SeReRoM: Secured Reliable Routing Scheme for Multicasting

R. Srinivasan, V. Vaidehi, K. N. Srivathsan, L. Ramesh Babu, and C. Karunagaran
(Corresponding author: R. Srinivasan)

Department of Electronics Engineering, MIT Campus of Anna University
Chennai 600 044, India (Email: srini0402@yahoo.com)
(Received Nov. 14, 2005; revised and accepted Dec. 31, 2005 & Jan. 27, 2006)

Abstract

This paper proposes a multipath routing scheme SeReRoM for a multicast group communication with a single source and multiple destinations. SeReRoM provides an environment that tackles the twin issues of security and reliability. In SeReRoM the message to be sent is divided into 'k' packets. An error correcting scheme used to provide reliability converts the 'k' packets into 'n' packets (n > k) and the 'n' packets are transmitted through 'n' node disjoint multicast distribution trees to reach all the destinations in the group. The reception of 'k' packets out of 'n' packets will enable the destination user to recover the original message. Thus failure of (n-k) distribution tree will not affect the regular communication in the group. Any hacker listening to one single tree will not be able to retrieve the entire portion of the message from the source node, thus providing security in addition to the secure key management scheme that exist for the multicast group. The functionalities of the proposed routing scheme are verified and the performance results of the proposed multicasting scheme are presented in this paper.

Keywords: Diversity coding, multicast, reliability, routing, security

1 Introduction

With the growth and commercialization of the Internet, simultaneous transmission of data to multiple receivers becomes a prevalent mode of communication. To avoid having to send the data separately to each receiver, several multicast routing protocols [3, 4, 6, 13] have been proposed and deployed. The underlying principle of multicast communication is that each data packet sent from the source reaches a number of receivers belonging to a group.

Today, most multicast routing algorithms that exist provide either security to the message by means of cryptography or reliability by means of sufficient retransmission of message packets. The scheme proposed in [12] explains multipath routing with security and reliability but it focuses only on unicast environment. Routing in multicast environment introduces a number of challenges [2, 5, 8] that are not encountered in unicast communication. In this paper a multipath routing scheme is proposed for multicasting environment that takes into account both reliability and security.

This scheme SeReRoM needs multiple multicast trees to be discovered from source to destinations of a group in order to provide security and reliability in a multicast environment. The multiple multicast trees are static until there is a major change in the group membership.

The operation of the proposed scheme is performed in three stages as follows:

1) Route Discovery;

2) Route Selection and Reservation;

3) Message Sending.

The functional operation of the proposed scheme is verified and the performance of the scheme is analyzed.

The rest of the paper is organized as follows. The overview of the scheme is presented in Section 2. The description of the proposed scheme is dealt in Section 3. The Section 4 provides the simulation results of SeReRoM. Concluding remarks are provided in Section 5.

2 SeReRoM: Overview

The processes of route discovery, route selection and reservation and message sending involved in the proposed scheme are explained in the following subsections.

2.1 Route Discover

Route discovery is a process that is initiated by the source node of the multicast group in order to discover all possible routes from source to different destinations.

Initially, the source generates a route request packet and sends it to its neighboring nodes. The neighbor can be a destination or an intermediate node. In case, the neighbor being an intermediate node updates the route request packet by augmenting its address at the end of the packet and increasing the packet length field. The intermediate node then forwards the updated packet to all its neighbors using selective flooding. Thus the request packets travel through all possible paths from source and reach the destinations. The destination on receiving a request packet mark an entry into the route cache table and then forwards it to all its neighbors. Thus the destination node also acts as an intermediate node in the route of some other destinations in the group. This process is continued until at least each of the destination nodes in the group has 'n' valid node disjoint paths or for a tolerable time duration.

