Reliable Ad-hoc On-demand Distance Vector Routing Protocol

Sandhya Khurana
Department of Computer Sc., University of Delhi , New Delhi, India skhurana@cs.du.ac.in

Neelima Gupta
Department of Computer Sc., University of Delhi, New Delhi, India ngupta@cs.du.ac.in

Nagender Aneja
Department of Computer Sc., Tecnia Institute of Advance Studies, New Delhi, India. naneja@gmail.com

Abstract

Mobile Ad hoc Networks' (MANETs) properties present major vulnerabilities in security. The threats considered in MANETS are due to maliciousness that intentionally disrupt the network by using variety of attacks and due to selfishness of node which do not perform certain operations due to a wish to save power. In this paper, a co-operative security scheme called Reliable Ad hoc On-demand Distance Vector (RAODV) routing protocol based on local monitoring has been proposed to solve the problem of attack by malicious node as well as selfish behavior. RAODV behaves as AODV in the absence of attack and, detects and isolates misbehaving nodes in the presence of attack. Also it recovers from the attack when a misbehaving node leaves the network or becomes good.

1. Introduction

Ad-hoc networks [1] have been proposed to support scenarios where no wired infrastructure exists. They can be set up quickly where the existing infrastructure does not meet application requirements for reasons such as security, cost, or quality. Examples of applications for ad hoc networks range from military operations, emergency disaster relief to community networking and interaction between attendees at a meeting or students during a lecture.

In Mobile Ad hoc Networks (MANET) each node has limited wireless transmission range, so the routing in MANETs depends on the cooperation of intermediate nodes. Two types of routing protocols have been defined for ad hoc networks: Table-driven protocol and On-demand routing protocol. Table driven protocols are proactive in nature and consume excessive network bandwidth. On the other hand, on demand routing protocol exchange routing information only when needed. Ad-hoc On demand Distance Vector (AODV) [3] routing protocol is an on demand routing protocol that focuses on discovering the shortest path between two nodes with no consideration of the reliability of a node.

The structure of an Ad-hoc network leads to some special kinds of attacks especially attacks on the connectedness of the network. Most ad hoc routing protocols rely on implicit trust-your-neighbor relationship to route packets among participating nodes. This naive trust model allows malicious nodes and selfish nodes to paralyze the network. Selfish nodes do not directly damage other nodes but their effect cannot be underestimated.

Security in the routing protocol is necessary in order to guard against these attacks but relatively little work has been done in securing ad hoc network routing protocols. Secure ad hoc network routing protocols are difficult to design, due to the high dynamic nature of the network. Some protocols have been proposed to secure the network from these attacks. Some of these protocols handle attacks by malicious nodes but not the selfish nodes and some handle selfish nodes nicely but malicious nodes not so nicely.

Secure Efficient Ad hoc Distance Vector routing protocol (SEAD) [4] is based on DSDV and hence consumes network bandwidth in exchanging routing tables among nodes. It claim to save bandwidth but that is only when compared with other cryptographic schemes used to impose security but not with respect to exchange of routing information. SEAD handles malicious node but not selfish node.

The Security-aware Ad-hoc Routing (SAR) [5] based on AODV introduces security metrics for path computation and selection. Each node is assigned a trust level according to organizational hierarchies with a shared key for each level. It does not discuss how shared key is distributed among nodes and what happens when a node leaves the group with shared trust level and become malicious or selfish.

Certain protocols have been proposed to handle the selfish nodes. "Incentives to co-operate" [6] scheme introduces a virtual currency called Nuglets used in every transaction. Nuglets serve as a per-hop payment for every packet forwarding, incremented when node forwards for others and decremented when it sends packets for themselves. This encourages nodes to forward the packets and discourages them from flooding the network. The main drawback of this approach is in difficulty of implementing the exchange of currency making their use difficult in practical systems.

CORE [7] handles selfish nodes but does not handle malicious nodes nicely. It uses reputation value for each node. Nodes monitor other nodes and, based on promiscuous observations and expected behavior, maintain reputation records indicating how much other nodes in the network are trust-able. Reputations have positive, negative and zero values. Reputation of a node decreases when it refuses to forward a packet. It prevents malicious nodes to deny a service by not allowing it to advertise negatively about other nodes. Negative advertisements are ignored. However, it allows malicious node to stay in the system for some time after it has shown some positive behavior by accumulating some positive reputation. To eliminate this problem Co-operation of Nodes Fairness In Dynamic Ad-hoc Networks (CONFIDANT) [8] keeps only two types of values, negative and zero.

