ISSN No. 0976-5697

Volume 1, No. 3, Sept-Oct 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Improved and Secured Routing using S-DSR in Mobile AD-HOC Networks

T. Venkat Narayana Rao*
Professor and Head,
Computer Science and Engineering,
Hyderabad Institute of Technology and Management,
Hyderabad, A P, India,
tvnrbobby@yahoo.com

Rajeshwar Moghekar
Associate Professor
Computer Science and Engineering,
Hyderabad Institute of Technology and Management,
Hyderabad, A P, India

Vani G
Assistant Professor,
Computer Science and Engineering
Malla reddy Institute of Technology & Science
Hyderabad, A.P, India

S. Janardhan Rao
Assistant Professor,
Computer Science and Engineering
Hyderabad Institute of Technology and Management,
Hyderabad, A P, India

Abstract: A Mobile Ad Hoc Network (MANET) is a collection of mobile nodes that can communicate with each other using multi hop wireless links without utilizing any fixed based-station infrastructure and centralized management. Each mobile node in the network acts as both a host generating flows or being destination of flows and a router forwarding flows directed to other nodes. With the rapid proliferation of wireless networks and mobile computing applications, Quality of Service (QOS) for mobile ad hoc networks (MANETs) has received increased attention. Security is a critical aspect of QOS provisioning in the MANET environment. Without protection from a security mechanism, attacks on QOS signaling system could result in QOS routing malfunction, interference of resource reservation, or even failure of QOS provision. Due to the characteristics of the MANETs, such as rapid topology change, limited communication and computation capacity, node failures and the conventional security measures cannot be applied and new security techniques are necessary. In this paper we propose a new scheme that is Secured Dynamic Source Routing Protocol(S-DSR) which is the enhancement to the DSR. Which include secured route discovery and QOS and achieves higher performance.

Keywords: Quality of service (QOS), MANET, DSR, Trust, Enhanced, Message Authentication Code (MAC).

I. INTRODUCTION

In ad hoc networks, communications are done over wireless media between stations directly into a peer to peer fashion without the help of wired support station or access points. Lots of efforts have been done on ad hoc networks. One of the important and famous groups developing ad hoc networks is Mobile Ad hoc Network Group (MANET). With the popularity of ad hoc networks, many routing protocols have been designed for route discovery and route maintenance. They are mostly designed for best effort transmission without any guarantee of quality of transmissions. Some of the most famous routing protocols are Dynamic Source Routing (DSR) [6], Ad hoc on Demand Vector (AODV) routing [10], Optimized Link State Routing protocol (OLSR) [1]. Quality of Service (QOS) [3] models. Ad hoc networks has become more and more required as many real time applications are implemented on the network. By considering QOS in terms of data rate and delay will help to ensure the quality of the transmission of real time media. For real time media transmission, if not enough data rate is obtained on the network, only part of the traffic will be transmitted on time. There would be no meaning to receiving the left over part at a later time because real time media is sensitive to delay. Data that arrive late can be useless. As a result, it is essential for real time transmission to have a QOS aware routing protocol to ensure QOS of transmissions.

In addition, network optimization can also be improved by setting requirements to transmissions. That is to say, prohibit the transmission of data which will be useless when it arrive the destination to the network. From the routing protocol point of view, it should be interpreted as that route which cannot satisfy the QOS requirement should not be considered as the suitable route in order to save the data rate on the network. QOS metrics for the DSR protocol are based on three primary parameters. These parameters are: bandwidth, latency and jitter. In S-DSR it is proposed to define the selection criteria for the routes from the cache which is based on these parameters. Minimum bandwidth is the bandwidth of the weakest link in the route. Latency and jitter are cumulative figures, as generated by all the intermediate nodes placed together. Latency and jitter are computed in milliseconds(ms), while bandwidth is typically mentioned in Kbps. The time stamping is used for stamping latest route verification for availability [2] [3]. For an efficient selection of route from the cache, the routes may be sorted on a periodic basis and the sorting criteria could be defined by the source, based on the application's need. The best cached route will top the list for efficient selection by the S-DSR.

The aim of this work is to give an overview of the popular MAC and routing layer solutions for ad hoc

networks and take a look at of how QOS can be added to ad hoc networks especially in the network layer. Various methods for calculation of QOS metrics are discussed. Simulations are done by using Java simulation to see how a concrete QOS conscious routing protocol performs.

II. VARIOUS TOPOLOGIES/STANDARDS/ PROTOCOLS

A. *A. Ad hoc Network*

There are two architectures that allow two wireless stations to communicate with each other. The first one relies on a third fixed party (a base station) that will hand over the offered traffic from a station to another, as illustrated in figure 2.1. This same entity will regulate the allocation of radio resources. When a source node wishes to communicate with a destination node, the former notifies the base station, which eventually establishes the communication with the destination node. At this point, the communicating nodes do not need to know about the route from one to the other. All that matters is that both source and destination nodes are within the transmission range of the base station; if one of them loses this condition, the communication will abort.

