A novel and Efficient Secured Routing Algorithm in Mobile Ad hoc Network

Himadri Nath Saha #1 , Dr. Debika Bhattacharyya #2 , Dr. P. K.Banerjee *3

Assistant Professor #1, Professor #2, Professor *3

Department of Computer Science and Engineering, Institute of Engineering &
Management, West Bengal, India #1, #2

Department of Electronics and Tele Communication Engineering, Jadavpur university,
West Bengal, India *3

him_shree_2004@yahoo.com #1 , bdebika@yahoo.com#2

Abstract: *Currently the mobile wireless technology is experiencing rapid growth. However the major challenge for deployment of this technology with its special characteristics is securing the existing and future vulnerabilities. The lack of static infrastructure causes several issues in mobile Ad hoc network (MANET) environment, such as node authentication and secure routing. In this paper we propose a new approach for secure routing of data packets in MANET. This approach will reduce the computational overhead to a lot extent. The scheme is based on a specific criterion of the nodes called "fidelity Index" .Data packets are routed based on this fidelity index. Simulation results show our proposed scheme is simple yet robust under some scenarios.*

Keywords: *fidelity; delay; fidelity index; sequence number; hop destination; flooding attack; black hole attack; co-operative black hole attack;routing.*

I. INTRODUCTION

Mobile Ad-hoc Network (MANET) is a collection of wireless mobile hosts without fixed network [1] infrastructure and centralized administration (Figure-1). Communication in MANET [2] is done via multi-hop paths. MANET contains diverse resources; Nodes operate in shared wireless medium; [3] Network topology changes unpredictably and very dynamically; Radio link [4] fidelity is necessary; connection breaks are pretty frequent. Moreover, density of nodes, number of nodes and mobility of these hosts may vary in different applications. There is no stationary infrastructure. Each node in MANET [5] acts a router that forwards data packets to other nodes. Therefore selection of effective, suitable, adaptive and robust routing scheme is of utmost importance .Our proposed scheme that is fidelity based on demand (FBOD)routing is a secure routing scheme based on fidelity index. FBOD is a robust, effective, suitable, adaptive and cost effective scheme.

.

DOI : 10.5121/ijwmn.2011.3519

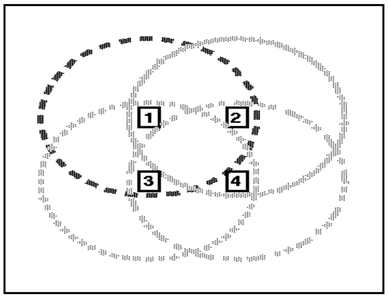


Figure 1: An ad-hoc mobile network with four nodes.

Section II describes some previous work in related field. Section III discusses fidelity. Delay matrix is discussed in section IV. Section V elaborates fidelity index. Section VI presents a description of the scheme Section VII presents FBOD algorithm Section VIII consists of simulation results. Section IX is a treatment on security aspects. Lastly we will discuss performance analysis of the scheme in section X and conclusions in section XI.

II. RELATED WORK

S. Matri [6] proposed to trace malicious nodes by using watchdog/pathrater. In watchdog when a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet by promiscuously listening to the next node's transmissions. If the watchdog finds the next node does not forward the packet during a predefined threshold time, the watchdog will accuse the next node as a malicious node to the source node; The proposal has two shortcomings: 1) to monitor the behavior of nodes two or more hops away, one node has to trust the information from other nodes, which introduces the vulnerability that good nodes may be bypassed by malicious accusation; 2) The *watchdog* cannot differentiate the misbehavior from the ambiguous collisions, receiver collisions, controlled transmission power, collusion, false misbehavior and partial dropping. In *pathrater* algorithm each node uses the *watchdog's* monitored results to rate its one-hop neighbors. Further the nodes exchange their ratings, so that the *pathrater* can rate the paths and choose a path with highest rating for routing. Shortcoming of this algorithm is that the idea of exchanging ratings genuinely opens door for blackmail attack.

SCAN [7] exploits two ideas to protect the mobile Ad Hoc networks [8]: 1) *local collaboration*: the neighboring nodes collectively monitor each other and sustain each other; and 2) *information cross-validation*: each node monitors its neighbors by cross-checking the overheard transmissions, and the monitoring results from different nodes are further cross validated. As a result, the security solution is self-organized, distributed, and fully localized. In SCAN once a malicious node is convicted by its neighbors, the network reacts by depriving its right to access the network by revoking its token. A powerful collusion among the attackers will break SCAN as it violates the assumption of the polynomial secret sharing scheme.

Gonzalez [9] 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 *vj* to forward and ii) the amount of packets forwarded by *vj* 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 *vj* to which *vi* has respectively transmitted packets to or received packets from. Node v_i 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. This algorithm does not require many nodes to overhear each others' received and transmitted packets, but instead 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.

III. FIDELITY

Fidelity is the most important concept of this routing scheme. Fidelity is an integer number that is associated with each node. This fidelity of a node denotes many things about the node itself and also deciphers other information regarding the topology of the entire network. It also helps to maintain security [10] to some extent.

To make it understandable in one sentence, "fidelity is a counter that is associated with a node, which is increased whenever it forwards a data packet successfully." Whenever a node comes in a network its fidelity is zero and whenever it goes permanently off from the network its value is again refreshed to zero. Otherwise whenever a node will forward any data packet it will always increase a counter value and that counter value is its fidelity. Note whenever a source node sends a data packet to a destination node, all the intermediate nodes helping to transmit its data packet will increase their counter but the source and the destination node do not increase their fidelity value.

Fidelity is a measure of these two factors:-

A. *How reliable a node is for forwarding a data packet*

Whenever we observe that the fidelity value of a particular node is greater that of another node then we can conclude that the one having the greater value is a more durable node than the other from who's its value is greater. It is quite logical because a node with greater value indicates that it is an experienced node in the network and it has transmitted packets most dutifully than other nodes.

B. *Network topology*

If we can find some nodes with higher fidelity in a region of the network, we conclude that the network activity is higher in that region. More precisely we can also infer that the node density is also higher in that region for it is impossible to have one node having very high fidelity [11] surrounded by nodes with low fidelity because a high fidelity [12] node must send packets to someone in its vicinity which will make that other node's fidelity value also high. Thus a high fidelity value accounts for high network activity as well as high density of nodes in its surroundings.