Once the route discovery is over, the destinations will have all possible paths in its route cache. The route discovery process has to be repeated if there is a major change in topology or there is a real need for distribution tree change by the application.

```
Procedure - Route Discovery Begin
Begin
if (source)
  {
    Generate_routerequest( );
    For (i=1; i < noofpathsneeded; i++)
    Forward ( );
  }
else
  {
    Receive_routerequest ( );
    Appendaddress( );
    Forward ( );
    if (destination)
      {
        Receive_routerequest ( );
        Update_routecache ( );
        Forward ( );
      }
   }
end
```

2.2 Route Selection and Reservation

Route selection is a process by which node disjoint multicast distribution trees with minimum number of hops from source to destinations are selected. The request packets received at all the destinations that contain the route information are shared with a control entity called the Route Moderator (RM) through concast communication. The RM is responsible for selecting the 'n' node disjoint trees form the source to the destinations in the group. The route selection is done in such a way that the distribution tree used for sending a packet of a message is unique. If there are more than one node disjoint trees available, then the tree with minimum number of hops is selected by the RM. The selected node disjoint trees for each of the destinations are informed for path reservation.

The route reservation is a process of reserving a route for sending a particular packet to all the destinations in the group. This route reservation ensures that sufficient input and output buffers are reserved in the intermediate nodes to provide a certain level of reliability in terms of reduction in packet drop at the reserved intermediate nodes [7].

Route reservation is initiated by the destinations by sending the reply packet through the route selected by the RM to the source. The intermediate nodes reserve themselves for sending the particular packet.

```
Procedure - Route Selection and Reservation
Begin
if (destination)
  {
    For (i = 1; i <= noofpathsdiscovered; i++)
      {
        If (! node disjoint)
          Deletepath( )
        else
          Save_path ( )
      }
    If (no of paths for a packet >=n)
      {
        Forward_to_RM();
      }
      Form_replypacket( );
      Send_replypacket( );
  }
If(RM)
  {
    Select_node_disjoint ( );
    Return_node_disjoint();
  }
If (intermediate)
  {
    Receive_replypacket( );
    Reserve_resource( );
  }
If (source)
    Update_routecache( );
end
```

2.3 Message Sending

The source uses diversity coding scheme [1, 9] to encode the message packets to be sent to the destinations. The message is split up into 'k' packets and they are encoded into $n = k + x$ packets, where 'x' is the link reliability requirement factor. The encoded packet, when sent through 'n' node disjoint multicast trees to the destinations, the failure of any '$n - k$' trees from source to the destinations does not affect the recovery of the message.

The message packets are sent by the source node to the destinations through the reserved paths. The destinations

on receiving all the parts of a message packet through multiple paths, decodes them to retrieve the original message.

2.4 Security and Reliability

In this multicast communication there exist a secure key management scheme [11] that provides all the members of the group with a secret session encryption [10] that will be used to encrypt all the packets sent out from any node acting as a source of information. In addition to this level of security, the multipath routing scheme proposed provides security to the data packets as they are routed in a distributed way, thus making the probability of any unauthorized destination receiving 'k' required packets less. In this scheme the destination node requires packets from a minimum of 'k' trees to be received successfully in order to construct the message. As the node disjoint trees are distributed any external hacker wanting to know the entire message cannot get it by just hacking a single multicast distribution tree. Thus security is improved by providing packets of a message through multiple node disjoint trees SeReRoM.

As multiple packet distribution trees are established from source to the destinations in the group, if any one tree fails to deliver the packet to the destination, the packets from the other trees can be used to retrieve the message. Thus failure of any '$n-k$' distribution tree(s) does not affect message retrieval at the destination. The property of the proposed scheme has proved it to be more robust to link or tree failure, thus providing reliability in delivering the message to the intended group members.

3 SeReRoM: Description

The proposed SeReRoM is a multipath based multicast routing scheme that has evolved from the multipath routing scheme for unicast MuSeQoR [12]. In the proposed scheme any node function can be classified as source, intermediate or destination functions. These functions of each node are described in the following subsection.

3.1 Functions of Source Node

The source node performs various functions such as request packet generation, forwarding of request packets and then updation of its route request table.

Upon generation of the request packet at the source the process of route discovery is initiated. The structure of the request packet (RP) is as shown in Figure 1. The first byte of RP is the option type and when set as '1' indicates request packet. The next field in RP gives the length of the entire packet in terms of bits. The value in this field varies as the packet proceeds through the intermediate nodes. The third field gives the packet number of a message, this depends upon the number of node disjoint paths need to be established for the given network. The next field is reserved for future use. The field that follows is used to identify the source that generated the packet.