There is an issue regarding negative effects of reputation system on well behaved nodes which may wish to decrease its reputation by behaving badly to prevent its resources being over used. CORE and CONFIDANT require sophisticated hardware to perform promiscuous observation. The network hardware may have to listen and construct packets over different radio technologies. This may not be possible for resource-constrained devices.

In this paper, we propose a co-operative security scheme based on local monitoring to solve the problem of attack by malicious as well as selfish nodes. We present "Reliable Ad-hoc On-demand Distance Vector routing protocol (RAODV)", an approach to routing that incorporates reliability level of nodes into traditional routing metrics for finding path. RAODV is based on AODV with the assumption that nodes cannot impersonate and all other network conditions are good. RAODV behaves as AODV in the absence of attack and, detects and isolates misbehaving nodes in the presence of attack. Also it recovers from the attack when a misbehaving node leaves the network or becomes good. It does not need any special type of hardware like CONFIDANT and CORE. This protocol is simple to implement.

2. ATTACKS

Malicious nodes attack by inserting erroneous routing updates, replaying old routing information, changing routing updates, or advertising incorrect routing information so that the network is not able to provide service properly. Attacks like reducing the amount of routing information available to other nodes, failing to advertise certain routes or discarding routing packets or parts of routing packets are due to selfish behavior of a node.

Misbehaving node model as defined in [9] has three types of selfish nodes depending upon their extent of non-cooperation in network operations. Selfish node of Type 1 forwards control packets but does not forward data packets and is saving a significant portion of its battery life by neglecting data packets. Selfish node of Type 2 uses energy only for its communication and neither forwards controls packets nor data packets. Selfish node of Type 3 depends on energy level. Let E be initial maximum energy of node. When energy of the node falls within (E, T_1) the node behaves properly and execute both routing functions and packet forwarding. When energy falls in (T_1, T_2), the node behaves like selfish node of Type 1 and thus disables data packet forwarding. If energy falls within (T_2, 0) then node behaves like selfish node of Type 2. With in a limited time interval the node's energy is set back to the initial value.

In our protocol, we aim at protecting the network against attacks by selfish nodes and malicious nodes exhibiting the following misbehavior:

1. BLACK HOLE attack is an active attack in which node responds positively to a request for a shortest route, even though it does not have a valid route to the destination node. The node is called BLACK NODE. Since a black node does not have to check its routing table it is the first one to respond to route discovery request in most cases. When the data packet routed by the source reaches the black node, it drops the packets rather than forwarding to the destination as it does not have a valid route to it. Black hole attack can be co-operative involving multiple nodes, where black nodes are acting in coordination with each other.
2. Replay attack or attack by replaying old routing information.

3. Lack of error messages, although an error has been observed.

3. AODV

AODV algorithm works in two phases: Path discovery and Path maintenance. For path discovery the algorithm uses Route Requests (RREQs) and Route Replies (RREPs) messages. For path maintenance the algorithm uses Route Errors (RERRs) and HELLO messages. When a node wants to communicate with another node it looks for a route in its table. If a valid entry is found for the destination it uses that path else the node broadcasts the RREQ to its neighbors to locate the destination. Neighbors again broadcast RREQ to their neighbors. A reverse path for the source is created at every node. This process continues until either the destination or an intermediate node with a fresh route to the destination is located.

The node, then builds an RREP packet and sends to the node from which it received RREQ packet. At each intermediate node on the reverse route the RREP packet is inspected and a forward path to the destination is constructed or updated. Path discovery completes when RREP reaches to the originator.

For path maintenance, each node periodically broadcasts a HELLO message to its neighbors and, if a node has moved out, it informs other nodes by RERR message.

4. RAODV

AODV has been extended to RAODV by adding two types of control packets: Reliable Route Discovery Unit (RRDU) and RRDU Reply (RRDU_REP). RRDU messages are control packets sent by the source node along with RRDU-ID, to the destination at regular intervals and RRDU_REP message is the response of RRDU by the destination to the source node. RRDU_REP can only be generated by the destination. We assume here that there is no impersonation i.e. no node other than the destination, can generate RRDU_REP on behalf of the destination.

We also add a field Reliability List (RL) in the routing table entry. An entry in the RL has Source address, a field called Forward Data Packet Count (FDPC) and RRDU-ID, i.e. the triplet (Source address, FDPC, RRDU-ID).

The format of RAODV Routing Table entry is same as that of AODV except for the additional RL field. As in AODV, RAODV uses RREQ, RREP messages for route discovery and RERR, HELLO messages for route maintenance. In addition, RAODV also uses RRDU and RRDU_REP to help discover the path and for reliability maintenance.

