Minimized Overhead and Administrator based Secure Routing Protocol

Himadri Nath Saha
Assistant Professor
Department of Computer Science and Engineering,
Institute of Engineering and Management, West Bengal, India

Dr. Debika Bhattacharyya
Professor
Department of Computer Science and Engineering,
Institute of Engineering and Management, West Bengal, India

Dr. P. K.Banerjee
Department of Electronics and Communication Engineering,
Jadavpur University
West Bengal, India

ABSTRACT
As an emerging new technology a wireless communication allows users to access services electronically, wherever they might geographically be positioned. A MANET (Mobile Ad Hoc Network)is a special wireless network without any fixed infrastructure and It has dynamic topology. In this paper we have discussed a new secure routing protocol named as minimized overhead and admsinistrator based secure routing protocol(MOAP) for data packets, which is signifiacndly different from existing routing protocol. We have reduced the amount of network activity for each node required to route a data packet. We have utilized our algorithm to implement this protocol and then simulate with diffentent test cases. Finally we have discussed how this protocol prevents various attacks which may jeopardize any wireless network.

Keywords
Administrator; associative node; special associative node pair; traversed Administrator field; watch nodes; hello flooding**.**

1. INTRODUCTION
Mobile communication[2] differs a lot from the wired communication. The communication in case of MANET is mainly based on the radio signals transmitted by the node. Again MANET, being a wireless network, is quite different from the common mobile communication. In mobile communication bridge networks within its own range are used by the nodes to communicate with other nodes. The bridge networks act mainly as base stations which the source node needs to contact while sending a data packet to its destination.

We need to remember that the nodes are constantly moving and thus when a node goes out of the range of a base station it must contact its new base station which it finds in its range. This is called **Handoff.**

But in MANET there is no base station or any other infrastructure, helping to setup or perform the network activity required. Thus in this case the nodes are the routers transferring the data packets themselves[25]. Hence a robust and good routing protocol that will perform all the functions but with an optimized network activity to decrease the network traffic as well as make the transmission fast is very essential[24]. Thus while building our routing protocol we kept in mind these two factors – making the transmission fast and decreasing the network traffic.

This paper deals with how the entire network needs to be setup from the start, algorithms required to implement the protocol and finally implementation of the entire network using snapshots of a network showing how the algorithm works when a data packet is sent from one node to another.

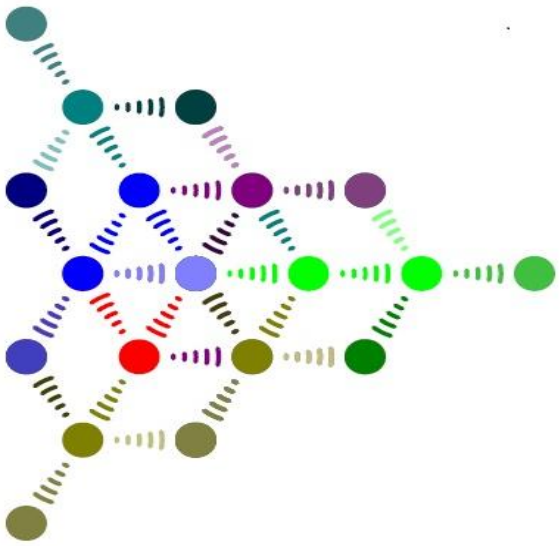

Figure 1: A mobile Ad Hoc Network

2. RELATED WORK
S. Matri[3] proposed to trace malicious nodes by using watchdog. According to this system, whenever a node forwards a data packet, the watch dog of the node checks whether the next node which receives the packet also sends the packet by listening to the broadcast signal of the next node. If the next node does not forward the packet within a predefined threshold time, the watchdog detects malicious behavior and accuses the node for aberration. This proposal has two shortcomings:

1. To monitor the behavior of nodes two or more hops away, the watch node has to trust the information from other nodes, which introduces the vulnerability of malicious activity.
2. The *watchdog* cannot differentiate between misbehavior and ambiguous collisions, receiver collisions, controlled transmission power and other such false alarms that might be generated during the data sending through the network.

We have used this concept in the form of a watch node in this protocol and have tried to eliminate the difficulties plaguing the watchdog by associating two watch nodes to each admin node.

Gonzalez [4] presents a methodology, for detecting packet forwarding misbehavior, which is based on the principle of flow conservation in a network. That states that if all neighbors of a node v_j are queried for

i. The amount of packets sent to v_j to forward and
ii. The amount of packets forwarded by v_j to them,

The total amount of packets sent to and received from v_j must be equal. They assume a threshold value for non malicious packet drop. A node v_i maintains a table with two metrics T_{ij} and R_{ij}, which contains an entry for each node v_j to which v_i has respectively transmitted packets to or received packets from. Node *vi* increments T_{ij} on successful transmission of a packet to v_j for v_j to forward to another node, and increments R_{ij} on successful receipt of a packet forwarded by v_j that did not originate at v_j. All nodes in the network continuously monitor their neighbors and update the list of those they have heard recently. The algorithm requires fewer nodes to overhear each others' received and transmitted packets since it uses statistics accumulated by each node as it transmits to and receives data from its neighbors. Since there is no collaborative consensus mechanism, such an algorithm may lead to false accusations against correctly behaving nodes.
Himadri[25],[26],[27] proposed different scheme for detecting and mitigating different attacks in MANET.

3. THE SCHEME

Every node in a MANET has a range of itself i.e. no node is capable of transmitting a data packet to an infinite distance. The nodes which fall in the range of a particular node are called its Neighboring nodes. In our algorithm we have alternatively used friend nodes for neighbor nodes, both of them being the same. In the network, three types of nodes have been used:
1. Common nodes
2. Associative nodes
3. Administrator nodes
4. Watch Nodes

The classification is based on the range and the position of the nodes in the network. But to understand classification we firstly need to understand how the entire network is set up. After a stipulated time period each node checks for its neighboring nodes, i.e. which nodes are present within its range. From this friend list, a list is prepared which contains all the neighboring nodes for all the nodes in the network.
Next, the node compares its previous and present list to check for network change and reports any difference to its administrator. The administrator node always lies in the range of the node in question.
We will describe how an Administrator node is chosen later. If there is no change in the topology of the network, there is no need to choose an administrator node; however for any change in the network, the previous Administrator nodes will choose a new Admin node which leads us to the discussion on what an Admin node is and how it is selected.

3.1 Administrator nodes

Now the topic of selecting a new Admin arises. If there is a need to elect a new Admin, all the nodes send their neighbor's list to their Admins(old admin) which all the Admins exchange among themselves thereby giving each Admin the knowledge of the neighbors of each and every node in the entire network. A list with the names of all the nodes and their corresponding neighbors written beside them is prepared and sorted according to the highest number of neighbors each node has and all the possible nodes are selected in a top-down sequence. If the list of names of all the nodes is a subset of the neighbor list of that node we will designate it as the Admin node else we have to take any two nodes from the comprehensive sorted list in a top-down sequence. Next the union of the neighbors of the two selected nodes is considered, if the entire list results as a subset of the union those two nodes are considered as the Admin nodes. However for a negative result we take any three nodes from the sorted comprehensive list in a top-down sequence and continue the process. We continue this process increasing the number of nodes considered by unity until the subset criteria is satisfied after which the Admins are selected as those nodes whose friend list covers the complete network.