IV. DELAY MATRIX

Delay is one of the most important factors in this scheme. Delay signifies the time delay between two nodes. Delay matrix is basically a square matrix having dimension n*n, where n=no of nodes. Each (i, j)th entry in the matrix indicates delay between ith and jth node in a network. For example, we are taking a 2*2 matrix which is a delay matrix of a network having 2 nodes 0, 1. The delay between those two nodes is 5 units.

(i,j)	0	1
0	0	5
1	5	0

V. FIDELITY INDEX

Fidelity Index (FI) is basically a real no associated with each node. This index is the main factor in order to choose any neighbor to forward the message to destination. Each node sorts its neighbors in decreasing order of FI. If we get the time delay for a particular node as T and the fidelity value is F, then we can calculate the fidelity index (FI) = X * F + (1-X)/T, where X<1 & its value depends on the practical behavior of the network. The value of the X can be user given input. For our simulation we are considering x=.6 to give more stress on fidelity (security issue).

VI. DESCRIPTION OF THE SCHEME

The term “friends of a node” used in this paper, indicates actually the nodes that fall in the physical range of a particular node. When a node is a message to send, the node will check which nodes are in its neighborhood and what are their fidelity value and the delay between those nodes. Sender will broadcast a request along with an echo message. After getting reply they will make their friend list. More precisely the friend list consists of a table that contains two attributes. The first one is the address [13] of the nodes which are within its range and other is the fidelity index of that particular node. When each node is updated then they will sort that table according to the decreasing order of the fidelity index (FI). Before we enter into the detailed discussion of our scheme (FBOD) there are some concepts that need to be understood. These are as follows-

There will be a sequence counter in every node. If a message is generated in a node then it will be increased by one. This sequence no. will be forwarded as a part of the message. Every node will maintain a buffer where (source, sequence no.) will be stored for last n no. of received messages. After getting a message a node will verify the tuple [9] (source, sequence no) of that message with those tuples in its buffer [14]. If anyone of them matches with that message then that node will reject that message silently. It will prevent flooding attack.

The timeout period of every node through which message is traversed, will be gradually decreased by a critical factor [15] i.e. if timeout period of sender node is x then timeout period of receiver node will be (x-m), where m will be critical factor. This factor [16] helps us to control the max no of hops a message can traverse to reach destination.

Now the scheme is as follows-A node can do either of three activities - message generate, message forward, message receive. If it is not doing any of the three then it is idle. Now if a message is generated in a node and it needs to be sent then the node will remain busy until an acknowledgement is received for this message. It is to be noted that a busy node can accept & process an acknowledgement and can send a fail message.

Now if destination is directly reachable from generator node then it will send message to destination node and will wait for acknowledgement, and remain busy until acknowledgement is received. If the destination node is busy it will send a fail message to generator node. After getting fail message or if timeout period exceeds, generator node will keep on sending the message after a certain time periodically until acknowledgement is received.

If destination is not directly reachable then generator node will send message to the node in its range that has highest fidelity index. If generator node get a fail message from that node or if timeout period exceeds then it will send the message to the node having second highest fidelity index and it will continue like this. If the whole list is exhausted in this way then the process will again continue from the node having highest fidelity index. Only generator node will follow this process. Other nodes will send a fail message to its predecessor if the whole list is exhausted.

When a node receives a message, if it is busy then it will send a fail message to sender, otherwise it will check whether it itself is a destination or not. If it is destination, it will accept the message and send acknowledgement to sender otherwise this node will send message to the node in its range that have highest fidelity index and that process will continue. In that acknowledgement message the sequence no. will be same as received message but source will be substituted by destination.

VII. FBOD ALGORITHMS

Update friend list

STEP 1: Send Hello packet, Echo packet and a special broadcast request to the friends for knowing the identity, delay and fidelity of the friends
STEP 2: Receive replies from friends
STEP 3: Calculate FI=0.6*F+0.4/T, where F=fidelity value of a neighbour node & T=time delay to reach that neighbour. We have taken here the value of X as 0.6.
STEP 4: Update my friend list
STEP 5: Sort friend list in a decreasing order of FI

Generated data

```
STEP 1: Set my status=busy
STEP 2: If destination directly reachable from here
      o  Send packet to destination
      o  Wait for ACK
      o  If  ACK received consider success
      o  Else if timeout occurs or FAIL received, arrange for resending
     Else
      o  Send data packet to the friend having highest FI
      o  Wait for ACK
      o  If  ACK received consider success and go to  step 3
      o  Else if timeout occurs or FAIL received, arrange for resending to the friend with next highest FI
      o  Continue above three steps until ACK received
      o  If list is exhausted without getting an ACK then again start from the friend with the highest FI
          and try each node in friend list in the same manner as above.
      o  While trying to send if the list is exhausted thrice  abort
STEP 3: Set my status=free
```

Received data

```
STEP 1: If my status=busy send FAIL to sender
STEP 2: Else
      o  Make my status=busy
      o  Process received data
      o  Make my status=free
```

Process received data

```
STEP 1: If message destination=my address
      o  Accept data
      o  Generate ACK
      o  Send the ACK to the node from which it directly received the message
      o  If the received packet is found duplicate then discard the received packet.
STEP 2: Else
      o  Forward data packet
      o  Check if forward operation is successful
      o  If successful increase my fidelity value by 1 and send ACK to the node from which it directly
         received the message
      o  Else send FAIL to the node from which it directly received the message
```

Forward data packet

```
STEP 1: If message destination is directly reachable from here
      o  Send packet to destination
      o  Wait for ACK
      o  If  ACK received consider success
      o  Else if timeout occurs or FAIL received, arrange for resending to destination.
      o  If resending fails 3 times consider failure.
STEP 2: Else
      o   Send data packet to the friend having  highest FI
      o  Wait for ACK
      o  If  ACK received consider success
      o  Else if timeout occurs or FAIL received, arrange for resending to the friend with next highest FI
      o  Continue above three steps until ACK received
      o  If list is exhausted without getting an ACK then consider failure.
```

VIII. Simulation Result

We have simulated this scheme with JAVA. We need to know something to make out these simulations. These are

1. Small circle signifies node in the network.
2. Blue circle around node signifies range of that node.
3. Red color indicates that the node is free.
4. Black color indicates that the node is busy.
5. Yellow line between two nodes indicates sending of request & echo message to probe fidelity value & delay.
6. Pink line between two nodes indicates reply of probing with information.
7. Red line between two nodes indicates sending of message.
8. Green line between two nodes indicates sending of acknowledgement.
9. Blue line between two nodes indicates sending of fail message.
10. Any node inside the range of a node is its neighbor node.