The second approach, called ad-hoc [4], does not rely on any stationary infrastructure. All nodes in ad hoc networks are mobile and can be connected dynamically in an arbitrary manner. Each node in such networks behaves as a router and takes part in discovery and maintenance of routes to other nodes[8].

Figure 1 illustrates a simple 3-node ad-hoc network. In this figure, a source node S wants to communicate with a destination node D. S and D are not within transmission range of each other. Therefore, both use the relay node R to forward packets from one to another. R functions as a host and a router at the same time. By definition, a router is an entity that determines the path to be used in order to forward a packet towards its final destination. The router chooses the next node to which a packet should be forwarded according to its current understanding of the state of the network.

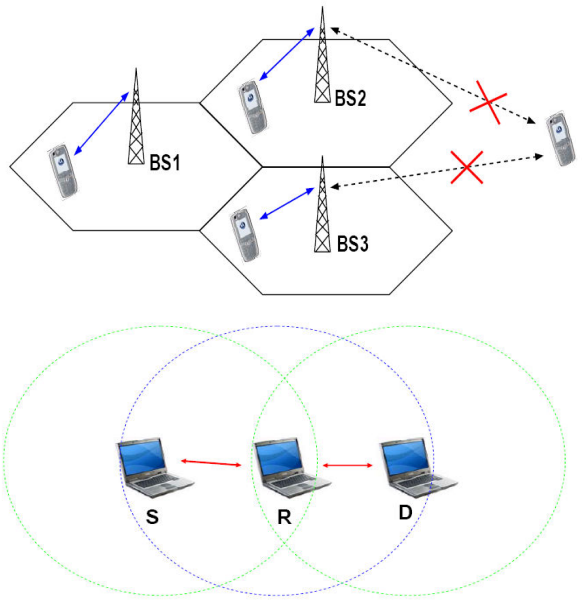

Figure: 1Illustration of the Infrastructure and Infrastructure-less Network Model

B. *Routing Classification in Ad Hoc Networks and Clustering*

Routing in wireless ad hoc networks is clearly different from routing found in traditional infrastructure networks. Routing in ad hoc networks needs to take into account many factors including topology, selection of routing path and routing overhead, and it must find a path quickly and efficiently. Ad hoc networks generally have lower available resources compared with infrastructure networks and hence there is a need for optimal routing. Also, the highly dynamic nature of these networks means that routing protocols have to be specifically designed for them, thus motivating the study of protocols that aim at achieving routing stability.

Designing a routing protocol for ad hoc networks is challenging because of the need to take into account two contradictory factors:

- A node needs to know at least the "reachability" information to its neighbors for determining a packet route.
- The network topology can change quite often.

Scalability is one of the important problems in ad hoc networking. Scalability in ad hoc networks can be largely defined as the network's ability to provide an acceptable level of service to [9] packets even in the presence of a huge number of nodes in the network. In proactive routing protocols, when the number of nodes in the network increase, the number of topology control messages increases non-linearly and they may consume a large portion of the available bandwidth. In reactive routing protocols, large numbers of route requests to the entire network may eventually become packet broadcast storms. Typically, when the network size increases beyond certain thresholds, the computation and storage requirements become infeasible. When mobility is considered, the frequency of routing information updates may be significantly increased, thus worsening the scalability issues.

C. *Dynamic Source Routing (DSR)*

DSR [6] offers a number of potential advantages over other routing protocols for mobile ad hoc networks. First, DSR uses no periodic routing messages (e.g., no router advertisements and no link-level neighbor status messages), thereby reducing network bandwidth overhead, conserving battery power, and avoiding the propagation of potentially large routing updates throughout the ad hoc network[12]. The Dynamic Source Routing protocol is able to adapt quickly to changes such as host movement yet requires no routing protocol overhead during periods in which no such changes occurs.

In addition, DSR has been designed to compute correct routes in the presence of asymmetric (uni-directional) links. In wireless networks, links may at times operate asymmetrically due to sources of interference, differing radio or antenna capabilities or the intentional use of asymmetric communication technology such as satellites. Due to the existence of asymmetric links, traditional link-state or distance vector protocols may compute routes that do not work. DSR, however, will find a correct route even in the presence of asymmetric links [13]. The operation of the DSR protocol can be summarized as follows:

1. The Route Cache **2. Route Discovery**
3. Data Transfer **4. Route Maintenance**

Secure Routing: The contemporary routing protocols for Ad hoc networks cope well with dynamically changing topology but are not designed to accommodate defense against malicious attackers. No single standard protocol capture common security threats and provide guidelines to secure routing protocol[11]. Routers exchange network topology informally in order to establish routes between nodes. External attackers injecting erroneous routing information, replaying old routing information or distorting routing information in order to partition a network or overloading a network with retransmissions and inefficient routing. Routing information signed by each node won't work since compromised nodes can generate valid signatures using their private keys. Detection of compromised nodes through routing information is also difficult due to dynamic topology of Ad hoc networks. It is possible to make use of some properties of ad hoc networks to facilitate secure routing. Routing protocols for Ad hoc networks must handle outdated routing information to accommodate dynamic change in topology. False routing information generated by compromised nodes can also be regarded as outdated routing information.