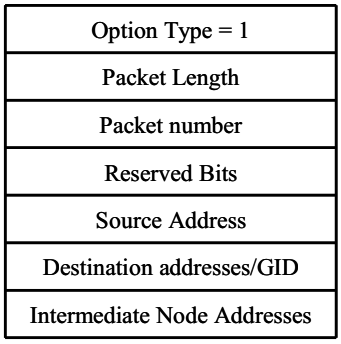

Figure 1: Request packet structure

The group ID of the multicast group is appended in the consecutive field. The last field in the packet is variable sized used to identify the neighboring intermediate node address through which the packet has to traverse.

After the generation of request packet, the source node has to forward the request packet to neighboring nodes. Each node maintains a 'configuration vector' that indicates its connection with other nodes. Depending upon its configuration vector, the source node forwards the request packet to its neighboring nodes. The number of node that it forwards depends upon the number of node disjoint routes needed. Thus the source does not forward the request packet to all neighbors and so the overhead involved is controlled.

The Route Request Table is present in all the nodes but will be used mainly in the source node. After the forwarding of the request packet, the source will update the route request table by adding its own address and the next intermediate node addresses. The route request table is updated only after the reception of acknowledgement packet by the source node. The intermediate node generates the acknowledgement packet after it receives the request packet. This route request table is used by the source after the route discovery for sending the message packets.

3.2 Functions of Intermediate Node

The function of the intermediate node is to send acknowledgement to the source for the request packet received, update the route request table and request packet and selectively flood the request packet to all its neighboring nodes.

The intermediate nodes that are immediately next to the source generate and send acknowledgement to the source upon receiving a route request packet. The intermediate node after sending the acknowledgement packet updates its route request table and the route request packet. The route request table is updated by adding its own address, the source address and packet number. The updation in the request packet is done by adding its own address in the intermediate node addresses field and

increasing the value by one in the packet length field. If the node address is already present in the intermediate node address field of the request packet then it will discard the packet. Thus the request packet on reaching a destination will have all intermediate nodes traversed and thus it has discovered the route. After updating the request packet intermediate node sends the packet to all neighboring nodes except the source node and the nodes through which the packet has traveled to reach it.

3.3 Functions of Destination Node

The destination performs the functions of intermediate node in addition to updation of the route cache present in it. The destination will end the route discovery process only when it has no further nodes to forward the request packet. A route cache is maintained in the destination node indicating the source, destination and the route through which the request packet has traversed. If a destination node receives a request packet, then it puts an entry into the route cache denoting that one route has been discovered. The intermediate node addresses are extracted from the request packet and stored in the route cache. Also number of hops and packet number are entered into the route cache. Then the destination node forwards the packet to the neighboring nodes. Thus the request packet travels through different intermediate nodes and reaches the destination in all possible routes.

3.4 Route Selection Process

The RM node that is responsible for the selection of 'n' node disjoint trees receives all the discovered routes from the destinations of the group. In addition to RM individual destinations are also equipped with an algorithm for route selection, which selects the optimal tree for receiving the different packets based on its own available knowledge. After route Discovery, the destination node will have all the possible paths from the source to the destination in the route cache table. This information is shared with the RM of the group. The RM checks for availability of at least one node disjoint tree for each packet of a message that is shared by all the destinations. In case the failure of RM, the individual destinations can take up the process of route selection. If there are more than one node disjoint path for a packet in a destination, then the route selection is done using the following steps.

Step 1: Take the routes corresponding to a packet (say packet number 1).

Step 2: Check whether the intermediate node used in a route for packet number 1 happens to be an intermediate node for another packet number say 'i'. If so, then delete that route for packet i.

Step 3: Repeat Step 2 for other intermediate nodes.

Step 4: If all the routes for a packet number 'i' have been deleted then, we have no routes for sending that packet. So undelete all the routes for packet number 'i' and delete the conflicting route in packet 1.

Step 5: Now repeat the procedure for the routes in packet number 1.

Step 6: If two or more routes remain without any conflict, then select the route with minimum number of hops.

Step 7: Step 6 is repeated for other packets and the routes are selected.

Step 8: If two selected routes are having same intermediate nodes, then they are not node-disjoint. So the common node will send an error packet to any of the destinations and the route selection process has to be repeated for that node.