Now we will describe one test case simulations.

This is a network having six nodes. Their corresponding fidelity values are written beside the nodes. Here we are trying to send a message from node 0 to node 5. Following figures depict the simulation results

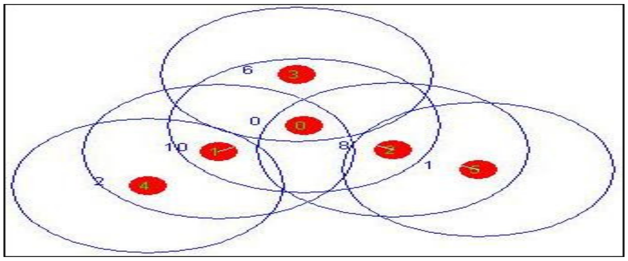


Figure 2: Design of network.

The result we get after network designing is given below-

Output of netdesign.jar:-

```
6<Number of nodes>
                         0      10      7      6      2      1
                        -1       0      0      0     -1     -1
                         0      -1     -1     -1      0     -1
                         0      -1     -1     -1     -1      0
                         0      -1     -1     -1     -1     -1
                        -1       0     -1     -1     -1     -1
                        -1      -1      0     -1     -1     -1
```

We got the adjacency list.txt as above and input **delay matrix**:-

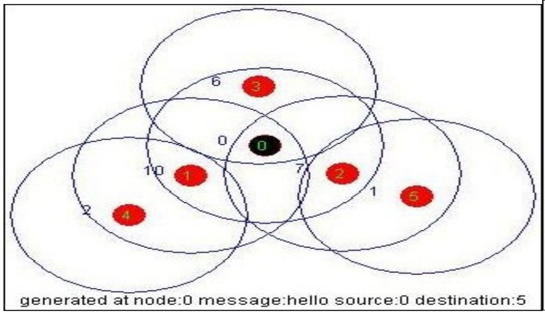
```
 0     3     2     1    -1    -1
 3     0    -1    -1     2    -1
 2    -1     0    -1    -1     4
 1    -1    -1     0    -1    -1
-1     2    -1    -1     0    -1
-1    -1     4    -1    -1     0
```

0<*time instance-0*>

0<source>   5<destination>   hello<*msg*>

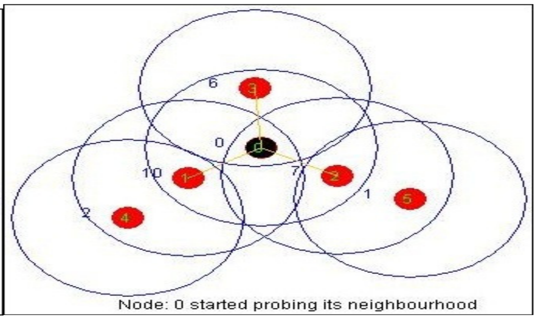
Then we run the simulation and see the results.

The steps of the visual simulation are given below-



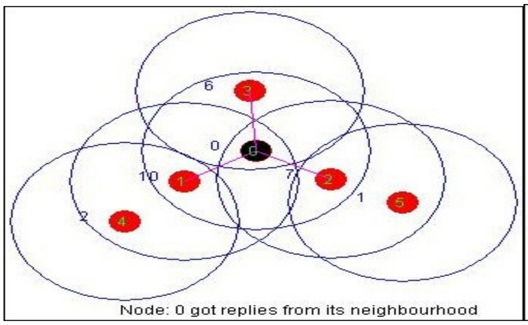
generated at node:0 message:hello source:0 destination:5

Figure 3: Message generated at node 0.



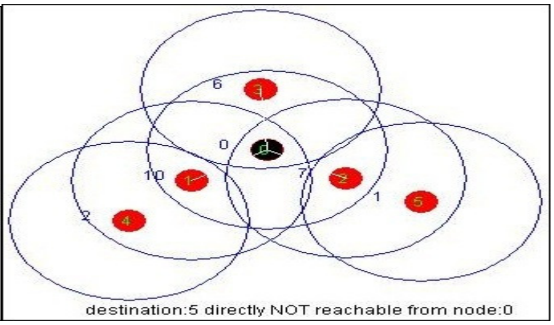
Node: 0 started probing its neighbourhood

Figure 4: Node 0 started probing.



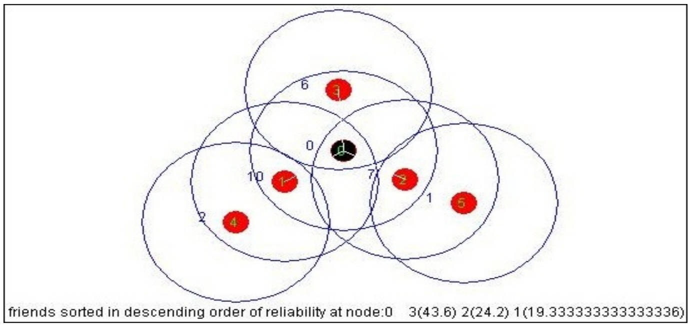
Node: 0 got replies from its neighbourhood

Figure 5: Node 0 got replies from neighbour nodes.



destination:5 directly NOT reachable from node:0

Figure 6: Destination is not directly reachable from source node.



friends sorted in descending order of reliability at node:0 3(43.6) 2(24.2) 1(19.333333333333336)

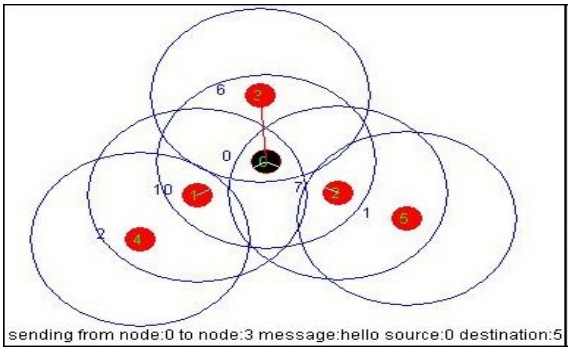
Figure 7: Friend nodes are sorted in descending order of Fidelity Index written beside neighbour nodes in brackets.

