Introducing Privacy in On-Demand Multicast Routing Protocol

Er. Amrita Chaudhary [1]
Assistant Professor, CSE Dept.
Chandigarh University
Mohali, India

Er. Puneet Kaur [2]
Assistant Professor, CSE Dept.
Chandigarh University
Mohali, India

***Abstract*: We have studied the various problems occur in multicast routing protocol. The exchange of link state tables or routing vector, generated by continuous topology changes, processing overhead and yields excessive control. Further, periods of routing table instability lead to instability of multicast tree, which in turn results in increased buffering time for packets, higher packet losses, and an increase in the no. of retransmissions. Therefore, using Java language or NS2 simulator, we have to introduce privacy in ODMRP, so that this multicast routing protocol performs efficiently in adhoc network.**

Keywords: Multicast Routing protocol, MANET, NS-2 Simulator.

I. INTRODUCTION

Connections of large number of mobile hosts and wireless links made “Ad-hoc Networks”. Each mobile node (MN) operates not only as an end-system, but also as a router to forward packets over the multi-hop ad-hoc networks. This n/w topology is dynamic, that’s why it might show some frequent changes due to the nodes’ movements. Collection of various mobile nodes made adhoc n/w, which form network temporarily without the support of standard support services or centralized administration available on conventional n/w, regularly [1]. The network created to transfer data from one end to another end of the computer is Ad-hoc network.

To add privacy in Multicast routing Protocol, with cryptographic techniques, a secure multicast routing protocol was designed, which has quality impact on the routing performance. Assuming, couple of nodes are willingly communicate one another, which are outside the range of wireless transmission, nonetheless, they’ll be capable to communicate, although other nodes are willingly forward the data packets from them. Ad-hoc network applications includes educational and commercial use in remote areas, emergency relief operations, military tactical communication, & also in meetings and other situations [2].

To encrypt and decrypt the multicast data, each and every member must hold a key in multicast secure communication. Whenever a member wants to join or leave the group, a key first updated and then distributed/shared with all members of the group. The process of updating and distributing the key amongst group members is known as rekeying operation [3].

II USED IN SIMULATION PROTOCOL

A) ON-DEMAND MULTICAST ROUTING PROTOCOL (ODMRP)

In wireless networking, on demand multicast routing protocol is a protocol used for routing multicast & unicast traffic which performs throughout wireless mesh network. On demand multicast routing protocol only creates routes on demand, rather than proactively creating them. However, there is a delay in route acquisition, but it helps to reduce the network traffic.

On demand multicast routing protocol depends upon the periodic flood for routing discovery & maintenance, and has a quite popular implementation. This is generally designed to ensure the robustness against mobility & not reliable on wireless line propagation [7].

On demand multicast routing protocol is an on-demand mesh-based protocol. It might also use of unicast tech. to transmit multicast data packets from sender to receiver in a multicast group.

Forwarding group concept is used to carry the multicast packet data through scoped flooding. The source is established in ODMRP [4]. At a point, when source has a data packets to send, member-soliciting packet has been periodically broadcast/declared, is known as Join Query. Non-duplicate Join Query packets are received, where upstream node address is saved by every node in a network, that is backward learning, in a routing table, is termed as Up-NodeID-Table, which saves the Join Query in message cache and the data packet has been rebroadcast into its neighbouring nodes, as shown in fig.1. Join table packet is created by the multicast receiver and broadcast it to its neighbours, when multicast receiver receives the Query packets.

Following the learned backward path, Join Table packets are relayed back to source. Reverse path nodes become forwarding groups and set the flags of their own forwarding group [8].

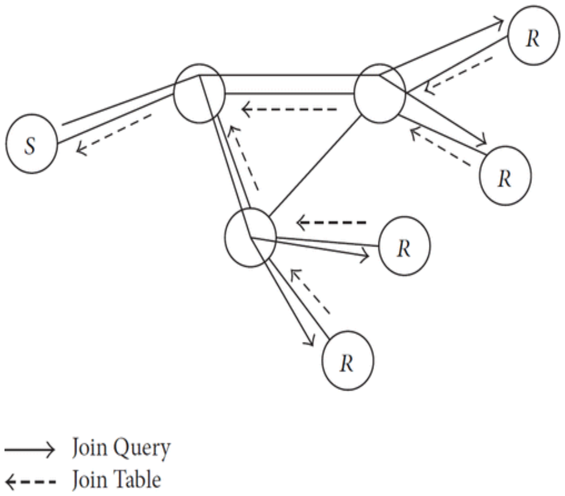

Fig1: Procedure of maintenance and membership setup on-demand [8].