3.2 Assotiative and common nodes

The Associative nodes are nodes lying in the region common to multiple Admin nodes. If any Admin node does not have an Admin node or an associative node attached to it, then an associative node pair is selected. It is a pair of nodes through which Admin can communicate with the next Admin node. All the nodes in the network excepting the Admin nodes and the associative nodes are the common nodes.
Let us consider the following network:

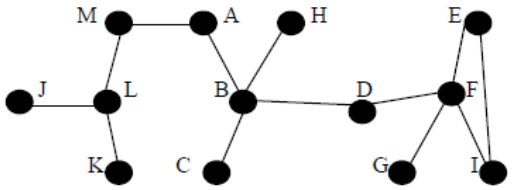

Figure 2: A snapshot of an Ad-Hoc network

For the above network in fig. 1 the neighbor list has to be prepared first and only then can we assign the admin nodes. The neighbor list is as follows:

Nodes	Friends
A	A,B,M
B	A,B,C,D,H
C	B,C
D	B,D,F
E	E,F,I
F	D,E,F,G,I
G	F,G
H	B,H
I	E,F,I
J	J,L
K	K,L
L	J,K,L,M
M	A,L,M

The list in top down sequence is as follows:

Nodes	Friends
B	A,B,C,D,H
F	D,E,F,G,I
L	J,K,L,M
A	A,B,M
D	B,D,F
E	E,F,I
I	E,F,I
M	A,L,M
C	B,C
G	F,G
H	B,H
J	J,L
K	K,L

As we choose the admin nodes, it is clearly visible that node B alone does not cover the whole network, its neighbor nodes A, B, C, D, H are not all the nodes of the network. Hence we choose a pair of nodes and their union is considered. Hence F along with B is chosen and the resultant union gives us the nodes A, B, C, D, E, F, G, H and I. However still some nodes are missing, so we need to take a third node for union. We take L and get all the nodes. Now we can see node D is neighbor of both admin nodes B and F. Hence D acts as the associative node for admin nodes B and F. It is clear that there is no associative node attached to admin node L. So we choose A, M pair by which admin L can communicate with admin node B. So A, M pair is called the special associative node pair.
Hence in our example,
Admin nodes: B, F, L
Associative node: D
Common nodes: A, C, E, G, H, I, J, K, M
Associative node pair: A, M

3.3 Watch nodes

Watch nodes have been used basically to promote security in the network. As is the case with our protocol, if we are able to implement security in the admin and the associative nodes, we can guarantee that the total network is secured from any attacks since the common nodes do not have any role to play in transmission of data apart from sending or receiving of data packets. Hence we have added two watch nodes to each admin and Associative node which checks after every time interval if any data packet entering into an Admin or Associative node goes out of the node within a stipulated time period, failing which it issues a warning to its previous admin that the node maybe a malicious node and the node is not used for transmission of data through the network. We have used the two neighboring nodes of each admin as the watch nodes in our protocol.
In the network in fig.1 if a data packet is sent from node A to node I then the path followed will be
A-B-D-F-I
When the data packet is at admin B, any two nodes from A, C, D and H will be selected as watch nodes and will keep an eye on the admin node B for any aberrant behavior.

This protocol also has certain aspects such as battery life, admin reselection and back tracking.

3.4 Battery life

We realize that the nodes in an Ad-hoc network are constantly in motion and hence will run out of battery power sooner or later. However for our protocol, the battery life is particularly important for the Admin nodes since they perform the maximum amount of work. Hence when the power of a certain admin decreases below a certain level we need to get that battery to re-charge before it can take part in any re-transmission. We have developed our protocol in such a way that for every admin, there is a special field for the battery life and if the threshold value of the battery of an admin is reached, it immediately withdraws itself from the network and recharges and admin reselection takes place. After recharging it can be again reconsidered in the network.

3.5 Admin reselection

There may be a few cases when admin reselection is required. If an admin node is found to be malicious it is blocked from the network immediately and the network chooses a new admin again in the same way as described above. Again if the battery power of an admin gets drained off completely then it is suspended temporarily from the network for recharging and admin reselection takes place.

3.6 Backtracking

In many cases it may so happen that when a data packet reaches an admin, it has multiple paths to move to. The data packet may choose one path, but the destination may very well be on the other path. In such cases a back tracking is required. We have added two bits along with the data packet which records the last traversed admin and then forwards the data packet, if the data packet does not find its destination on the traversed path then it backtracks to the admin where it finds a multiple path and then moves to the other alternative path. In this way we re-transmit the data from an admin with multiple paths to reach the destination.

4. ALGORITHM

4.1 Select_admin

Step 1: Every node which enters the network broadcasts hello packets //new node insertion
Step2: If there is no response then
There is no need to flood the neighbor list
The node itself is the admin
Step 3: If there is any response then
Update the friend list
The nodes flood their neighbor list across the network
Send a special request for presence of admin in the network //Admin_present = 1 or Admin_present = 0
Step 4: If there is no admin // Admin_present = 0
The node with the least ID number calls Compute()
The result is flooded across the network
Step 5*:* Else if there are previous admins
//Admin_present = 1
Then the previous admin with the least ID number will call Compute()
The result is flooded across the network
Step 6: After a certain time period every node in the network broadcasts hello packets //check for deletion or relocation of nodes
Step 7: Continue from Step 2

4.2 Compute()

Step 1: Sort the friend list in descending order of number of friends
Step 2: union_result = 0
Step 3: While (union_result is a subset of entire_list)
Check neighbor list
Take next highest entry of list in descending order
Union_result = Union_result + neighbor_of_node[i]
if (union_result = entire_friend_list)
Set the nodes as admins
End If
End While
Step 4: Nodes common to multiple admins are associative nodes
Step 5: Other nodes are common nodes

4.3 Packet_sending

Step 1: If sender = receiver
Sender is same as receiver, so packet will always be successfully sent
Else Step 2
Step 2: Packet is sent to the admin of the node
Set sender=admin
Traversed_admin_field =sender

If multiple path possible from current admin
Then
Set Bactracking_bit = 1 for current admin
/ Backtracking_bit is a bit field for every admin node whose value indicates whether backtracking is possible or not from that node onwards */*
End If

/ Traversed_admin_field is a one dimensional array which stores the id of the admin nodes which have already been traversed by the data packet. This is used to prevent loopback.*
*When a packet reaches a node, it checks the Traversed_admin_field for the next admin's entry. If it finds the admin in the array, then it checks for the value of the Backtracking_bit. If Backtracking_bit=1, only then it allows the packet to move to the next admin */*
Step 3: While (packet is not sent to the receiver)
If sender = receiver
Packet is sent successfully
Generate and Send Ack
Else
If receiver is neighbor of admin
Packet sent
Generate and Send Ack
Else if receiver is not neighbor of admin
Packet sent to the next admin /* if current admin is within the range of the next admin */
Sender=next admin
Traversed_admin_field =sender
If multiple path possible from current admin Then
Set Backtracking_bit = 1 for current admin
/ Bactracking_bit is a bit field for every admin node whose value indicates whether backtracking is possible or not from that node onwards */*
End If
Else if current admin is not within next admin's range
Packet sent to associative node to send it to next admin
Traversed_admin_field =sender
If multiple path possible from current admin Then
Set Backtracking_bit = 1
End If
Else
Packet sent to special associative pair nodes to send it to next admin
Traversed_admin_field =sender
If multiple path possible from current admin Then
Set Bactracking_bit = 1
End If
If (next admin is not in traversed_admin field)OR (admin is in traversed_admin_field and Backtracking_bit=1)
Send packet to the next admin
Sender=admin
Traversed_admin_field =sender
If multiple path possible from current admin Then
Set Bactracking_bit = 1 for current admin