After this route selection is completed at the RM or at the destination, the route cache of the destinations is updated.

3.5 Message Encoding

In order to have sufficient level of reliability in the routing scheme an encoding scheme is required. In order to encode the message, the message is split into k packets and provided to the encoder that converts the k packets into $n = k + x$, $(n > k)$ packets. These encoded 'n' packets are sent through 'n' node disjoint distribution tree. This encoding allows $(n-k)$ failures without affecting the retrieval of messages. In this implementation, one route failure has been considered and encoding of K packets into n packets has been done using diversity coding scheme. Where $K = n - 1$.

This encoding has been done in order to ensure reliability i.e., out of n paths, Even if we lose 1 packet, we will be able to retrieve the message. In order to have more ruggedness in the routing scheme more complex code scheme can be incorporated in the basic model mentioned.

This diversity coding scheme used in this work has been explained as follows.

Let n be the number of paths established. The message is divided into $n-1$ parts. Let $M_1, M_2, \ldots, M_{n-1}$ be the message packets. Let $P_1, P_2, \ldots, P_n$ are the encoded message packets. Then

$$\begin{aligned} P_1 &= M_1 \oplus M_2 \\ P_2 &= M_2 \oplus M_3 \\ &\vdots \quad\quad \vdots \\ P_{n-2} &= M_{n-2} \oplus M_{n-1} \\ P_{n-1} &= M_{n-1} \\ P_n &= M_1. \end{aligned}$$

Where $\oplus$ denotes Bitwise XOR Operation. XOR operation is used in this scheme since it is reversible.

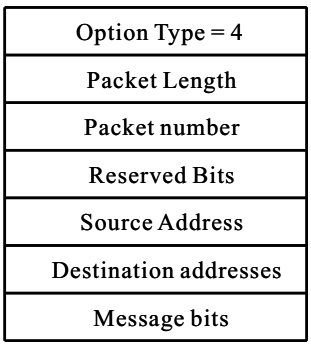

Figure 2: Message packet structure

Thus the Message is divided into $n-1$ packets and encoded into n packets. After encoding the message bits, the message packets are to be generated by the source with the packet structure as shown in Figure 2.

3.6 Message Decoding

The message packets reaching the destination are stored temporarily as they are received. The corresponding bits in the 'Received packets bit vector' is set so that the packet if any missed can be found out.

The Message Decoding procedure is as follows.

Let n be the number of packets received. Let $P_1, P_2, \ldots, P_n$ are the packets received. Let $M_1, M_2, \ldots, M_{n-1}$ are the original message packets.

$$\begin{aligned} M_1 &= P_n \\ M_2 &= P_1 \oplus M_1 \\ M_3 &= P_2 \otimes M_2 \\ \vdots &\quad \vdots \\ M_{n-1} &= P_{n-1}. \end{aligned}$$

Where $\oplus$ denotes Bitwise XOR Operation. Thus the message is decoded and retrieved. If a packet is missed, the order of the above steps has to be slightly modified in order to retrieve the message. The above mentioned simple encoding scheme is chosen for the purpose implementation and functional validation. Stronger encoding can be chosen and implemented for achieving better reliability and security.

4 Performance Analysis

This scheme SeReRoM has been validated for its functionality in a wired environment. The proposed scheme for multicast routing is found to discover the optimal routes and does the function of sending the message from the source to the destinations through multiple paths. The simulations are performed to study the following metrics and their responses to changes in number of nodes are analyzed.

1) Resource consumption ratio;

2) Control overhead ratio;

3) Number of request packets forwarded.

Also the number of request Packets forwarded is plotted against varying number of paths. The Resource Consumption Ratio (RCR) is defined as

$$\frac{No.\ of\ data\ packets\ transmitted\ by\ nodes\ across\ the\ network}{Total\ no.\ of\ data\ packets\ sent\ by\ the\ source\ *\ average\ hop\ count}.$$

The control Overhead Ratio (COR) is defined as

$$\frac{No.\ of\ Control\ bytes\ transmitted\ across\ the\ network}{Total\ No.\ of\ bytes\ sent\ by\ the\ sorce}.$$