Forwarding Group concept, is shown in the fig.- 2. Data is delivered by the forwarding group node. They rebroadcast the unique packet destined for the associated multicast group, thus, letting them to forward to the receiver-end.

Set of nodes make forwarding group which is responsible to propagate the multicast data, forming a mesh-structure between the senders & the receivers. Forwarding group shared by the multicast groups U & V.

Members of multicast group maintained by the soft-state approach in ODMRP; to join or leave the group, no explicit management message is required [8].

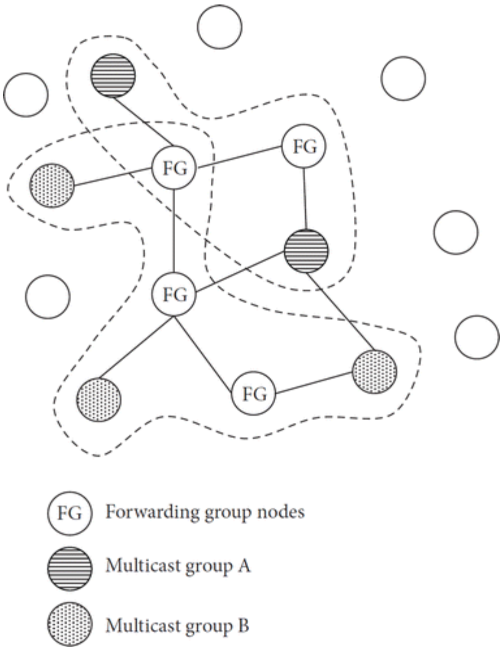

Fig 2: Forwarding group concept [8].

A) ALGORITHM OF DIFFIE HELLMAN

This algorithm is exponential key exchange algorithm i.e. securely exchanging cryptographic keys over a public channel and it is used to establish secret communications.

Assuming, four variables where one is prime R and S (Primitive roots of R), whereas u and v are private values.

- R and S are public numbers. Private values of u & v can be picked by the users (let X & Y). Users generate the keys & publicly exchange the keys over a public channel, on the other-end person receive the keys and using these keys, secret key is generated. After this process, they have the same secret key which is used to encrypt [6].

Table 1: ALOGRITHM OF DIFFIE HELLMAN [4]

X			Y		
Secret	**Public**	**Calculates**	**Sends**	**Calculates**	**Public**
U	r, s		r, s →		
U	r, s, U	s^u mod r=U	U →		r, s
U	r, s, U		← V	s^v mod r=V	r, s, U, V
u, t	r, s, U, V	V^v mod r= t		U^v mod r= t	r, s, U, V

C) *ALGORITHM USED:*

```
Algo(S, D)
/* Where S: Sender Node and R: Receiver Node */
{
    1) Discover the route between S and R.
    2) Generate the Key Group for each node using the algorithm of Diffie-Hellman.
        i. We have Unique Private Key (Upvk)
        ii. And Global Public Key (Gpuk)
        iii. Shared Key (Srdk)
    3) At S which is source node, we recover the Public key of Receiver R & the encryption is then performed.
            INFO: welcome
                Encrypted INFO:
              welcome + 1 = xfmdpnd
    /* "xfmdpnd" is encrypted data will proceed further to receiver side */
    4) Perform Sender receiver communication using encrypted information.
    5) Verify the shared key for each node in the route (Ri)
    6) Perform Decryption using Private Key (R) on receiver side.
          Decryption INFO: xfmdpnd - 1
          Decrypted INFO = welcome
    7) Dispose of bad packets originating from unauthorized nodes.
    8) Communicate safely over a network.
}
```

D) *NETWORK SIMULATOR NS-2:*

The Simulation process is a process of learning by doing, as we know. NS-2 tool in one amongst the many open-source simulation tools that runs on Linux OS. It's a Discrete-Event Simulation (DES) which is written in C++ and OTcl especially to research in computer communication networks. The substantial support provided by simulator for simulation of routing, IP protocols and multicast Protocols [7]. It's emerged as a good alternative that’s used for analysing the results of computer systems. Many features of NS-2 stimulator, make simulations possible. It not only supports generally used IP protocols but also allows the users to extend or generate/build their own protocols. The complete NS-2 source code is downloaded and then assembled for multiple platforms as Cygwin, UNIX & Windows [4].