D. ***Problems associated with Ad-hoc routing***

a) Infrastructure: An Ad-hoc network is an infrastructure less network (figure 2(a)). Unlike traditional networks (figure 2(b)) there is no pre-deployed infrastructure such as centrally administered routers or strict policy for supporting end-to-end routing. The nodes themselves are responsible for routing packets. Each node relies on the other nodes to route packets for them. Mobile nodes in direct radio range of one another can communicate directly, but nodes that are too far apart to communicate directly must depend on the intermediate nodes to route messages for them.

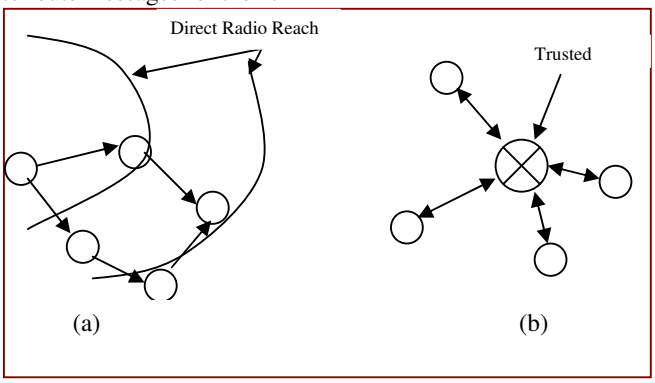

Figure 2 (a) Routing in Ad-hoc networks (b) Routing in traditional networks using router

b) Frequent changes in network topology: Ad-hoc networks contain nodes that may frequently change their locations. Hence, the topology in these networks is highly dynamic [14]. This results in frequently changing neighbors on whom a node relies for routing. As a result traditional routing protocols can no longer be used in such an environment. This consents new routing protocols that can handle the dynamic topology by facilitating fresh route discoveries.

c) Problems associated with wireless communication: As the communication is through wireless medium, it is possible for any intruder to tap the communication easily. Wireless channels offer poor protection and roting related control messages can be altered. The wireless medium is susceptible to signal interference, jamming, eavesdropping and distortion [17]. An intruder can easily eavesdrop to know sensitive routing information or jam the signals to prevent propagation of routing information or worse interrupt messages and distort them to manipulate routes. Routing protocols should be well adopted to handle such problems.

E. Problems with existing Ad-hoc routing protocols

a) Implicit trust relationship between neighbors: Current Ad-hoc routing protocols inherently trust all participants. Most Ad-hoc routing protocols are cooperative by nature and depend on neighboring nodes to route packets. This naive trust model allows malicious nodes to paralyze an Ad-hoc network by inserting erroneous routing updates, replaying old messages, changing routing updates or advertising incorrect routing information. While these attacks are possible in fixed network as well, the Ad-hoc environment magnifies this makes detection difficult.

b) Throughput: An ad-hoc network maximizes the total network throughput by using all the available nodes for routing and forwarding. However, a node may misbehave by agreeing to forward packets and then failing to do so, because it is overloaded, selfish, malicious or broken. Misbehaving nodes can be a significant problem. Although the average loss in throughput due to misbehaving nodes is not too high, in the worst case it is very high.

c) Attacks using modification of protocol fields of messages: Current routing protocols assume that nodes do not alter the protocol fields of messages passed among nodes. Routing protocol packets carry important control information that governs the behavior of data transmission in Ad-hoc networks. Since the level of trust in a traditional Ad-hoc network cannot be measured or enforced, enemy nodes or compromised nodes may take part directly in the route discovery and may intercept and filter routing protocol packets to upset communication [13]. Malicious nodes can easily cause redirection of network traffic and DOS attacks by simply altering these fields.

III. THE PROPOSED WORK

We shall now focus on the integration of SRP with DSR to get S-DSR for Secured Route Discovery. There exists no security association in the DSR protocol and it is presumed that among the nodes participating in the network none are having malicious intent. SRP can work over and above basic protocols, which now in our discussion is limited to DSR. The source S, trying to find a route to destination D, will trigger a route discovery if there is no route available in the route cache of the source node.