For node 1, FI=.6*10+.4*(100/3) =19.333333333

For node 2, FI=.6*7+.4*(100/2) =24.2

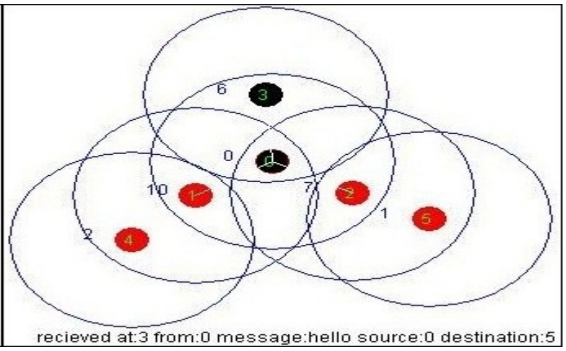
For node 3, FI=.6*6+.4*(100/1) =43.6

NOTE: We have taken (100/T) instead of (1/T) for the sake of calculation.



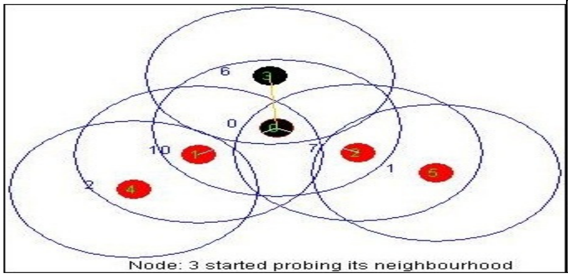
sending from node:0 to node:3 message:hello source:0 destination:5

Figure 8: Node 0 is sending message to node 3.



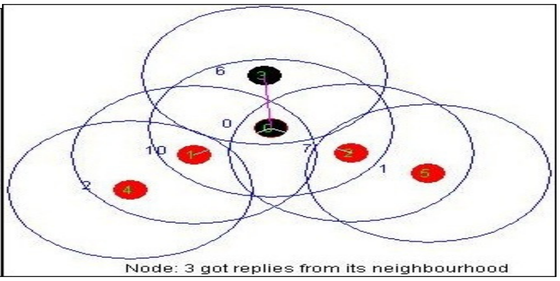
recieved at:3 from:0 message:hello source:0 destination:5

Figure 9: Message is received at node 3.



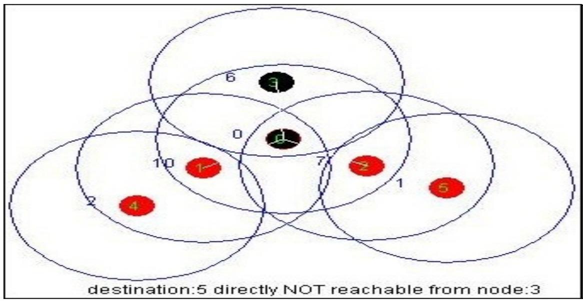
Node: 3 started probing its neighbourhood

Figure 10: Node 3 is starts probing neighbour nodes.



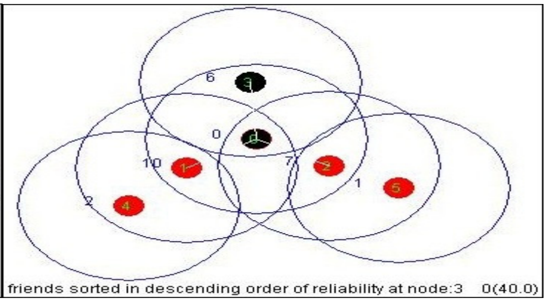
Node: 3 got replies from its neighbourhood

Figure 11: Node 3 receives reply from neighbours.



destination:5 directly NOT reachable from node:3

Figure 12: Destination is not reachable from node 3.

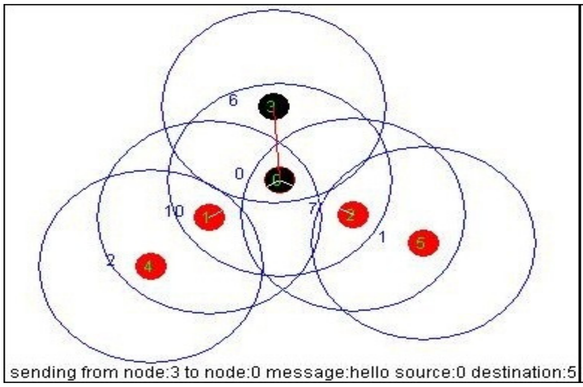


friends sorted in descending order of reliability at node:3 0(40.0)

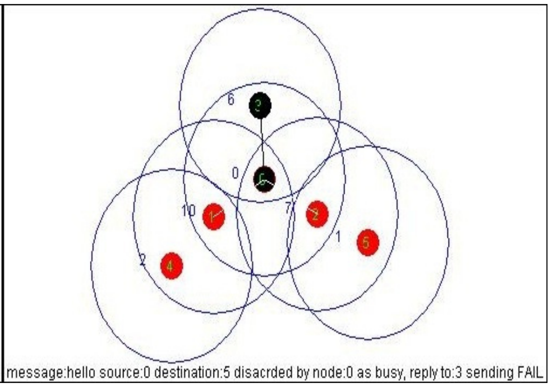
Figure 13: Friend nodes are sorted in descending order of Fidelity Index written beside neighbor nodes in bracket.

For node 0, FI=.6*0+.4*(100/1) =40.0

NOTE: We have taken (100/T) instead of (1/T) for the sake of calculation.



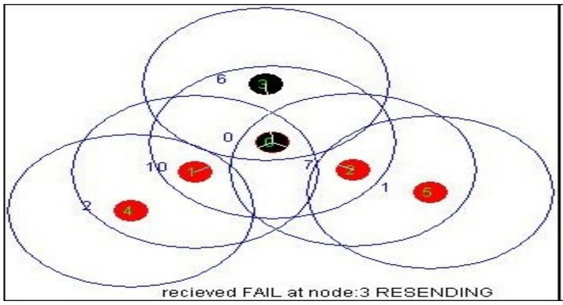
sending from node:3 to node:0 message:hello source:0 destination:5



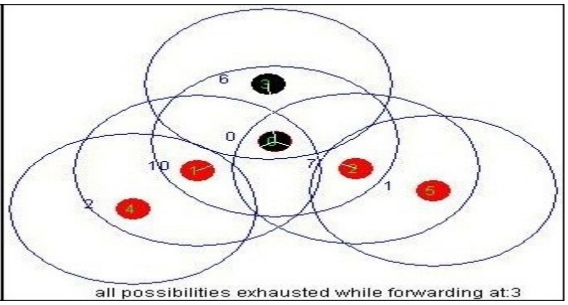
message:hello source:0 destination:5 disacrded by node:0 as busy, reply to:3 sending FAIL

Figure 14: Node 3 is sending message to node 0.