End If
ElseIf Bactracking_bit = 1 for current node
Try alternative path
Set Bactracking_bit = 0 for current node /* if all alternative paths have been exhausted */
End if
End if
End if
If all admins have been traversed atleast once but receiver not found /* receiver left the network or failed or no such receiver id exists*/
Then
Drop packet
Break while
End If
/* In this case, Sender does not get Ack and it assumes that either the packet was lost in transit and did not reach the receiver OR the receiver is not present in the network, hence retransmits the packet once more */
End While
End if

4.4 Watch_node

Step 1: While(data_packet is at an admin[j])
Set friend_node[i] and friend_node[i+1] of admin[j] as watch nodes
End While
Step 2: Set count = 0
Step 3: If packet_sending(admin[j]) == false
/* Watch_node monitors the admin network traffic pattern */
Watch_node detects malicious activity
Report activity
count = count +1;
If count == 3 /* if the watch_nodes detect malicious activity more than 3times */
remove admin from network
Select_admin();
End If
End If

4.5 Battery_Life

Step 1: Set threshold_value for battery life of each admin
Step 2: While battery_life > threshold_value
perform network activity
battery_life = battery_life – 1;
End While
Step 3: If Admin(battery_life < threshold_value) then
Remove admin from network
Recharge admin
Select_admin(if allowed by network topology)
Else
Remove admin from network
Recharge admin
Insert the recharged admin into the network
End If

4.6 Admin_Failure

If current admin crashes
admin reselection takes place
/*Only if network does not gets disconnected due to admin failure*/
Else
disconnected part of the network stalls until network topology changes or admin recovers, whichever is earlier
End If

5. SIMULATION RESULTS (using Java)

We have given the pictorial simulated results of our protocol. Before we check the simulation results of the protocol, we need to understand a few concepts.

- ✓ A green colored ellipse signifies that it currently holds the data and will forward the data to the next node.
- ✓ A red ellipse means it does not have a data and is free, i.e., it has either passed on the data, or is going to receive the data sometime in the future or will not receive a data during the data transfer.
- ✓ A black node indicates that it has been chosen as the administrator node for the network.
- ✓ A black line broadcasts the administrator nodes information to the common nodes by which it lets them know that it is the admin node for those nodes in the network.
- ✓ A green line will be drawn whenever there is a transfer of data from one node to another.
- ✓ A yellow line indicates transit of the final ACK.
- ✓ A red outer ellipse shows the range of the node.

Time instance I: When the first node enters the network

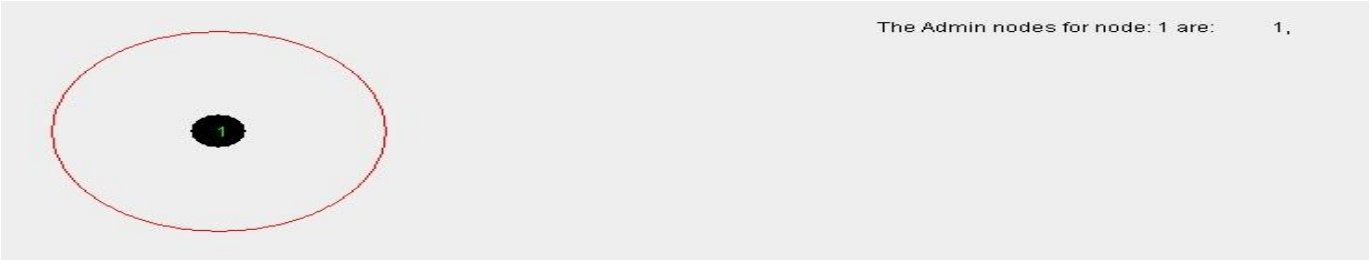

Figure 3: The first node on entering the network broadcasts a hello packet within its range. But as it gets no response it does not flood its neighbor list. It is its own admin.

Time instance II: Entry of the second node

Figure 4: New node on entering the network broadcasts a hello packet to update its friend list

Figure 5: In this case, either of nodes 1 or 2 can be selected as the admin. Here, Node 1 is selected as the admin. The previous admin, i.e., node 1, selects the new admin

Time instance III: A third node enters the network as shown

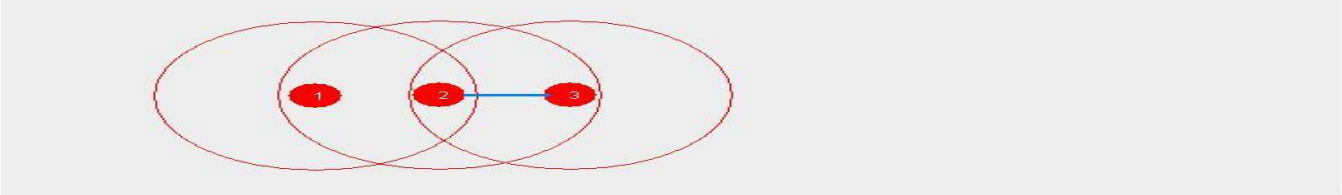

Figure 6: The third node on entering the network broadcasts a hello packet within its range to update its friend list

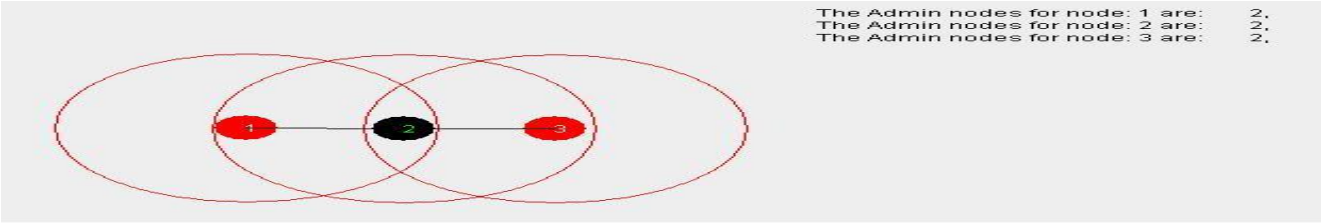

Figure 7: Node 2 is selected as the admin. The previous admin, i.e., node 1, selects the new admin.