A. Enhancement of DSR for Secured Route Discovery

SRP needs a SA between the two communicating nodes and it uses two identities, for it, random request identifier and request id. MAC(Message Authentication Code) is calculated based on these ids and $K_{S,D.}$ where $K_{S,D}$ is key shared key between source and destination. The Figure 3 (a) and 3 (b) showcases how packets are transferred from source to destination and how acknowledge is obtained. It may be noted here that DSR also needs a random id for its

operation and it also accumulates ids of traversed nodes in the route request packet. In S-DSR it is proposed to integrate the DSR and SRP functionality into a single protocol. The route request packet format for S-DSR will be :{ S, D, requested, randomrequestidentifier, MAC, Node List: S} only the relevant components, which are applicable for the S-DSR, are listed in the above format. As the Route Request packet will flow, ids of the intermediate nodes will get accumulated in the list of the request packet.

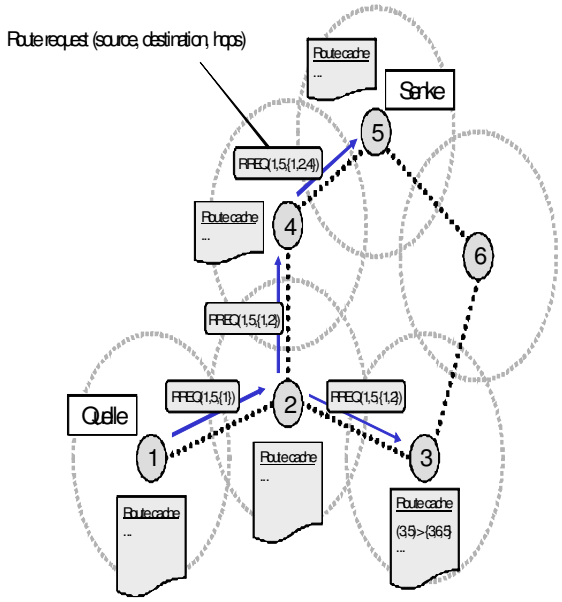

Figure 3 (a) Sending data from Source to Destination

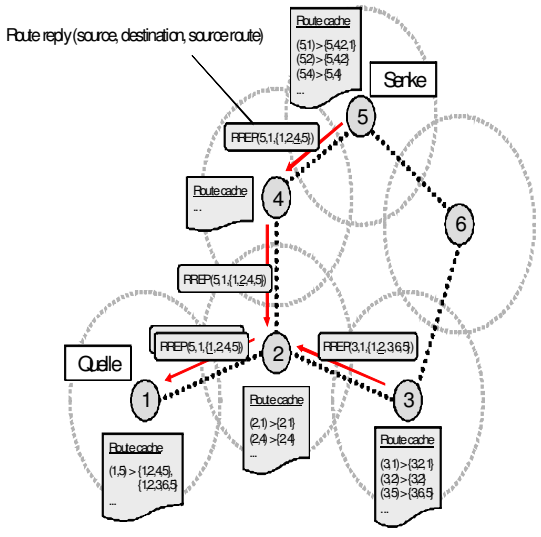

Figure 3 (b) Receiving acknowledgement from Destination

Successful verification will cache the discovered routes in the route cache. In this process multiple routes will be discovered [3], since DSR and SRP do not prevent discovery of multiple routes. Thus S-DSR retains the basic route discovery functionality of DSR and integrates the security aspects based on SRP proposals into its basic functioning. The secured route discovery of multiple routes between two communicating nodes is achieved in S-DSR with minimum modifications in the methodology of DSR and SRP

B. Enhancement using Multiple Routes

DSR, by its design, suggests using alternate cached routes only when the 'in use' link is broken. An alternate link from the route cache is used for continuing the data transmission. As an enhancement to the ad hoc network operations, it is now proposed to use multiple routes concurrently for data transmission, as per the methodology suggested in SMT. It may be noted here that DSR is only for route discovery and not for data transmission, although route maintenance is a part of the DSR operation. SMT strongly relies on usage of multiple routes between the communicating nodes. Data packets to be transmitted from the source to the destination are dispersed into multiple packets (P) and are routed through multiple routes simultaneously. At the destination, receipt of Q out of P packets can ensure reconstruction of the original packet protocols. Let us say for example, the data packet is dispersed into four parts (P=4) and each dispersed piece is transmitted through different routes and carries a Message Authentication Code (MAC), based on which the destination can verify the integrity of the packet and authenticity of its origin. Three out of four packets are enough to reconstruct the original message. Each packet received at the destination is acknowledged through a feedback. The feedback mechanism is also fault tolerant, secure, dispersed and cryptographically protected. If two packets are received at the destination and two are either lost or compromised [11] [16]. The destination extracts information from first received packet and waits for remaining packets while setting are caption timer. On expiry of the timer, the receiver generates acknowledgement for the two successfully received packets. The sender rejects the two failing routes, on receipt of the acknowledgement packets and retransmits the two packets. One of the retransmitted packets is again compromised. Since only three out of four packets are enough to reconstruct the message at the destination, the receiver acknowledges successful reception, even before expiration of timer.