Figure 15: Node 0 discarded the massage.



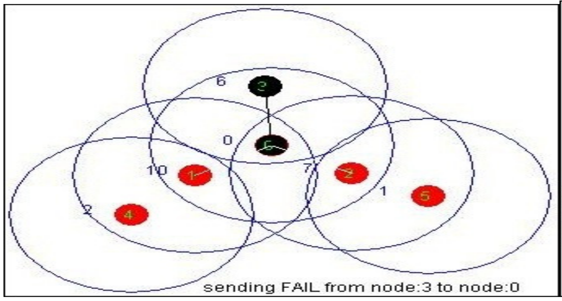
recieved FAIL at node:3 RESENDING



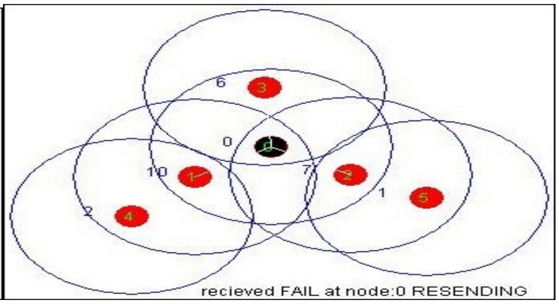
all possibilities exhausted while forwarding at:3

Figure 16: Node 3 receives a FAIL message.

Figure 17: Message cannot be forwarded from node 3.



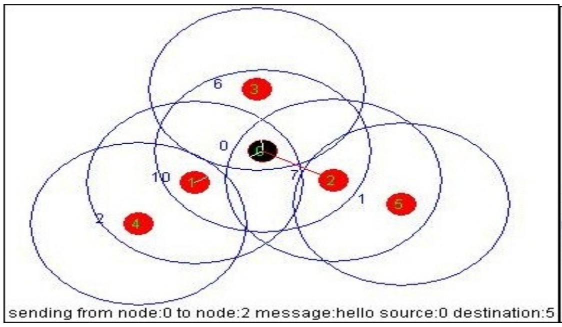
sending FAIL from node:3 to node:0



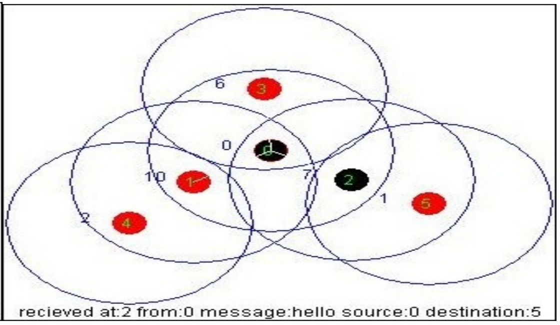
recieved FAIL at node:0 RESENDING

Figure 18: Node 3 is sending FAIL message to node 0.

Figure 19: Node 0 receives FAIL message.



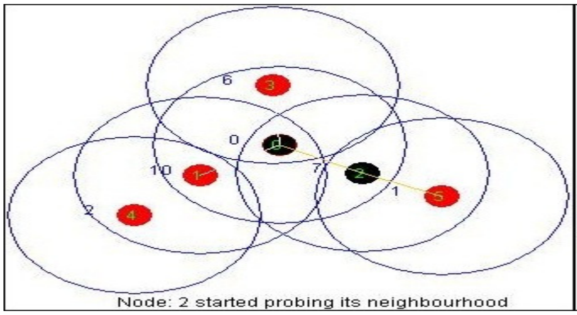
sending from node:0 to node:2 message:hello source:0 destination:5



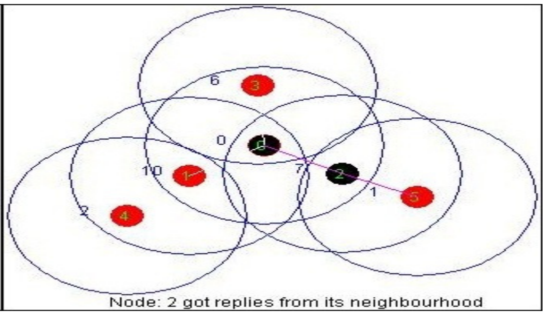
recieved at:2 from:0 message:hello source:0 destination:5

Figure 20: Node 0 forwarding message to node 2.

Figure 21: Node 2 receives message.



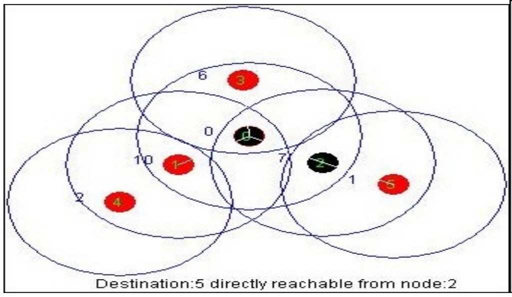
Node: 2 started probing its neighbourhood



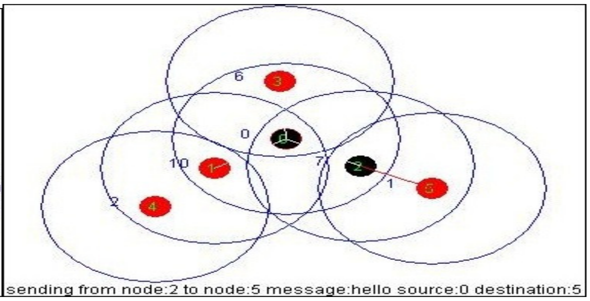
Node: 2 got replies from its neighbourhood

Figure 22: Node 2 started probing neighbours.

Figure 23: Node 2 receives reply from neighbours.



