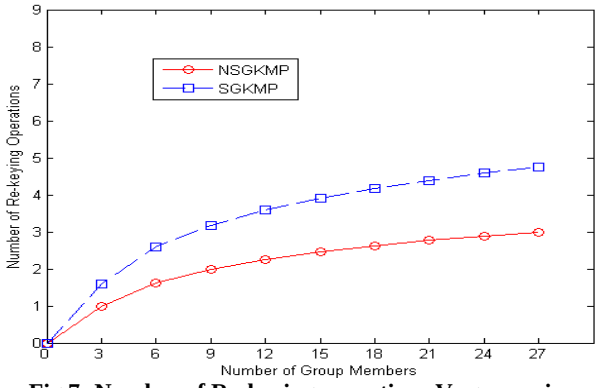

Fig 7: Number of Re-keying operations Vs group size

7. CONCLUSION

To improve the scalability of the Group key management we proposed A Novel Scalability Group Key Management Protocol and demonstrated that it has better scalability in terms of number of Re-keying operations. The process of entire approach was simulated in network simulator and presented the simulation results. Finally we conclude that if we increase the order of B-Tree then automatically we can decrease the number of re-keying operations further more.

8. REFERENCES

[1] Amrutasagar Kavarthapu and Seshagirirao Ganta, "A Novel Scalable Group Key Management Protocol (**NSGKMP**)", International Journal of Computer Applications, VOL. 39, NO 7, pp. 535-538, February 2012.

[2] Ronggong Song and George O. M. Yee, "A Scalable Group Key Management Protocol (**SGKMP**)", IEEE Communication Letters, VOL. 12, NO. 7, pp.541-543, July 2008.

[3] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: issues and architecture," National Security Agency, RFC 2627, June 1999.

[4] R. Varalakshmi and Dr. V. Rhymend Uthariaraj, "A New Secure Multicast Group Key Management (NSMGKM)Using Gray Code" IEEE –International Conference on Recent Trends in Information Technology, ICRTIT 2011, Anna University, june 2011.

[5] G. H. Chiou and W. T. Chen, "Secure broadcast using secure lock," *IEEETrans. Software Engineering*, vol. 15, no. 8, pp. 929–934, Aug. 1989.

[6] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) architecture", RFC 2093, July 1997.

[7] H. Harney and C. Muckenhirn, "Group Key Management Protocol (GKMP) specification", RFC 2094, July 1997.

[8] D.SAMANTHA, Classic Data Structures, Prentice-Hall of India Private Limited, New Delhi-110001, 2006.

[9] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne and Ya Xu Haobo Yu, "Network Visualization with Nam, the VINT Network Animator" R E S E A R C H F E A T U R E, IEEE, pp. 63-68, November 2010.

[10] Aliff Umair Salleh, Zulkifli Ishak , Norashidah Md. Din and Md Zaini Jamaludin, "Trace Analyzer for NS-2" , 4th Student Conference on Research and Development (SCOReD 2006), Shah Alam, Selangor, MALAYSIA, pp. 29-32 , June-2006.