C. Enhancement of DSR Route Caching

DSR discovers multiple routes between two communicating nodes and these routes are cached at the end nodes, as well as on the intermediate nodes. DSR also suggests techniques to improve the cache contents [2]. The most practical are, by supporting techniques based on caching over heard routing information and by replying to route requests using cached routes. However as far as usage of multiple routes cached at the node is concerned, DSR is silent, except for using an alternate route in case of link failure. In this section an augmented approach to cache management is proposed.

D. Route Selection Based on QOS Metrics from the Cache

QOS metric for the DSR protocol is based on three primary parameters. These parameters are: bandwidth, latency and jitters. In S-DSR it is proposed to define the selection criteria for the routes from the cache which is based on these parameters. Minimum bandwidth is the bandwidth of the

weakest link in the route. Latency and jitter are cumulative figures, as generated by all the intermediate nodes put together. Latency and jitter are computed in milliseconds (ms), while bandwidth is typically mentioned in Kbps. The time stamping is used for stamping latest route verification for availability. For an e-client selection of route from the cache, the routes may be sorted on a periodic basis and the sorting criteria could be defined by the source, based on the application's need. The best cached route will top the list for efficient selection by the S-DSR algorithm[15]. The S-DSR algorithm contains the following modules.

a) Route Builder: Route builder performs the route discovery process to find the possible routes between sources to destination based on the node communication ranges. Whenever a node 'S' wants to communicate with another node 'D' in the network, it should initiate a route discovery process if S is not aware of any paths to D or all such paths already known to S are broken. In order to initiate such a process, S signs and broadcasts a route request (RREQ) message. Whenever an intermediate node 'IM' receives a route request message it simply sign and rebroadcast it. However, in order to prevent malicious nodes from tampering with RREQ messages, intermediate nodes verify each of the signatures in the RREQ message they received [12]. Also, before re-broadcasting the message, node 'IM' should check the sequence of the signatures in the message. If the certificate of 'IM' is already in sequence, the message should be discarded rather than re-broadcasted. This will prevent the RREQ messages from being trapped in a loop. Each intermediate node can receive a route request, originated by node S looking for a path to another node D, from several of its neighbors. Since our protocol is a multi path protocol, the intermediate node should in principle rebroadcast all such messages. However, since the set of all possible routes from S to D can be very large, discovering or keeping track of all such paths does not scale well as the size of network increases. Therefore, we decided to limit the number of routes in each route discovery process to some constant. When the destination node D receives the first route request message from a source node S, it sets a timer for that node and starts to respond to every route request message it receives from S, except for route requests from S which are not node-disjoint with the other paths D has already sent back to S. When D reply to a route request, it should sign and send back a route reply message.

b) Path Manager: Path Manager manage path table for source routing and based on node trust it removes path from path table. The path manager module is responsible to evaluate the routes based on the trust value of the nodes in this route and selects a route based on this evaluation. The routes are evaluated and a route with the highest rating is then selected, i.e. the best route will be considered as one that the highest trust rating which means that it has the lowest number of malicious nodes [5]. Therefore, a route that contains a malicious node is not good because it will always result in a packet dropping. Once a node A detects another node B as misbehaving, A isolates B from communications by not using B for routing and forwarding and by not allowing B to use A either. This isolation has three purposes. The first is to reduce the effect of misbehavior by depriving the misbehaving node of the opportunity to participate in the network activities. The second purpose is to serve as an incentive to behave well in order not to be denied service [7]. Finally, the third purpose is to obtain better service by not using misbehaving nodes on the path. The *path manager* performs the following functions:

- Path re-ordering according to trust metric.
- Deletion of paths containing malicious nodes.
- Action on receiving a request for a route from a malicious node.
- Action on receiving request for a route containing
- A malicious node in the source route.

c) Trust Manager: In an ad hoc environment, trust management has to be distributed and adaptive. This component deals with messages provided by the source node and path manager. It updates the trust values of each node based on the source node message and even provide dual factor trust value to path manager for making necessary decision before eliminating the misbehaving node routes from the path table. The trust manager stores the trust information about all known nodes during run time, and it offers methods to query for information about stored trust values. It is used as the interface between the existing dynamic source routing protocol on a hand and the trust formatter and trust updater modules on the other hand.

The trust manager performs the following functions:

- Trust function to calculate trust levels.
- Trust table entries management for trust level administration.
- Forwarding of misbehaving messages.
- Trust table manage trust levels for nodes.