TABLE I. RAODV ROUTING TABLE ENTRY

Destination (IP Address , Sequence Number)	Hop count	Next Hop	Valid Sequence #	Precursor	Life Time	RL

Path discovery in RAODV can be thought of as consisting of two phases. Phase I is same as that in AODV. That is, when a node wishes to communicate with another node it looks for a route in its table. If a valid entry is found for the destination it uses that path else the node broadcasts the RREQ to its neighbors to locate the destination. Neighbors again broadcast RREQ to their neighbors. The process continues until either the destination or an intermediate node with a fresh route to the destination is located. At each intermediate node, a reverse path is created for the source. It must be noted that several reverse paths may be created in this process. The source receives RREPs from all these paths. In AODV, it selects the one with minimum hop count and others are discarded. However, in RAODV, at this point Phase II starts.

In Phase II the source node sends an RRDU packet to all the nodes from which it gets the RREPs. Now since replies to RRDU, i.e. RRDU_REP packets are generated only by the destination and there is no impersonation, the source node will receive a unique RRDU_REP and the path discovery is completed. Here note that if there was a malicious node on any of those paths discovered by RREPs then that path would be isolated as there will be no RRDU_REP from that path. Once a path free of malicious node has been discovered, RRDUs are sent periodically to maintain the reliability of the path, i.e. to detect if any misbehaving node has crept into the path. RRDU-ID is incremented every time a new RRDU packet is sent by the source.

To maintain the reliability of the path we have introduced the field FDPC in the routing table as well as in the RRDU_REP packet. FDPC in the routing table keeps a count of the number of data packets forwarded by the node. This FDPC is copied by the node, on return, in the RRDU_REP packet to tell its previous neighbor as to how many data packets it has forwarded. The neighbor uses this FDPC to detect whether the node has forwarded all its packets or not. In case not, it detects that the node is selfish. Since selfish node does not intend to disrupt the functioning of the system, it does not lie. First time RRDU is sent only to discover the path and FDPC in RRDU_REP has no significance so it is set to zero by all the nodes. Once a path has been discovered RRDUs are sent periodically and FDPC field is used to discover if any selfish node has crept into the discovered path. In case a selfish node is detected on the discovered path, a fresh route discovery is initiated.

4.1. Route Discovery – Phase I

When a source node desires to send a message to some destination node and does not already have a valid route to the destination, it initiates a path discovery process.

TABLE II. RREQ MESSAGE FORMAT

Type	Hop Count	RREQ ID	Destination IP Address	Destination Sequence #	Originator IP Address	Originator Sequence #

The format of RREQ is same as that in AODV. Each node maintains its own sequence number as well as RREQ ID. The RREQ ID is incremented for every RREQ the node initiates. When an intermediate node receives an RREQ it does the following steps:

1. If it has a fresh route to the destination then it replies to the source with RREP else it broadcasts (forwards) the RREQ to its neighbors with hop count incremented by 1. If additional copy of the same RREQ is later received, these packets are discarded.
2. Sets up a reverse path for the reply message.
   (a) If it has an entry in its routing table for the source as the destination but it is not fresh enough it refreshes it. If there is an entry for the destination in RL, delete it.
   (b) If there is no entry for the source in the routing table it creates a new entry to the source node by copying the hop-count, source sequence number from the RREQ packet and address of neighbor from which first copy of the broadcast packet is received, as the next hop.
3. In either case, append an entry (destination, 0,0) in RL with IP address of destination copied from the destination field of the RREQ packet and FDPC, RRDU-ID set to zero.

TABLE III. RREP MESSAGE FORMAT

Type	Hop Count	Destination IP address	Destination Sequence #	Originator IP Address	Life Time

When the destination receives RREQ packet it sends back RREP using the reverse path. RREP may also be sent by an intermediate node having a fresh route to the destination. The format of RREP is same as that of AODV. On reverse path, each node receiving the RREP message does the following steps:

1. If it has an entry in its routing table for the destination but it is not fresh enough it refreshes it. Else, if it doesn't have an entry for the destination creates a new entry for the destination.
2. In either case, append an entry (source, 0,0) with IP address of source copied from the originator field of the RREP packet and FDPC, RRDU-ID set to zero.
3. Forwards it to the next hop on the reverse path.

In AODV, the path discovery is completed when the originator receives the RREP. Now, if I is a 'malicious node' then it may send RREP without having a route to the destination declaring that it has a fresh route to the destination and sets up a wrong path from the source to the destination through itself. Refer to Figure 1. In AODV, if B receives the first RREP from I, it will keep the path through I and discards others if the hop count of others is more. Hence *s* will have the wrong information. It will have an entry for the destination *d* with hop count three and next hop B. At this point, AODV has selected the path through I and the path discovery is completed. In RAODV, Path Discovery is completed only after sending RRDU packets and receiving RRDU_REP packets as explained below.