Time instance IV: A fourth node enters the network as shown

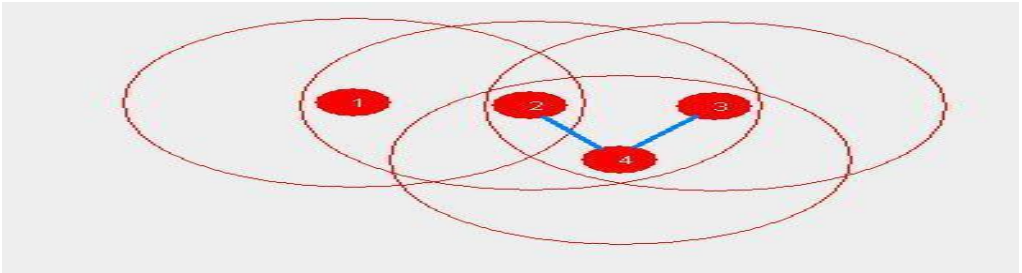

Figure 8: The fourth node on entering the network broadcasts a hello packet within its range to update its friend list

Time instance 'N': After a certain time interval, say n, let the network have the following configuration

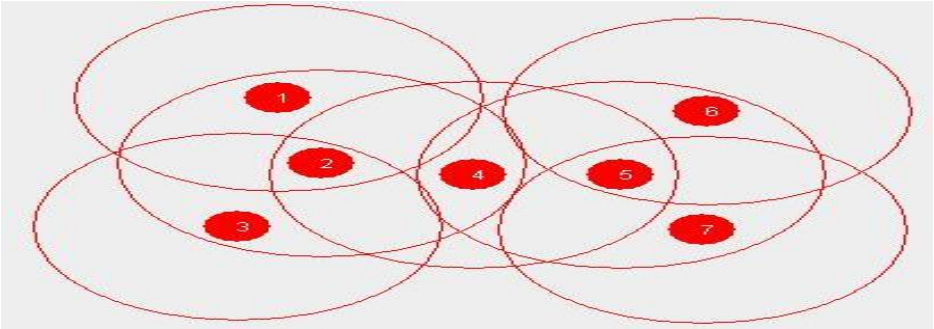

Figure 9: Network with 7 nodes

The Admin Nodes of the network are: 2, 5
The Associative Node in the network is: 4
The friend list is:

```
0 1 2 3 4 5 6 7
1 1 1 0 0 0 0 0
2 1 1 1 1 0 0 0
3 0 1 1 0 0 0 0
4 0 1 0 1 1 0 0
5 0 0 0 1 1 1 1
6 0 0 0 0 1 1 0
7 0 0 0 0 1 0 1
```

This is an adjacency list where 1 represents that node i is in the range of node j

Alloted Admin is:

```
1  2 0 0 0 0 0 0
2  2 0 0 0 0 0 0
3  2 0 0 0 0 0 0
4  2 5 0 0 0 0 0
5  0 5 0 0 0 0 0
6  0 5 0 0 0 0 0
7  0 5 0 0 0 0 0
```

1st column represents node number
& the following columns represent admin for the corresponding node. 0 indicates sentinel value.

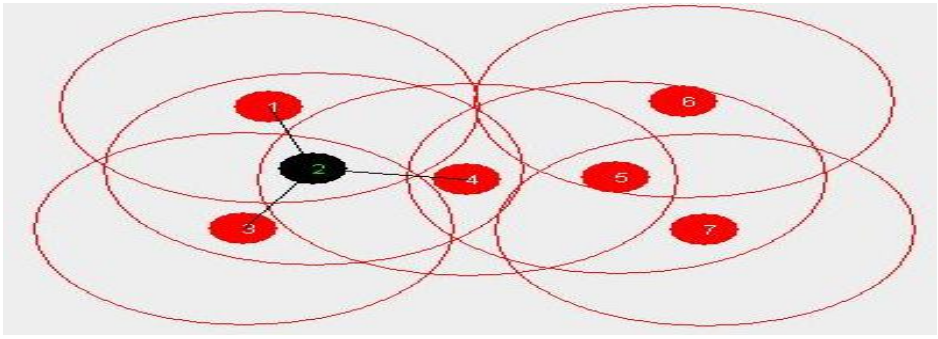

Figure 10: Node 2 is chosen as an Admin Node. It broadcasts this information to its neighboring common nodes

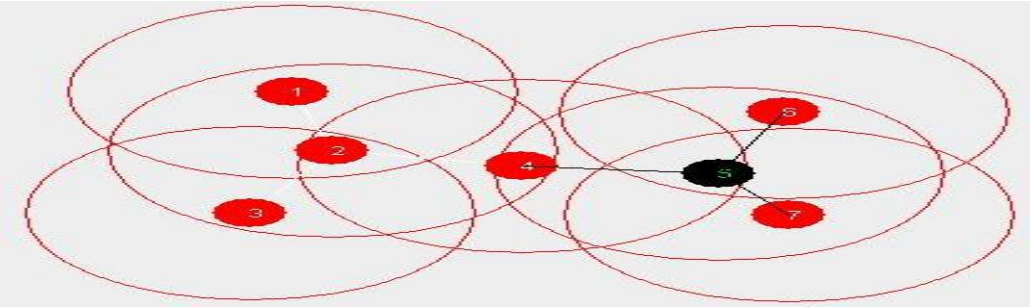

Figure 11: Node 5 is chosen as an Admin Node. It broadcasts this information to its neighboring common nodes