Trust is important when making a decision about the following issues:

- Providing or accepting routing information.
- Accepting a node as part of a route.
- Taking part in a route originated by some other node.

The *trust updater* implements the functions for updating trust. The trust value depends on a given node experience in a given situation. The trust updater updates trust value by using following parameters:

- Previous trust value.
- Number of positive and negative experiences in the past.
- Number of positive and negative experiences in the past.

IV. IMPLEMENTATION OF ENHANCED DSR

A) Algorithm for Path Manager

Step I: Take ArrayList mac, Get database connection.
Step II: Void makeFPath(String fname,int brcnt,int src,String prot).
Step 1: repeat nodes until you get all the malicious nodes to malicious_node.
Step 2: create new file "simnoderange".
Step 3: Read new file simroutetable.
Step 4: Get node range in ArrayList sfnodes.
Step 5: Repeat steps from 6 to 24 until rcnt<brcnt

Step 6: increment count, read data from route table in S.
Step 7: Repeat the steps from 8 to 12 until string becomes null.
Step 8: create StringTokenizer st. get next token values in com, mame, srcnode.
Step 9: macflag=false.
Step 10: Repeat the values until all the tokens are getting read.

Step 11: Get next token value in next.
Step 12: If array list mac contains the next values make macflag=true.
Step 13: Repeat steps from 14 to 24 until macflag becomes true.
Step 14: create StringTokenizer st. get next token values in com, mame, srcnode
Step 15: mb=false.
Step 16: Repeat the values until all the tokens are getting read.
Step 17: Get next token value in next
Step 18: If array list malicious_node contains the next values
Step 19: take variable m and initialize to zero.

Step 20: mb=true;
Step 21: Add next value to mac
Step 23: call thread function thrice
Step 24: if (!mb) Display route value

B) Algorithm for Route Manager

Step I: Get database connection.
Step II: SRoute(String s,int pcnt).
Step 1: Repeat steps below until database contains values in the table.
Step 2: Create ArrayList arraylist, arraylist1. Add values to arraylist1.
Step 3: Create Random Variable random.
Step 4: Repeat the following steps from 5 to 12 until based on string choice 0 or 1 or 2 or 3.
Step 5: Clear arraylist.
Step 6:Create new file DSR.sim SADSR.sim, MDSR.sim, MSADSR.sim depending on choice 0, 1, 2, 3 and get the file output in it.
Step 7: if(pcnt<1) d=0
Step 8: Generate 3 random numbers and add to arralist.
Step 9: Repeat steps from 10 to 12 by taking arraylist1 size.
Step 10: s6 = "DSR ".
Step 11: Append arraylist and arraylist1 values to s6 and get bytes from s6.
Step 12: Write bytes into DSR.sim, SADSR.sim, MDSR.sim, MSADSR.sim file according the choice 0,1,2, or 3.

C) Algorithm for Trust Manager

Step I : public int getRouteTrust(ArrayList rt)
Step 1: get database connection.
Step 2: get size of rt.
Step 3: generate the Limit.
Step 4: Repeat steps from 5 to 10 by taking rt size.
Step 5:get data fro trust table.
Step 6:repeat the steps 7 and 8 until the database contains values in the table.
Step 7:nx=rs.getInt(1).
Step 8: ny=rs.getInt(2).
Step9:nodetrust=Math.round((nx/(nx+(MAL_CONST*ny)))).
Step 10: tot_rt+=nodetrust.
Step 11: if(tot_rt<trLimit) tot_rt=0.

V. IMPLEMENTATION AND RESULT ANALYSIS

Table I Obtained results from S-DSR Algorithm

Reputation Table		Trust Table	NODES				
Node_no	Node_repT	TrustY	Node_Name	Node X	Node Y	Node S	mseq
0	2	0	0	365	450	N	0
1	2	0	1	248	492	N	0
2	2	0	2	428	493	N	0
3	2	0	3	114	37	N	0
4	2	0	4	371	474	MC	1
5	2	0	5	15	345	N	0
6	2	0	6	425	28	N	0
7	2	0	7	204	176	N	0
8	2	0	8	496	280	N	0
9	2	0	9	237	24	N	0
10	2	0	10	291	425	N	0
11	2	0	11	114	76	N	0

The above table represents the malicious nodes that were present in the route while transferring the data from source to destination.