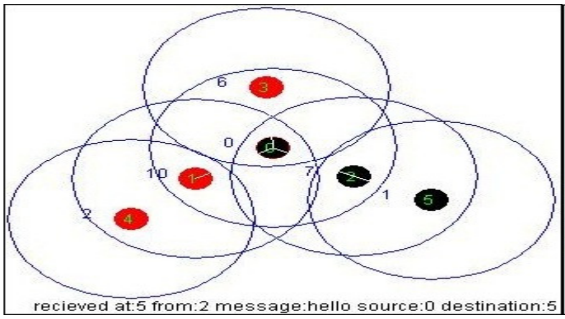
Destination:5 directly reachable from node:2



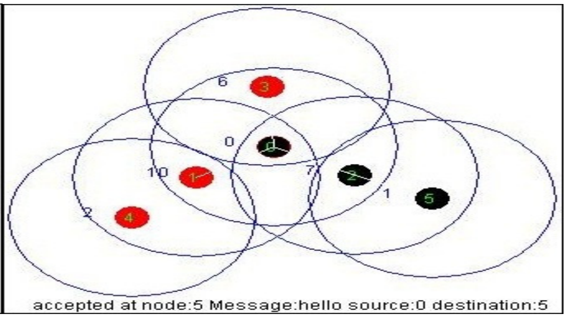
sending from node:2 to node:5 message:hello source:0 destination:5

Figure 24: Destination node 5 is reachable from node 2.

Figure 25: Node 2 sending message to node 5.



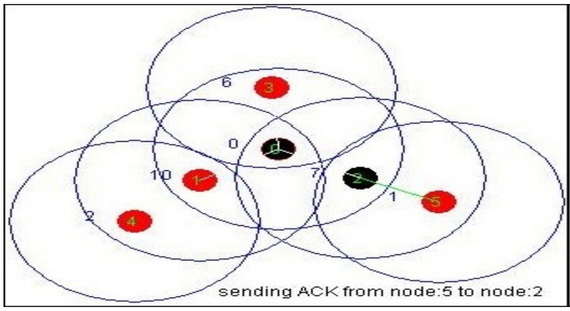
recieved at:5 from:2 message:hello source:0 destination:5



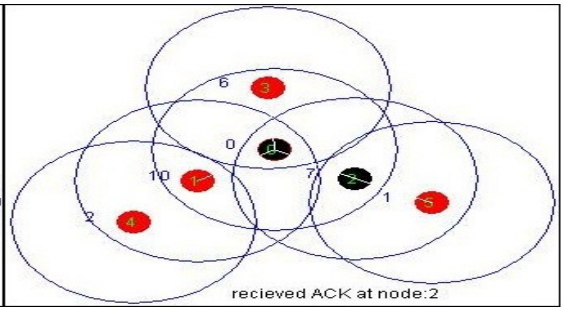
accepted at node:5 Message:hello source:0 destination:5

Figure 26: Node 5 receives message from node 2.

Figure 27: Node 5 accepts the message and generates an ACK.



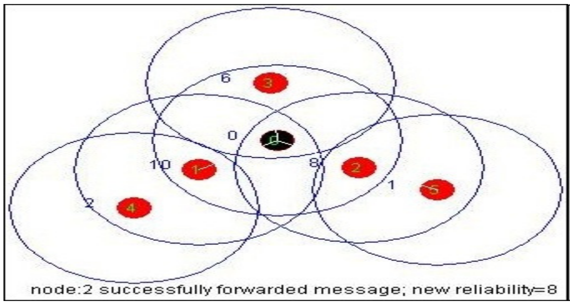
sending ACK from node:5 to node:2



recieved ACK at node:2

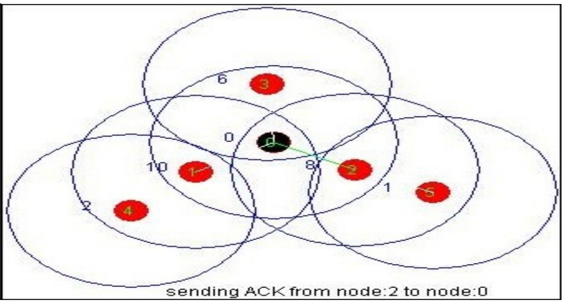
Figure 28: Node 5 sending ACK to node 2.

Figure 29: Node 2 receives ACK.



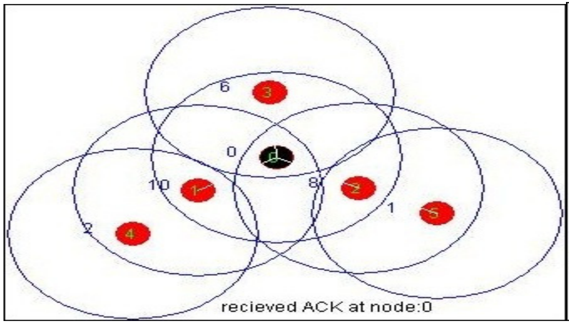
node:2 successfully forwarded message; new reliability=8

Figure 30: Fidelity of node 2 increases.



sending ACK from node:2 to node:0

Figure 31: Node 2 forwarding ACK to node 0.



recieved ACK at node:0

Figure 32: Node 0 received ACK from node 2.

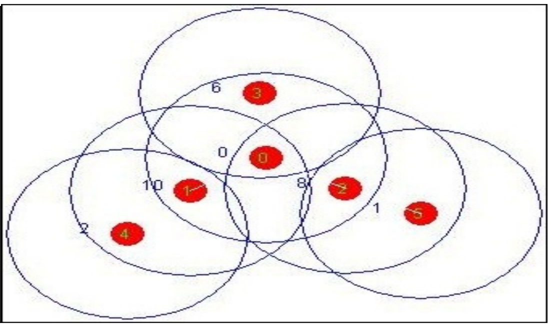


Figure 33: Messages transferred successfully.

Message transfer is completed.

IX. SECURITY ASPECTS

A. Flooding Attack

Flooding attack [17] is an attack that attempts to cause a failure in a computer system or other data processing entity by providing more input than the entity can process properly. Flooding attack occurs when a network or service becomes so weighed down with duplicate packets that it can no longer process genuine connection requests.

In our scheme, we have two fields in the data packet *a)* Sequence number. *b)* Source node number. These two fields together uniquely identifies [18] a message packet. Each data packet generated by a single node in this scheme is guaranteed to have different sequence number. Here if a node encounters a data packet that was recently 'seen' by it, it discards the packet. This prevents propagation of duplicate packet through the network, mitigating flooding attack

B. Black Hole Attack

Black holes [19] refer to places or nodes in the network where incoming traffic is silently discarded (or "dropped"), without informing the source that the data did not reach its intended recipient. The black hole problem is one of the security [20] attacks that occur in MANET.