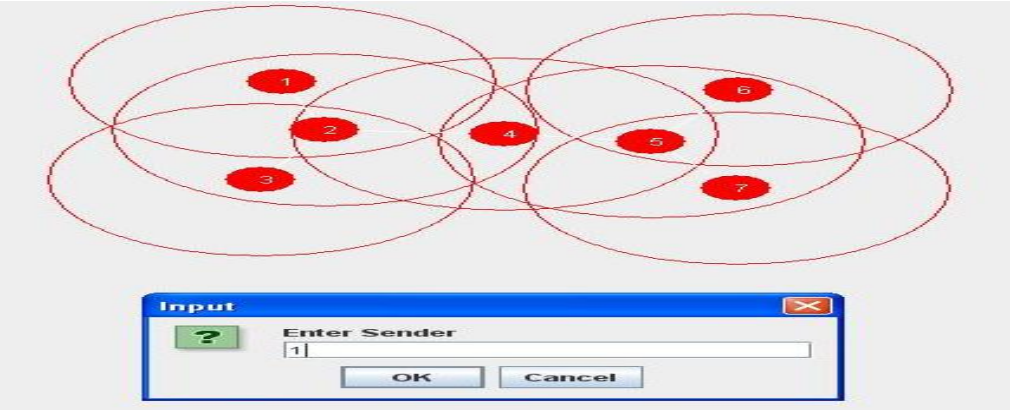

Figure 12: Node 1 is chosen as the sender Node

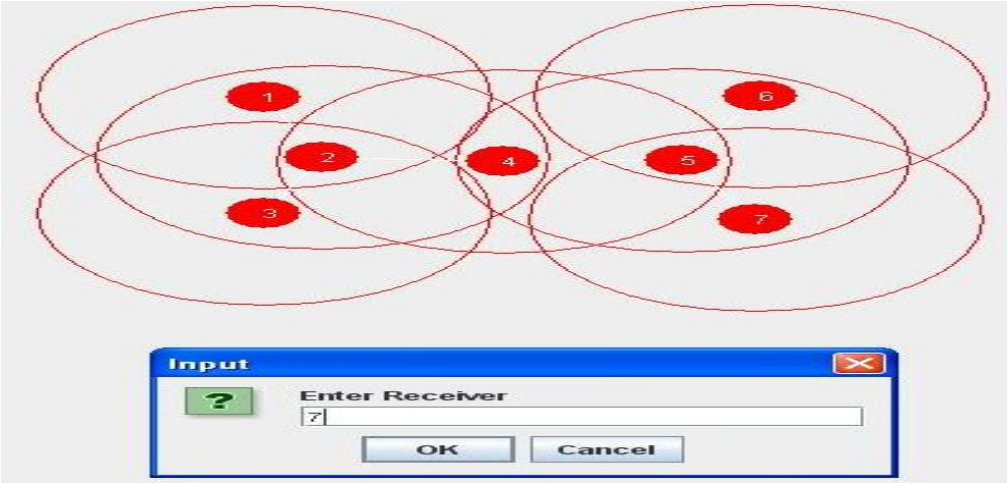

Figure 13: Node 7 is chosen as the receiver Node

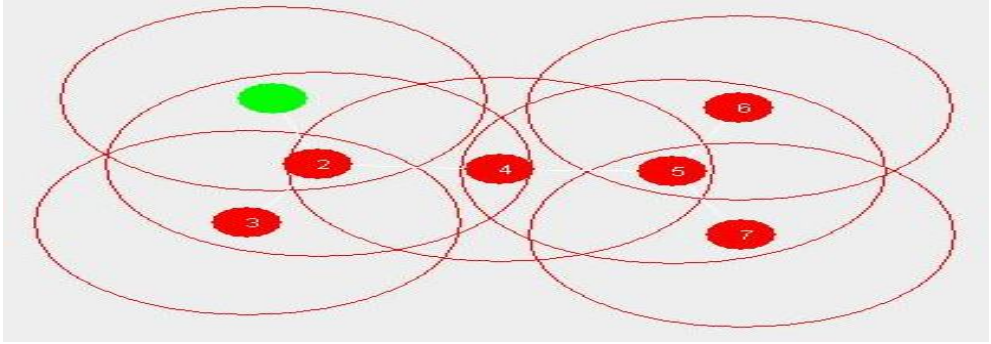

Figure 14: Packet sending starts from Node 1

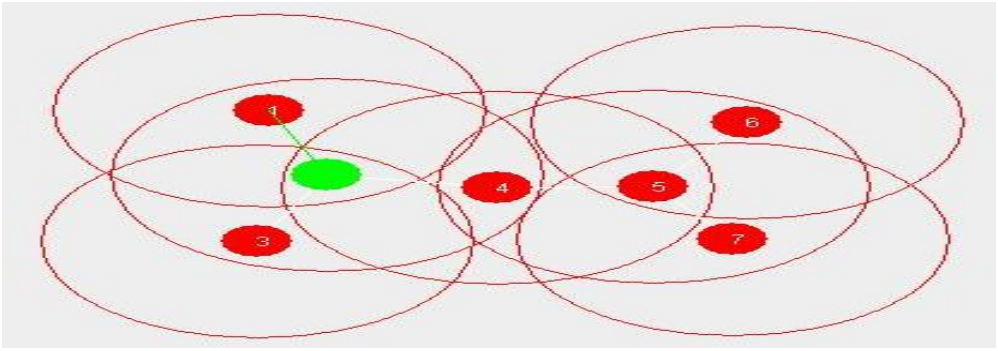

Figure 15: Packet is sent from Node 1 to its admin Node 2

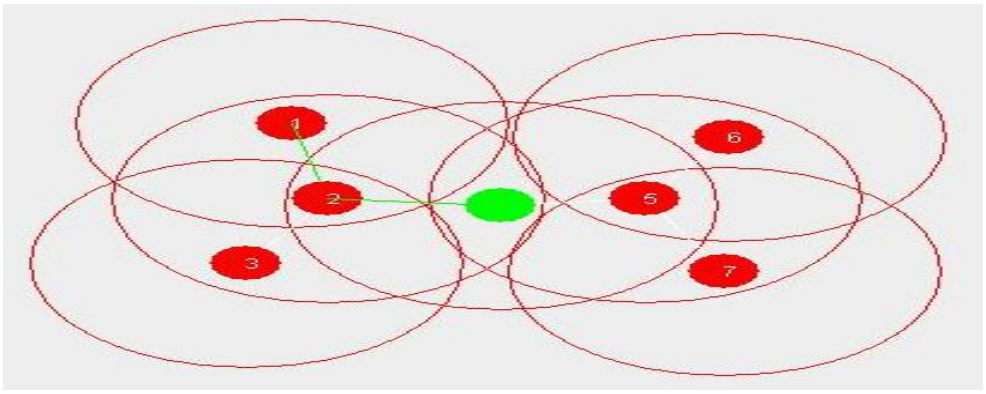

Figure 16: As Receiver 7 is not in the friend list of the admin hence the Packet is sent from Node 2 to the associate Node 4

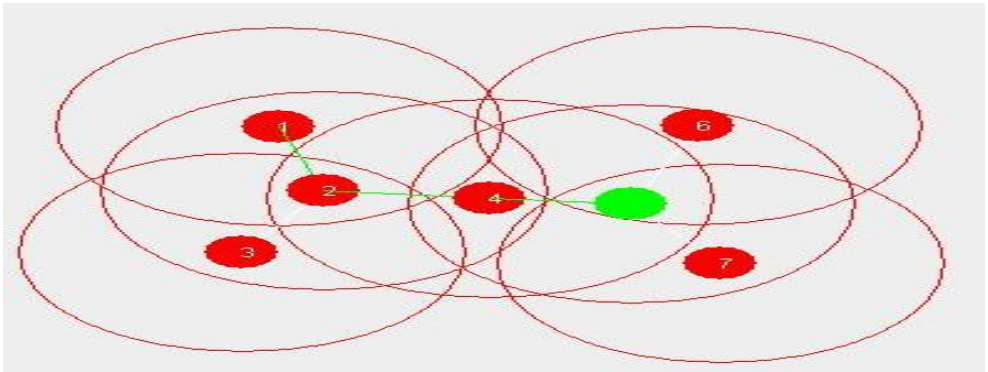

Figure 17: Associative node 4 sends the packet to the next Admin node 5

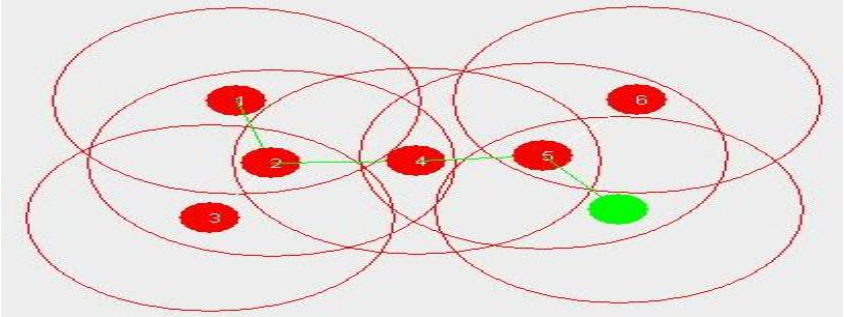

Figure 18: Receiver Node 7 is the neighbor of Admin Node 5; Node 5 sends the data packet to Node 7