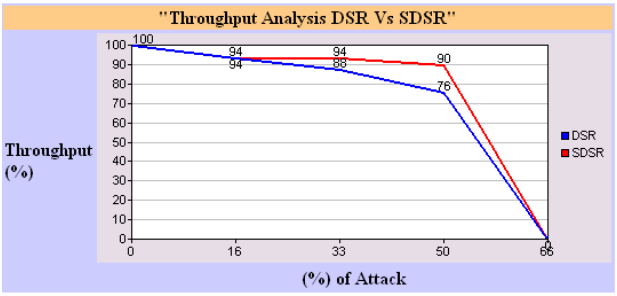

Figure 4 Comparison of Throughput in DSR Vs S-DSR

The throughput in DSR and S-DSR can be calculated as:

Throughput = *(Total No. of Data Packets Transmitted / Total No. Data Packets Ack .Recvd.).*

Suppose Per sec if it is sending 4 Data PKTs

Per 30 sec => 30*4 =120 packets

Since during communication Source the Path through which data need to be routed so packet deliver in the absence of malicious nodes will be 100%.

In case of node malicious behavior packet drop decrease the deliver ratio.

Choosing an alternate path in case DSR is more delay as it is going one by one path till it finds correct path, but in case of S-DSR it simply discards the routes which contains the malicious node and route the data in next path.

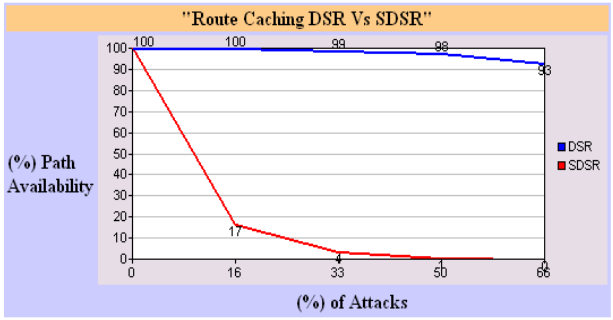

Figure 5 Comparison of Route cashing in DSR Vs S-DSR

No. of routes present in the Route Table after removing malicious paths in S-DSR will be low when comparing with DSR. Each time when malicious occurs S-DSR remove the paths having the malicious Node. Which make S-DSR for better throughput over DSR.

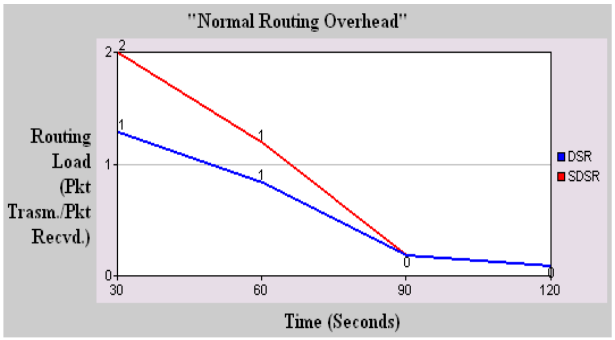

Figure 6 Comparison with Routing Overhead of DSR Vs S-DSR.

The routing overhead can be calculated for DSR and S-DSR as follows.

OVH = *(Total No. of Route Request Packet Transmitted / Total No. Route Reply Received.).*

Example: Per sec it sending 4 RREQ.

Per 30 sec => 30*4 =120 pkts

Finding a destination with in 30sec count as 1 RREP. If we find 60 routes in 30 secs then Total RREP Received is 60.

Then OVH = 120/60 = 2.0

In case of S-DSR no. of RREP decreased due to intermediate node trust validation, which in turn increases OVH for S-DSR.

CONCLUSION

DSR is a much matured protocol and a lot of research work has verified its functioning and electiveness. Here, possible enhancements to DSR is to provide much secured features and thus proposed. Further, proposals are also made for better route cache maintenance and management. By incorporating the functioning of SRP into DSR, new secured protocol, which has been named as S-DSR is proposed. Concurrent use of multiple paths, as per the functioning guidelines of SMT, has further enhanced the capabilities of DSR for secured delivery of data packets, even in presence of malicious nodes. Concurrent use of multiple paths can provide an additional strength to S-DSR for improved QoS by enhancing the availability, throughput and reliability. It has been found that with limited increase in the overhead, a far more robust and efficient protocol named S-DSR has emerged out with a improved performance.

FUTURE ENHANCEMENTS

Here only some of the security services such as Authentication, Integrity are provided while avoiding Looping problems. In future the solutions to the remaining attacks which are discussed above in MANETs can be implemented in S-DSR.

REFERENCES

[1] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR)”, RFC 3626 of IETF, Oct. 2003.

[2] S. Corson and J. Macker, “Mobile Ad hoc networking (MANET) routing protocol performance issues and evaluation considerations” Request for Comments 2501(RFC) of Internet Engineering Task Force (IETF), Jan. 1999.

[3] Y. C. Hu and D. B. Johnson, “Securing quality-of- service route discovery in on-demand routing for Ad hoc networks”, in Security of Ad Hoc and Sensor Networks 2004(SASN 2004), Washington, USA, Oct. 2004.