Figure 3 gives the plot of Resource Consumption Ratio (RCR) against number of nodes with different fixed number of node disjoint paths. From the graph it could be incurred that initially the resource consumption increases with the number of nodes in the network, because initially the resource requirement is more and so as number of nodes increase the resource consumption increases. The resource requirement reaches a saturation point at a particular number of nodes. As the number of nodes increase beyond that value the resource available goes unused and so the resource consumption ratio decreases gradually. Also the resource consumption is more for the case when 3 node disjoint trees are established than the case when two node disjoint trees are needed. This is because when more paths are needed to be established, the resource consumption increases. Figure 4 is a plot of Control Overhead Ratio against number of nodes fixing the number of paths. For the topology considered, the Control overhead ratio increases when number of nodes increases from 4 to 6. After this, the control overhead ratio remains nearly constant. This is because as the number of nodes increase, the number of request packets forwarded increases and so the control overhead ratio increases. The constant region of COR is due to the presence of more than sufficient nodes need to establish the required number of paths. So, the control overhead remains approximately constant. It is seen that the rate of increase of control overhead ratio is less when 3 node disjoint trees are needed rather than when 2 node disjoint trees are needed, because as the number of paths increase, the number of message packets forwarded also increase and the raise in control overhead ratio is slow down.

Figure 5 is a plot of number of Request packets (REQ) forwarded against the number of nodes for fixed values of number of paths. It is evident that as the number of nodes increase, the number of route request packets to be forwarded also increases. Figure 6 is a plot of number of Request packets (REQ) forwarded against the number of paths for fixed values of number of nodes. It is seen from the graph that as the number of paths increase, more request packets are forwarded to discover more routes.

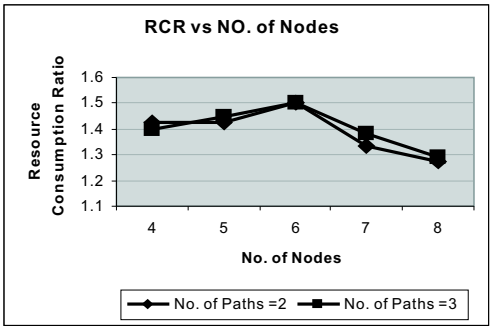

Figure 3: Plot of RCR vs no. of nodes

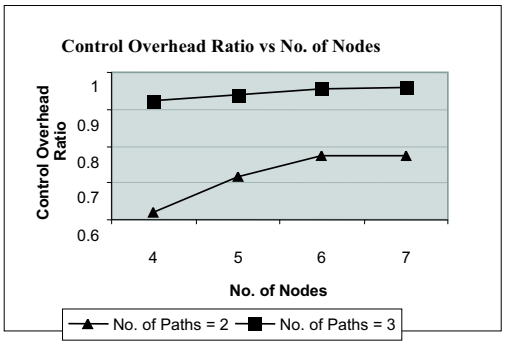

Figure 4: Plot of COR vs. no. of nodes

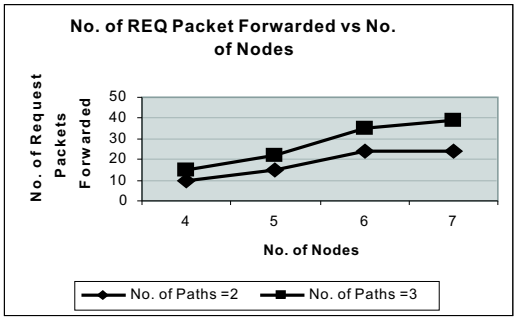

Figure 5: Plot of no. of request forwarded vs. no. of nodes

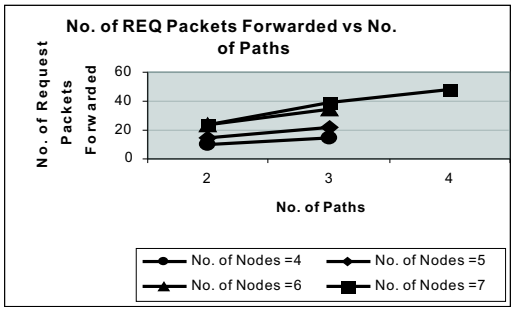

Figure 6: Plot of no. of requests forwarded vs. no. of nodes

5 Conclusion

In this scheme, a general framework of a multipath multicast routing scheme for multicast application that is not time sensitive has been developed and validated. The scheme provides reliability and security due to the multipath node disjoint routing. These two parameters can be modified to the required extent by means of varying the number of node disjoint paths and the strength of the coding scheme. The developed multipath routing scheme is simulated for multicasting environment for varying network topologies. The scheme has incorporated the method of reserving sufficient input and output buffers in order to provide a improved level of reliability in terms of reduction in packet drop at the reserved intermediate nodes. The performance of this scheme is analyzed and the results are presented.