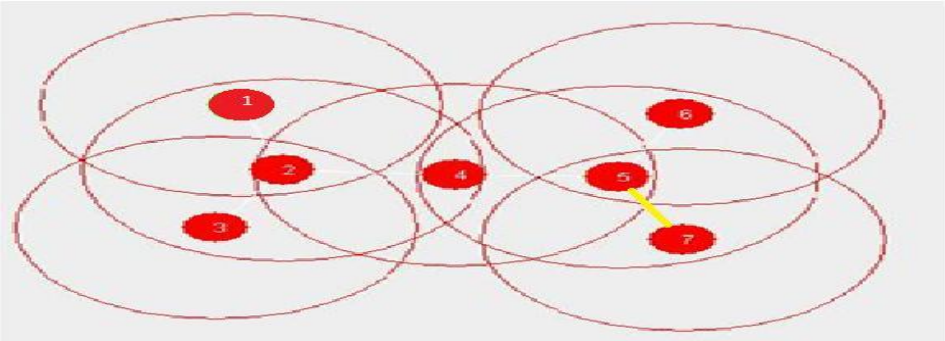

Figure 19: Final ACK generated by receiver 7 & send in reverse direction

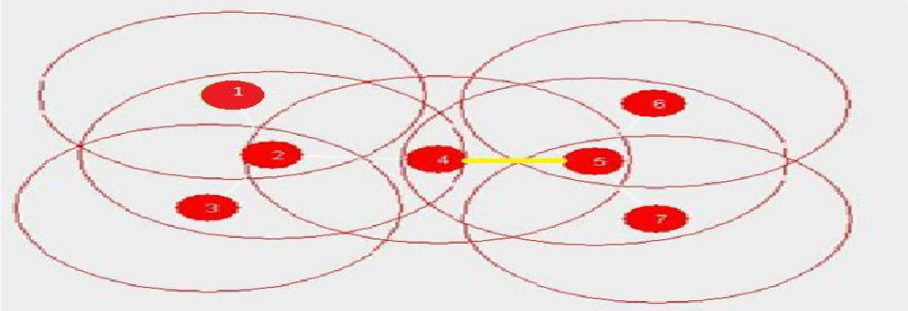

Figure 20: ACK send from node 5 to node 4

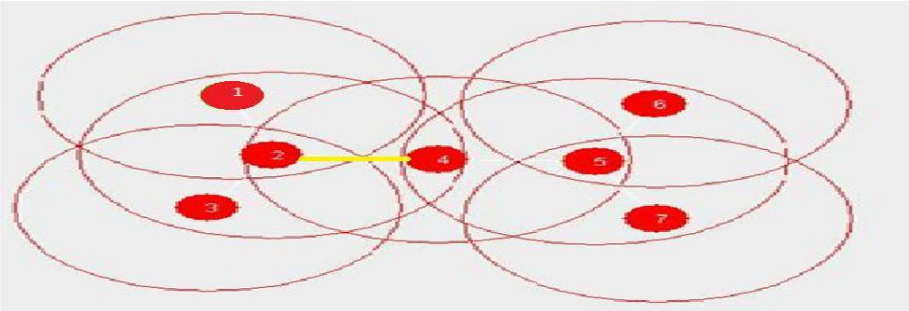

Figure 21: ACK send from node 4 to node 2

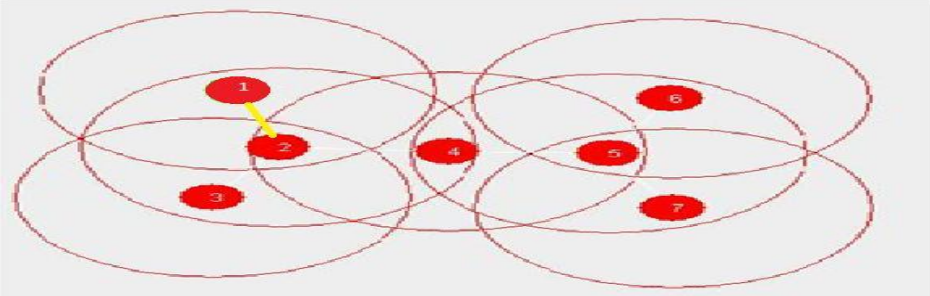

Figure 22: ACK send from node 2 to sender node 1

Figures 3-22 show the actual simulation of the protocol in a network and how the data packet moves in the network from the sender to the receiver node.

6. ATTACK SIMULATION

We have simulated the possibility of an attack on our network and shown how this protocol can detect malicious nodes and also prevent attacks. Now the nature of this protocol is such that if we can make sure that the admin nodes are secured, then we are certain that the network will perform perfectly without any network attacks since any common node only takes part in either sending or receiving data packets. We will now provide the pictorial representations of the attack simulation using watch-nodes.