There are two possible solutions. The first is to exploit the packet sequence number included in any packet header. Computer simulation shows that compared to the original *Ad Hoc on-*

demand distance vector (AODV) routing scheme [21] [22] [23]; this solution can verify the route to the destination depending on the pause times at a minimum cost of the delay in the networks. The second is to find more than one route to the destination. Here we are going to discuss about how our scheme implements the second solution vividly.

Let us assume that there are four nodes A, B, C and D. A wants to send a data packet to node C. Node A cannot communicate with destination node directly. Data packet can be sent from A to C in two ways –

i) A→ D→ C

ii) A→ B→ C.

A will pick first path to send the data packet as fidelity of D is greater than B. A will wait for acknowledgement after sending the data packet to node D. But node D is a black hole node. Data packet can't be reached to destination node by this path.

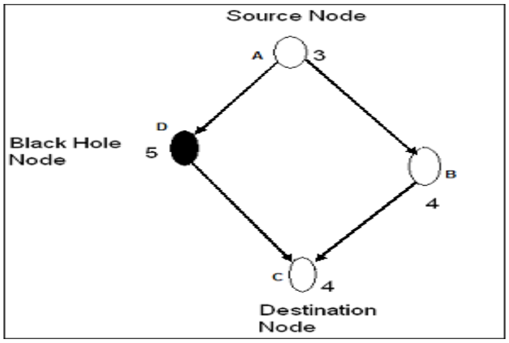


Figure no. 34.1

Node A will pick the second path after timeout period shown in *figure no. 34.1*. Node A will send the data packet to node B and node B will relay the data packet to the destination node C. Node C will send an acknowledgement to node B. Node B will forward the acknowledgement to the source node A. After the successful transmission of data packet, the fidelity of node will be increased and updated. Thus, the fidelity of node B will increase to 5 from 4 as shown in *figure no. 34.2.*

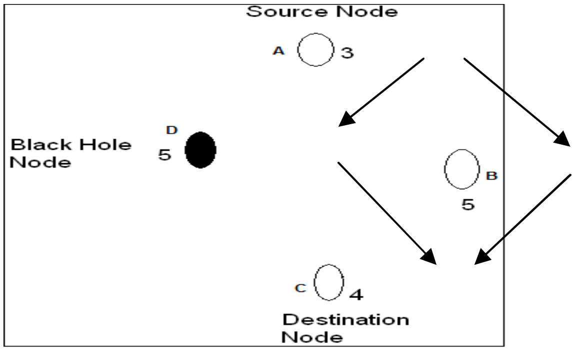


Figure no. 34.2

After every successful transmission of data packets the fidelity and the fidelity of friend nodes will increase. This way fidelity of node B will become 6 shown in *figure no. 34.3* after third transaction.

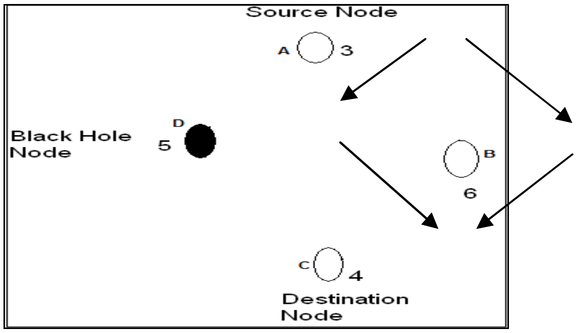


Figure no. 34.3

Afterwards node A will always pick node B to send data packet to node C as the fidelity of B is higher than node D. This way the black hole node will be terminated as shown in *figure no. 34.4.*

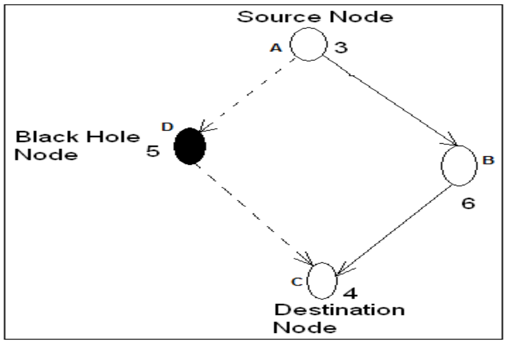


Figure no. 34.4

C. *Co-operative Black Hole Attack*

In co-operative black hole attack [19], there remains a group of malicious nodes. One of them takes the data packet and keeps on forwarding it among them so that the TTL of that data packet finishes off and the data packet is automatically dropped. We understand that the aim of these co-operative black hole attackers is to form a closed loop among themselves and keep on forwarding the message within this loop..

In our scheme a node will remain busy after sending a packet until it gets the acknowledgement for this message. So whenever any node try to send the message to a node already visited, it will simply reject the message and send a fail message to the sender as it is busy. As this scheme do not allow messages to transfer through the same node more than once there is no possibility for these malicious nodes to form the loop and hence co-operative black hole attack is mitigated.

D. Gray Hole Attack

A gray hole attack [17] is a variation of black hole attack, where an adversary first behave as an honest node during the route discovery process, and then silently drops some or all of the data packets sent to it for further forwarding even when no congestion occurs. Detection [24] of gray hole attack is harder because nodes can drop packets partially not only due to its malicious nature but also due to overload, congestion or selfish nature. A selfish node is unwilling to spend its battery life, CPU cycles or available network bandwidth to forward packets not of direct interest to it, even though it expects others to forward packets on its behalf.

If the probability of packet dropping of the malicious nodes is more than ½ then it is possible to mitigate gray hole attack by this scheme, otherwise it may or may not be possible. Whenever a node drops most of the packets sent to it then it behaves almost like a black hole attack and in that case this scheme can mitigate gray hole attack.

E. Black Mail Attack

Blackmail attack [17] causes false identification of a good node as malicious node. In Ad Hoc [17] [19] wireless networks [25], nodes usually keep information of perceived malicious nodes in a blacklist. An attacker may blackmail a good node and tell other nodes in the network to add that node to their blacklists as well, thus avoiding the victim node in future routes.

In our scheme there is no scope of broadcasting any details of a particular node to the nodes of the network by any other node. Here one node can only broadcast its own details to its neighbor nodes. So no attacker node can convince other nodes that a particular node is malicious. Thus there is no chance of black mail attack.