The proposed SeReRoM is being tested for scenarios like unreliable wireless communication link and in mobile node environment.

References

[1] E. Ayanoglu, I. Chih-Lin, R. D. Gitlin, J. E. Mazo, "Diversity Coding for transparent self-Healing and fault tolerant Communication Networks," *IEEE Transactions on Communication*, vol. 41, no. 11, pp. 1677-1686, 1993.

[2] K. C. Chan and S. H. Chan, "Distributed servers approach for large-scale secure multicast," *IEEE Journal on Selected Areas in Communications - special issue on Network Support for Multicast Communications*, vol. 20, no. 8, pp. 1500-1510, Oct. 2002.

[3] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: A survey of protocols, functions, and mechanisms," *IEEE Journal on Selected Areas Communications*, vol 15, no. 3, pp. 277-290, Apr. 1997.

[4] J. J. Garcia-Luna-Aceves and E. L. Madruga, "A multicast routing protocol for ad-hoc networks," in *Proceedings of IEEE INFOCOM'99*, pp. 784-792, Mar. 1999.

[5] P. Judge and M. Ammar, "Security issues and solutions in multicast content distribution: A survey," *IEEE Network*, vol. 17, no. 1, pp. 30-36, Jan/Feb. 2003.

[6] T. Hardjono and G. Tsudik, "IP multicast security: issues and directions," *Annales de Telecom*, pp 324-340, Jul-Aug. 2000.

[7] R. Leung, J. Liu, E. Poon, "MP-DSR: A QoS-aware multipath dynamic source routing protocol for wireless ad hoc networks," in*Proceedings of 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, pp. 132-141, 2001.

[8] S. Mittra, "Iolus: A framework for scalable secure multicasting," in *Proceedings of ACM SIGCOMM'97*, pp. 277-288, Cannes, France, 1997.

[9] J. Schiller, *Mobile Communications*, 2nd ed, Pearson Education, Singapore Pvt. Ltd, 2002.
[10] B. Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., 2001.
[11] R. Srinivasan, V. Vaidehi et al, "An efficient secure group key management scheme for multicast VoIP networks," in *Proceedings of ICIS 2005*, Kuala Lumpur, Malaysia, Dec. 2005.
[12] S. Sriram, T. B. Reddy, B. S. Manoj and C. SivaRamMoorthy "MuSeQoR: Multi-path failure-tolerant security-aware QoS routing in ad hoc wireless networks," in *Proceedings of International conference on High Performance Computing 2004 (HiPC2004)*, pp. 81-90, Dec. 2004.
[13] R. Wittmann and M. Zitterbart, *Multicast Communication Protocols and Applications*, Morgan Kaufmann publishers, 2001.

R. Srinivasan received his BE (ECE) from University of Madras, India, ME (Applied Electronics) from Madurai Kamaraj University, India and Presently Pursuing his PhD from Anna University, India. He worked as Project Associate in the Microsatellite Project of Anna University from June 2003 to February 2006. He had published several papers in journals and conference proceedings. His areas of interest include Multicast Networking, Network Security and VoIP.

V. Vaidehi received BE (ECE) from College of Engineering Guindy, ME (Applied Electronics) and PhD from Madras Institute of Technology, Anna University. She had joined MIT in 1982 and currently she is a Professor in the Department of Electronics Engineering, MIT, Anna University. She had published several papers in journals and conference proceedings and had taken up several sponsored research projects. Her areas of interest include Networking, parallel processing and ADSP.

K. N. Srivathsan, L. Ramesh Babu, C. Karunagaran Final Year B.Tech students in the Department of Electronics engineering, Madras Institute of Technology Campus of Anna University, India