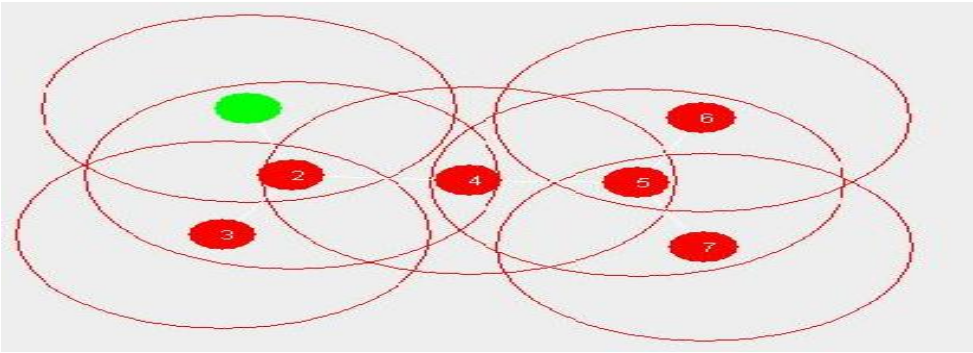

Figure 23: Node 1 is chosen as the sender node

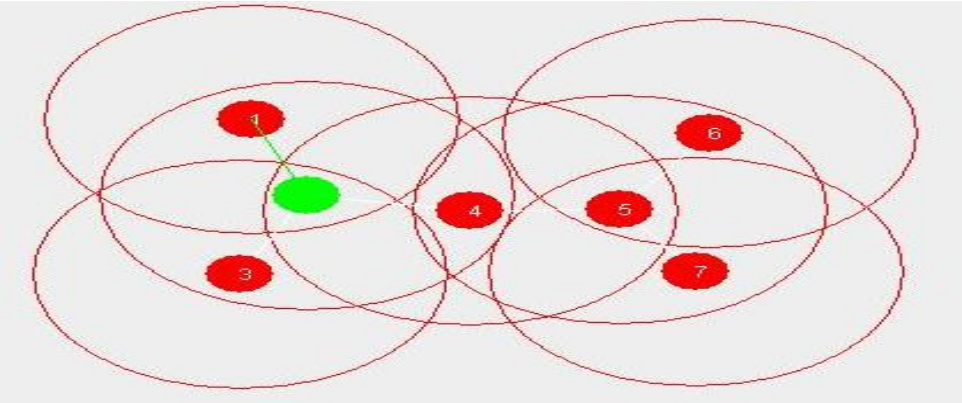

Figure 24: Node 1 sends the data packet to its admin Node 2

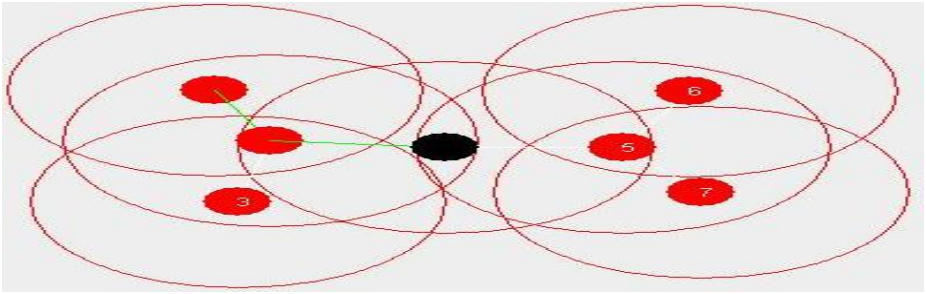

Figure 25: Node 2 sends the data packet to the next admin Node 3, which does not forward the data packet

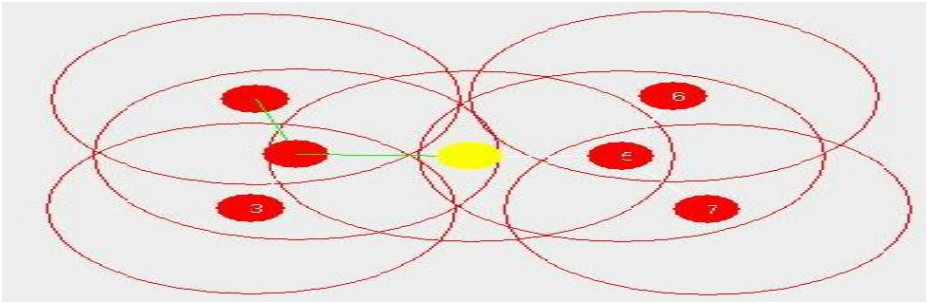

Figure 26: The watch nodes for Node 4 are Nodes 2 and 5, they detect malicious behavior and Node 3 suspected as a malicious node

7. MAINTENANCE OF SECURITY

For any protocol, maintaining security [5]is absolutely necessary. If it does not secure data transmission among, then there is no point in using the protocol simply because it does not guarantee the proper delivery of correct data to correct receiver. There are many security attacks possible on a network. The ones this protocol guards against are:

1. *Hello Flooding[6]:* In this attack a dishonest node sends a repeating message through the network causing network congestion. Our protocol deals with this attack by associating a timestamp with every data packet; if a data packet, having the same timestamp[7] value and same source node number containing the same data, is repeated then the receiving node will simply discard the data packet.

2. *Co-operative Black Hole attack [8]:* In this attack a group of dishonest nodes act as a black hole, i.e., when a node receives a data packet it circulates the packet among them without sending it out of the loop, hence the data packet never reaches the destination. This protocol deals with re-transmission in such a way so as to stop this attack. If there is a requirement of a data packet to be sent back along the path it had come, then a it is noted that the previous path does not have the receiver and the data packet can no longer go in that direction. In other words, our protocol avoids data being sent through loops in the network. Hence this attack is avoided.

3. *Black Hole attack [9]:* Black holes are those nodes in a network where incoming traffic is silently discarded but the source has no information that the data did not reach the intended recipient. This is one of the biggest security attacks that occur in a MANET.

   We have used the concept of watch nodes in our network. Watch nodes act as guardians and check if an Admin node is correctly forwarding a data packet or not. If it finds out that on multiple occasions a data packet is not being forwarded by an Admin node, it assumes that the Admin may be malicious and will simply discard[10] the Admin node from the network. Since this protocol is very lightweight and the computations depend solely on the admin nodes, hence the security of the admin nodes ensures that the network is secured.

4. *Gray Hole Attack:[11]* A gray hole is similar to a black hole but it starts its action after it has been the part of a network for some time. It will behave as normally when the network starts its functioning but after a certain amount of time it will either consume all or some of the data packets that come its way.
   Like the previous attack, the watch nodes once again detect if any Admin node does not successfully forward a data packet and if a node tries to act as a gray hole; it will suspend the node for aberration.

5.Sleep Deprivation[12]: The attacker attempts to consume batteries of other nodes by requesting routes, or by forwarding unnecessary packets. This protocol has no route request mechanism, the entire route is based on the dynamic nature and it is decided during the packet sending. Hence this stops sleep deprivation. However if unnecessary packets are sent, hello flooding attack is stopped by checking the time stamp value. If packets having different time stamps are sent, then it is very difficult to distinguish a real data packet from an unnecessary one, in such a case, the battery life is drained. This protocol is designed in such a way that if the battery life of an Admin node is below a threshold level, it simply disconnects itself from the

network until it can recharge itself. Hence although the battery power is drained off, the network activity does not stop.

8. COMPARISON WITH EXISTING PROTOCOLS

8.1 Comparison with AODV[13]:

AODV has a lot of network activity associated with it since there are routing packets transmitted all over the network to know the desired route. In this protocol however, the admin nodes take the duty of network transmission and so the overall load on the network decreases many folds. The network traffic depends on the dynamic nature of the network, lower the amount of changes in the network, lower will be the network traffic.

8.2 Comparison with DSDV[14],[15]:

For DSDV each node connects to all other nodes in order to maintain their routing table[16]. However in this case, the admin nodes do this work and so there is a very low routing maintenance required for the networks.

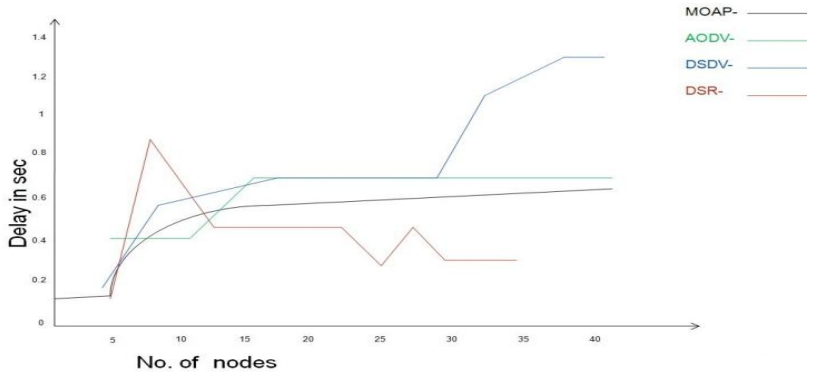

Figure 27: Graphical comparison of MOAP with existing protocols

This algorithmic complexity of $O(n^2)$ gives the parabolic nature[17] of the graph. As the number of nodes increases, the network congestion increases, gradually increasing the delay. But the delay is much stable compared to the other protocols.