4.2. Route Discovery Phase-II

Once the source node receives RREPs, it sends an RRDU packet to all the nodes from which it gets the RREPs. The format of RRDU message packet is as follows

TABLE IV. RRDU MESSAGE FORMAT

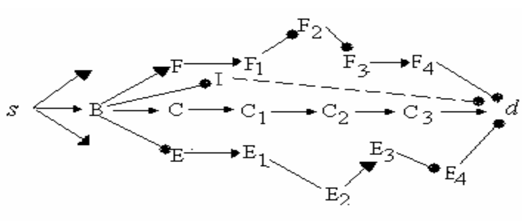

Figure 1.

Type	RRDU-ID	Source IP address	Source Sequence #	Destination IP	Destination Sequence #	Hop count

Let us denote the source node by "*s*" and the destination by "*d*". Every node receiving RRDU does the following steps:

1. If there is a reverse path entry for *s* then it sets the RRDU-ID by copying it from the RRDU packet. Else, it creates an entry for *s* in its routing table in the same manner as it is done on seeing RREQ.
2. Forwards it to all the nodes from where it had received RREPs earlier. For example, in Figure 1, B sends RRDU to F, I, C and E.
3. Each node on the path of RRDU must be having a table entry for the destination. Denote this entry by E_d. The node sets RRDU-ID in the RL entry for *s* in E_d. If additional copy of the same RRDU is later received, these packets are discarded. Note that it is possible for a node to receive multiple RRDUs.

When the destination receives the RRDU packet. It replies with RRDU_REP to the neighbor from which it received the first RRDU packet and discards others. The format of RRDU_REP is as follows

TABLE V. RRDU_REP MESSAGE FORMAT

Type	Source IP address	Destination Sequence #	Destination IP Address	FDPC	Reliability Flag

The destination sets the reliability flag in the RRDU_REP packet to 1. Note that on the reverse path, each intermediate node receives only one copy of RRDU_REP. Each intermediate node receiving the RRDU_REP message for the first time (RRDU-ID = 1) does two steps: sets FDPC in RRDU_REP to zero and forwards it to the next hop on the reverse path. Eventually, the source node gets RRDU_REP. Since no intermediate node can generate RRDU_REP, this RRDU_REP is unique and the path is discovered.

Now suppose that there is a malicious node I as shown in Figure 1, then it may send RREP without having a route to the destination. Also as it does not have to check its routing table it is the first one to respond to route discovery request i.e. it sends an RREP with wrong information declaring that it has a fresh route to the destination. We send RRDU on this path. However, since no node other than the destination can generate a reply to RRDU therefore B does not receive RRDU_REP from I. Also, B will receive RREP from C, E and F. It will send RRDU to all of them, and receive RRDU_REP from one of them, and a reliable path, which is free from a malicious node, is discovered.

Next, the source node starts sending the data packets on the discovered path. For every data packet forwarded by a node, FDPC in RL entry for *d* in E_s is incremented. As mentioned earlier, to maintain reliability, RRDU packets are sent periodically. Next time when the RRDU packet is sent and its reply received, each node sets the FDPC to the number of data packets it forwarded earlier.

Now, suppose that the path discovered in Figure 1, is *s*-B-C-C_1-C_2-C_3-*d* and let, that a node on this path becomes selfish say C_1. Let C_1 be a selfish node of Type 1. So all nodes other than C_1 forward the data packets that they receive. Suppose *s* sends *n* data packets to *d* before sending next RRDU. Then B and C forward all the *n* data packets to the successor. Let C_1 forwards only *p* out of them. Then C_2, C_3 forward *p* packets and the destination receives only *p* packets. After some time, *s* sends another RRDU and *d* sends back its acknowledgement RRDU_REP. This time FDPC is set to *p* by the destination and at every node *x* on the reverse path from the destination to the source, FDPC in RRDU_REP is set to the number of data packets forwarded by *x* on the forward path (copied from the entry for *d* in E_s). Hence, C_3, C_2, and C_1 set FDPC to *p* and C and B set the FDPC in RRDU_REP packet to *n*. When C sees that it had forwarded *n* packets to C_1 but C_1 forwarded only *p* ($<n$) out of them it knows that C_1 is selfish and it sets the Reliability Flag in the RRDU_REP packet to 0. When the source receives this RRDU_REP packet with RF set to zero it knows that something is wrong on this path and it initiates a fresh route discovery. This time C ignores C_1 in broadcasting the RREQ packet to its neighbors. The previous set of data packets are sent again.