E) RESULT ANALYSIS:

The mobile adhoc n/w includes 25 mobile nodes is built in the NS-2 simulator. The positions of these mobile nodes are well-defined as X and Y co-ordinates value, as Fig.3 shows. The specified situation shows the data packet source node transmission to destination node in ODMRP in MANET. Fig.3 shows the packet loss, Packet received, number of bytes transferred and packet delay.

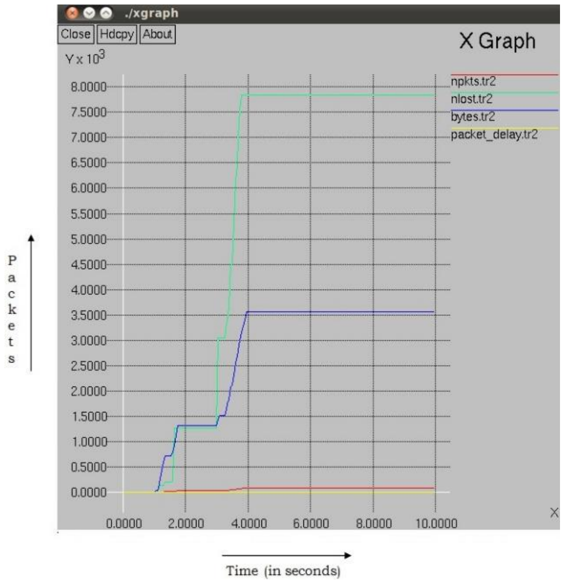

Fig.3: Packet delay, packet received, number of bytes transferred, packet loss, shown in graph X.

III. CONCLUSION

When the traffic loads are low, ODMRP performs well. But fails to uphold it throughput, when there is the increase in traffic loads. Because of multicast routing protocol, ODMRP, sometimes, suffers from scalability problems, as overhead in network is directly related to the traffic flows, the overhead increases if there is increase in the flow of traffic in network. Research Paper says, with the assistance of cryptographic algorithm, using NS2 simulator, introducing privacy in On-Demand multicast routing protocol (ODMRP) that helps ODMRP perform efficiently and securely. By doing this we only received limited amount of data packet and remaining data packets are lost or we say that only limited data packet reached securely.

REFERENCES

[1] https://www.cs.wmich.edu/wsn/doc/adhocrouting/AdhocRouting.pdf by SR Thampuran

[2] Krishna Paul, S. Bandyopadhyay, A. Mukherjee, D. Saha ,”A Stability-based On-Demand Multicast Routing in Ad-hoc Wireless Networks“(https://www.iimcal.ac.in/sites/all/files/sirg/11-8-routing-Stability-Based.PDF)

[3] J. Ye, W.C. Wong and K.C. Chua, “Power-Efficient Multicasting in Ad Hoc Networks” , IEEE 2003

[4] Rajneesh Gujral, Amrita Chaudhary, “Study and Comparison of Mesh and Tree-Based Multicast Routing Protocols for MANETs”, IJLTET, Vol. 1 Issue 2 July 2012

[5] https://www.researchgate.net/publication/290771797_A_Survey_of_Multicast_Routing_Protocols_in_Ad-Hoc_Networks

[6] https://www.geeksforgeeks.org/implementation-diffie-hellman-algorithm/

[7] https://www.linuxjournal.com/article/5929

[8] https://journals.sagepub.com/doi/full/10.1155/2015/652572

[9] https://www.researchgate.net/publication/300366833_Secure_Multicast_Routing_Algorithm_for_Wireless_Mesh_Networks.

[10] Hasnaa MOUSTAFA and Houda LABIOD,” A Performance Comparison of Multicast Routing Protocols In Ad hoc Networks”

[11] https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

[12] https://www.sciencedirect.com/topics/engineering/secret-key