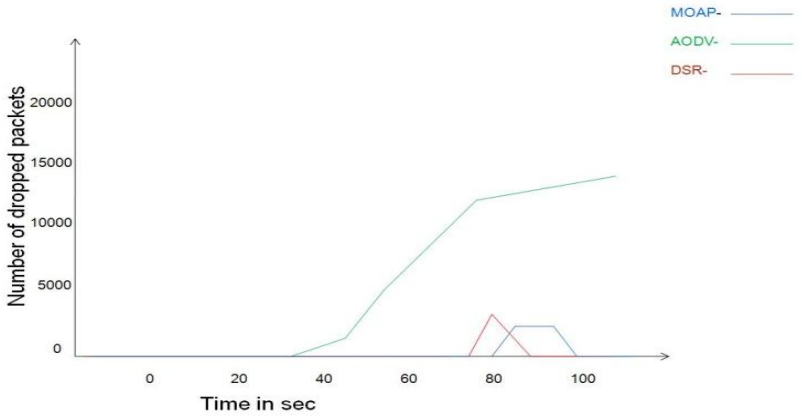

Figure 28: Packets Dropped by MOAP, AODV & DSR

In our protocol packets will be dropped only when the admin battery gets exhausted below a threshold[18]. In the meantime a new admin will be selected & the previous admin will be recharged. It performs consistently well.

9. CONCLUSION

The algorithm actually is a basic design built to reduce network overhead[19] and computations and also to ensure absolute security[20] of the data packet transmitting through the network and it performs very efficiently in that respect. The rate of topology [21]change must remain less or medium. If the topology changes constantly the protocol may be vulnerable. This protocol will be much more optimal[22] compared to the existing protocols such as DSDV[23] and also AODV[24] unless a small network is considered. There are various protocols which can send data very fast but then they have a lot of overhead attached to them. This protocol leads to a decent solution as it sends data at an optimal speed[25] while taking care of the computational overhead.

10. REFERENCES

[1] Izhak Rubin, Arshad Behzad, Ruhne Zhang, Huiyo Luo and Eric Caballero."TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks". In Proceedings of IEEE Aerospace Conference, 2002, vol. 6, pp. 2727-2740.

[2] Kejun Liu, Jing Deng, Member, IEEE, Pramod K. Varshney, Fellow, IEEE, and Kashyap Balakrishnan, Member, IEEE, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs" IEEE Transaction on Mobile Computing, VOL. 6, NO. 5, May 2007.

[3] S. Matri, T. J. Giuli, K. Lai and M. Baker. "Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks". In Proceedings of Mobicom 2000, pp. 255-265.

[4] Gonzalez- Computers & Security, Vol 25, No. 18, pp 736-744, 2000.

[5] Panagiotis Papadimitratos , Zygmunt J. Haas , "Secure message transmission in mobile ad hoc networks, Ad Hoc Networks" IEEE 2003, 193–209 .

[6] Y.-C. Hu, A. Perrig, D.B. Johnson, Ariadne: a secure on-demand routing protocol for ad hoc networks, in: ACM MobiCom'02, 2002, pp. 12–23.

[7] Y.-C. Hu, D.B. Johnson, A. Perrig, SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks, in: Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), 2002, pp. 3–13.

[8] Komala CR, Srinivas Shetty, Padmashree S., Elevarasi E., "Wireless Ad hoc Mobile Networks", National Conference on Computing, Communication and Technology, pp. 168-174, 2010

[9] L. Zhou and Z. Hass. "Securing Ad Hoc Networks", IEEE Network Magazine, vol.13, pp. 24-30, 1999.

[10] Yih-Chun Hu, Adrian Perrig and David B. Johnson, "Wormhole Attacks in Wireless Networks", IEEE Journal on Selected Areas in Communication, pp.370-380,2006

[11] [Johnson99] David B. Johnson, Davis A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks" October 1999 IETF Draft, 49 pages.

[12] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, E.M. Royer, A secure routing protocol for ad hoc networks, in: Proceedings of the 10th IEEE International Conference on NetworkProtocols, 2002, pp. 78–87.

[13] [Perkins99] Charles E. Perkins, Elizabeth M. Royer, Samir R.Das, "Ad Hoc On-demand Distance Vector Routing", October 99 IETF Draft.

[14] [Perkins94] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", Comp. Comm. Rev., Oct. 1994, pp.234-244.

[15] Samir R. Das, Charles E. Perkins and Elizabeth M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks"

[16] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", 2003

[17] Avoiding Black Hole and Cooperative Black Hole Attacks in Wireless Ad hoc Networks http://www.scribd.com/doc/26788447/Avoiding-Black-Hole-and-Cooperative-Black-Hole-Attacks-in-Wireless-Ad-hoc-Networks

[18] Shukor Abdul Razak, Steven Furnell, Nathan Clarke, Phillip Brooke: A Two-Tier Intrusion Detection System for Mobile Ad Hoc Networks- A Friend Approach, Springer-Verlag Berlin Heidelberg 2006.

[19] David B. Johnson, David A. Maltz and Josh Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks"

[20] Y. Zhang and W. Lee. "Intrusion Detection in Wireless Ad Hoc Networks". In Proceedings of Mobicom 2000, pp. 275-283.

[21] Rashid Hafeez Khokhar, Md Asri Ngadi and Satira Mandala, "A Review of Current Routing Attacks in Mobile Ad Hoc Networks", International Journal of Computer Science and Security, pp. 18-29, Volume-2 Issue-3 International Conference on Wireless Networks (ICWN'03), Las Vegas, Nevada, USA, pp. 570-575

[22] Jeremy J. Blum, Member, IEEE, and Azim Es kandarian, Member, IEEE, "A Reliable Link-Layer Protocol for Robust and Scalable Intervehicle Communications" IEEE Transactions On Intelligent Transportation Systems, vol. 8, no. 1, March 2007.

[23] N.Shanthi, Dr.Lganesan and Dr.K.Ramar,"Study of Different Attacks on Multicast Mobile Ad-hoc Network", Journal of Theoretical and Applied Information Technology, pp.45-51

[24] Luke Klein-Berndt, "A Quick Guide to AODV Routing"

[25] Jung-San Lee, Chin-Chen Chang, "Secure communications for cluster-based ad hoc networks using node identities" Journal of Network and Computer Applications 22 October 2006 International Journal of Computer Science and Security, Volume (1): Issue (1) 67.

[26] Himadri Nath Saha,Dr Debika Bhattacharyya,Dr.P.K.Banerjee,"A Priority Based Protocol for Mitigating Different Attacks in MANET",International Journal for Computer Science and Communication,Volume I,Number2,pp-299-302,Sept.2010

[27] Himadri Nath Saha,Dr Debika Bhattacharyya,Dr.P.K.Banerjee,"A Distributed Administration Based Approach for Detecting and Preventing Attacks in MANET",International Journal for Scientific and Engineering Reasearch,Volume-2,Issue-3,pp-1-11,Mar-2011

[28] Himadri Nath Saha,Dr. Debika Bhattacharyya,Dr.P.K.Banerjee,"Semi-Centralized Multi-Authenticated RSSI Based `Solution to Sybil Attack",International Journal of Computer Science and EmergingTechnologies",VolumeI,Issue-4,pp-338-341,Dec 2010