In case C_1 is a selfish node of Type 2. It does not forward control packets. Hence, it does not forward RREQ initiated by *s* for route discovery and path between *s* and *d* is not established through C_1.

When C_1 is a selfish node of Type 3 then at first its energy falls in (E, T_1) and node behaves properly. It forwards both data packets as well as control packets. After sometime energy falls below T_1 in the interval (T_1, T_2), the node behaves like Type 1 and is handled similarly by initiating a fresh route discovery avoiding C_1. Next, we have two scenarios. One, C_1 recharges after sometime and second, its energy level falls below T_2. In the second case, we don't have to do anything since we have already discovered a path ignoring C_1. However, in the first case, we would like to send our data packets again through C_1 as path through C_1 is a shorter path. Since C_1 recharges after a fixed interval of time, say T, we will send fresh RREQs after T time, this time through C_1 to let C_1 also participate in packet forwarding. The entire process is repeated with RRDU_ID reset to 1.

4.3. Route Maintenance

As in AODV, RAODV also broadcasts periodically HELLO messages to its neighbors. Besides Hello Packets, we also send RRDUs periodically to check if any intruder has come in the discovered path or any node has changed to selfish.

5. PEFORMANCE COMPARISON OF AODV and RAODV

RAODV behaves as AODV in the absence of attack by malicious node. If there is no malicious or selfish node present in the network, the source will get one or more RREPs. It will choose a path for sending data packets as that of AODV. In case, a malicious or a selfish node is present in the system, AODV fails. On the other hand, RAODV will find an alternate path after a loss of few data packets. The lost packets are retransmitted. Hence RAODV outperforms AODV in presence of attack. After some time if some reliable node comes in place of the malicious node and there is some shortest path to destination through it, the path through this node is included for all future communication when fresh RREQs are sent. Hence RAODV recovers from the attack and start behaving like AODV.

6. CONCLUSION

In this paper we have suggested a protocol based on AODV protocol to prevent attacks by malicious nodes and selfish nodes. Here we handle malicious nodes, which are either black nodes i.e. they advertise short route to the destination through themselves without even having a route to the destination or attack by replaying old routing information and lack of error messages. This protocol is simple to implement, without any special hardware requirement. We have made an assumption that a node cannot impersonate. Since several methods have been proposed to take care of impersonation, this assumption is not impractical. Many previous routing protocols also have considered the attacks by misbehaving nodes. Some handle malicious nodes but not selfish nodes and some handle selfish nodes but malicious nodes not so nicely. The RAODV protocol provides a foundation for secure operation with little impact on existing protocols of an ad hoc network and can be used in bandwidth constrained nodes. RAODV actually outperforms AODV in terms of secure path discovery, although it does create more overheads in terms of computation power by maintaining Reliability List. Our protocol does not handle the case in which malicious node changes some fields in control packets, in particular the destination address.

We are working to implement the RAODV in Network Simulator and results are expected soon. In future, we plan to extend RAODV so that malicious node cannot modify the fields in the control packets.

7. References

[1] Ram Ramanathan and Jason Redi, “A brief overview of Ad-hoc Networks: Challenges and Directions”, IEEE Communications Magazine May 2002, pp. 20-22.

[2] Charles E. Perkins and Pravin Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, August 1994, pp. 234–244.

[3] Charles E. Perkins, Elizabeth M. Belding Royer and Samir R. Das, “Ad-hoc On-Demand Distance Vector (AODV) Routing”, Mobile Ad-hoc Networking Working Group, Internet Draft, February 2003

[4] Yih-Chun Hu, David B. Johnson, and Adrian Perrig, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks”, Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), IEEE, Calicoon, NY

[5] Seung Yi, Prasad Naldurg, Robin Kravets, “A Security-Aware Routing Protocol for Wireless Ad Hoc Networks”, Proceedings of 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI), Orlando, FL, July 2002, pp. 286-292

[6] Levente Butty´an and Jean-Pierre Hubaux. “Enforcing Service Availability in Mobile Ad-Hoc WANs”, Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA, August 2000.

[7] Pietro Michiardi and Refik Molva, “CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks”, Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia., 2002.

[8] S. Buchegger and J.-Y. Le Boudec, “Performance analysis of the CONFIDANT protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)”, Proceedings of The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing, 9-11 June, 2002, Lausanne, Switzerland, ACM Press, 2002, pp. 226–236.

[9] P. Michiardi and R.Molva, “Simulation based Analysis of Security Exposures in Mobile Ad Hoc Networks”, European Wireless 2002 Conference, 2002.