[4] Y. C. Hu and A. Perrig, “A survey of secure wireless Ad hoc routing”, IEEE Security & Privacy, vol. 2, no. 3, IEEE Computer Society, pp. 28-39, May-June 2004.

[5] Y. C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A secure on-demand routing protocol for Ad hoc networks”, in Proceedings of the Eight Annual International Conference on Mobile Computing and Net- working (MobiCom 2002), pp. 12-23, Sept. 2002.

[6] D. B. Johnson, D. A. Maltz, and Y. C. Hu, “The dynamic source routing protocols for mobile Ad hoc networks”, Internet-Draft, draft-ietf-manet-dsr-10.txt, July 2004.

[7] R. Ogier, F. Templin, and M. Lewis, “Topology dissemination based on reverse-path forwarding (TBRPF)”, RFC 3684 of IETF, Feb. 2004.

[8] P. Papadimitratos and Z. J. Haas, “Secure message transmission in mobile Ad hoc networks”, Elsevier Ad Hoc Networks Journal, vol. 1, no. 1, pp. 193-209, July 2003.

[9] P. Papadimitratos, Z. J. Haas, and P. Samar, “The secure routing protocol (SRP) for Ad hoc networks”, Internet-Draft, draft-secure-routing- protocol-srp-00.txt, Sept. 2002.

[10] C. E. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing”, in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, pp. 90-100, Feb.1999.

[11] M. G. Zapata, “Secure Ad hoc on-demand distance vector routing for wireless networks”, in Poster presentation, ACM Symposium on Mobile Ad Hoc Networking & Computing (Mobihoc'01), pp. 106-107, Long Beach, California, Oct. 2001.

[12] R. Zucceratto and C. Adams, “Using elliptic curve Diffie-Hellman in the SPKM GSS-API” Internet Draft, IETF, Aug. 1999.

[13] J.-P. Hubaux, L. Buttyan, and S. Capkun, “The quest for security in mobile ad hoc networks,” in The 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing, October 2001.

[14] L. Zhou and Z. Haas, Securing ad hoc networks,” IEEE Network Magazine, vol. 13, November/December 1999.

[15] Sergio Marti and T. J. Giuli and Kevin Lai and Mary Baker, “Mitigating routing misbehavior in mobile ad hoc networks, Mobile Computing and Networking” (2000).

[16]Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet Leashes: “A Defense against Wormhole Attacks in Wireless Ad Hoc Networks”. Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies.

[17] Anil Rawat, Prakash Dattatraya Vyavhare and Ashwani kumar Ramani, “Enhacned DSR for MANET with improved secured route discovery and QoS” in International Journal of Network Security, Vol.5, No.2, PP.158-166, Sept. 2007 .

AUTHORS

Author1:*Prof.T.Venkat Narayana Rao*, received the B.E in Computer Technology and Engineering from Nagpur University, Nagpur , India , M.B.A (Systems) and M.Tech in computer Science from Jawaharlal Technological University , Hydarebad, A.P., India . He has 19 years of vast experience in Computer Engineering area pertaining to academics and industry related I.T issues. He is presently Professor and Head, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management (HITAM) R.R District, A.P, INDIA. He is Editor and Reviewer of Number of International Journals. He is currently working on research areas which include Digital Signal Processing, Digital Watermarking and Data Mining, Network Communications and Security.

Author2:Vani.G received my B.Tech degree in 2005 from Kuppam Engineering College affiliated to JNTU in Computer Science and Information Technology. Presently I am working in Malla reddy Institute of Technology & Science as an Assisstant Professor in CSE Department. I am working in the filed of Ad-hoc networks. My research interests includes computer networks, cluster and grid computing.

Author3:Rajeshwar Moghekar graduated his B.Tech from Gulbarga University, Karnataka, India, in 1997, Masters Degree in Computer science from Jawaharlal Nehru Technological University (JNTU), Kakinada, A.P, India, in 2007.He is having 11 years of total experience. He is currently working as an Associate Professor in the department of Computer Science at Hyderabad Institute of Technology And Management, India. He has worked as Technical executive for MicroUniv a Training Division of MicroLand. His main research fields include Network security, ad-hoc networks, Digital watermarking .

Author4:Syamalapalli Janardhan Rao graduated his AMIE from Kolkata, India, Masters Degree in ECE from Jawaharlal Nehru Technological University (JNTU), Anantapur, A.P, India, in 2010. He got another Master Degree in CSE from IETE Delhi, India in 2009. He is currently being Fellow of IETE, New Delhi, and working as an Assistant Professor in the Department of Computer science at Hyderabad Institute of Technology And Management, India. He has served for 20 years in Indian Air force on different RADAR Systems. His main research fields include Network security, and Digital Water Marking.