F. Rushing Attack

An offensive that can be carried out against on-demand routing schemes [23] is the rushing attack. Typically, on-demand routing schemes [26] state that nodes must forward only the first received Route Request from each route discovery; all further received Route requests are ignored. This is done in order to reduce cluttering. The attack consists, for the adversary, in quickly forwarding its Route Request messages when a route discovery is initiated. If the Route Requests that first reach the target’s neighbors are those of the attacker, then any discovered route includes the attacker.

In our scheme there is no concept of route discovery which shows the path from sender to receiver. Here each node only has the information of its neighbor nodes and based on this information (fidelity value) sender selects a neighbor and sends the packet to that selected node. So rushing attack cannot be happened in the network following this scheme.

G. Worm Hole Attack

When the source node broadcast the RREQ packet, a malicious node which is at one part of the network receives the RREQ packet. It tunnels that packet to a second colluding party which is at a distant location near the destination, it then rebroadcasts the RREQ. The neighbors of the second colluding party receive the RREQ and drop any further legitimate requests that may arrive later on legitimate multi hop paths [27]. The result is that the routes between the source and the destination go through the two colluding nodes that will be said to have formed a wormhole [17] between them. This prevents nodes from discovering legitimate paths that are more than two hops away.

In our scheme there is no concept of route discovery which shows the path from sender to receiver. Here each node only has the information of its neighbor nodes and based on this information(fidelity value) sender selects a neighbor and sends the packet to that selected node So here is no need of RREQ packet broadcasted from the source node. This type of worm hole attack can not hamper the network following this scheme.

X. Simulation Analysis And Performance Metrics

In order to evaluate the performance of Ad Hoc network routing schemes, the following parameters were considered:

A. *Packet Delivery Fraction(PDF)*

PDF is defined as the ratio between no. of packets originated by application layer [28] in the source node to the no of packets received by the destination node. It will describe the loss rate that will be seen by the transport schemes, which in turn affect the maximum throughput that the network supports. In terms of packet delivery fraction, our scheme FBOD performs well. As the no of nodes getting increased the number of packets generated is higher so it may not transfer some of the packets, but the no of these packets are very small. When the no. of nodes is small then in ideal case PDF value is 1. But in case of DSR [29] the PDF is very fluctuating it is lesser in some of the points with respect to the other schemes but it is very higher in some of the points which are not tolerable. DSDV [30] is better in more no. of nodes but AODV [31] [32] is better in smaller no. of nodes region.

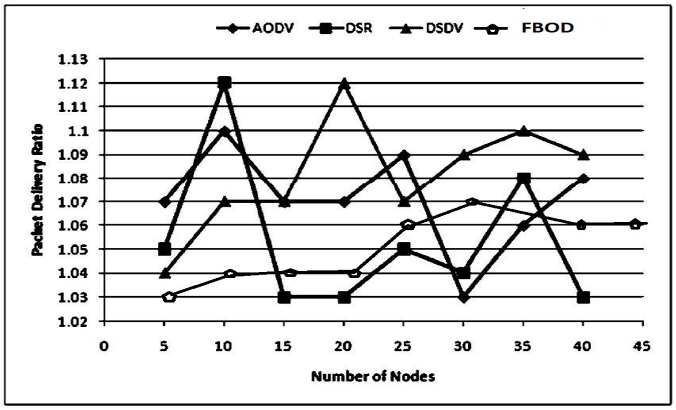


Figure 35.1: Packet Delivery Ratio for AODV, DSR, DSDV, FBOD

B. End to End Delay

The delay is affected by high rate of CBR Packets as well as the buffers become full much quicker, so packets have to stay in the buffer for a longer period of time before they are sent. This can be seen in DSR [2] when it reaches around 2300 packets in 0- mobility. For average end to end delay, the performance of DSR [33] decreases and varies with the number of nodes. In our scheme that is in FBOD the delay is getting increased with the increased no of nodes as the congestion is getting increased. But the rate of this increment is lesser as compared to other because of packets have to stay lesser time in buffer and lass computational overhead. The performance of DSDV [33] id degrading due to increase in the number of nodes the load of exchange of routing tables becomes high and the frequency of exchange also increased. Due to the mobility of nodes the performance of AODV [26] decreases and remains constant as the no of nodes increases.

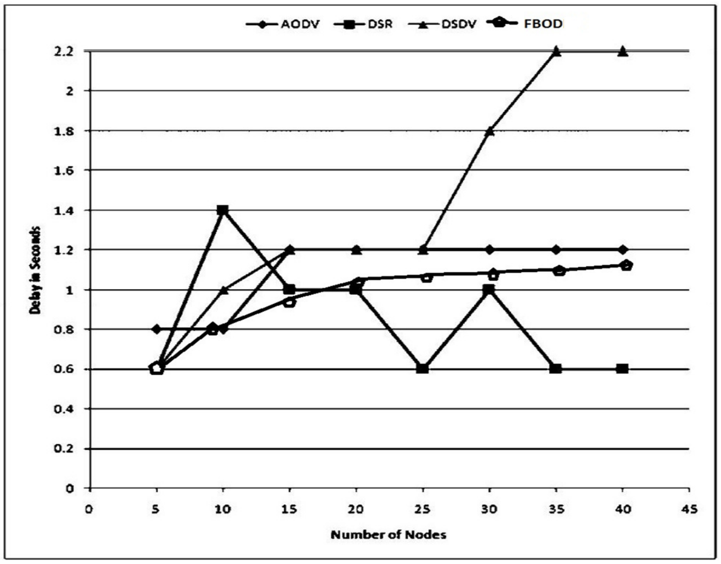


Figure 35.2: Average End to End Delay Ratio for AODV, DSR, DSDV, FBOD

C. Number of Packets Dropped

The number of data packets that are not successfully sent to the destination is the no of packets being dropped. In terms of dropped packets AODV's [2] performance is the worst. The performance decreases with the increase in the number of packets. As the no of nodes increases the no. of packets dropped increases which means that the no of packets not successfully reaches to the destination increases. DSDV [2] [33] performs consistently well with increase in the no. of nodes. DSR [29] [33] performs well when no of nodes is less but fails slightly when no of nodes is increased. In our scheme also in ideal case there is no drop of packets with the increase in no of nodes. It performs consistently well.

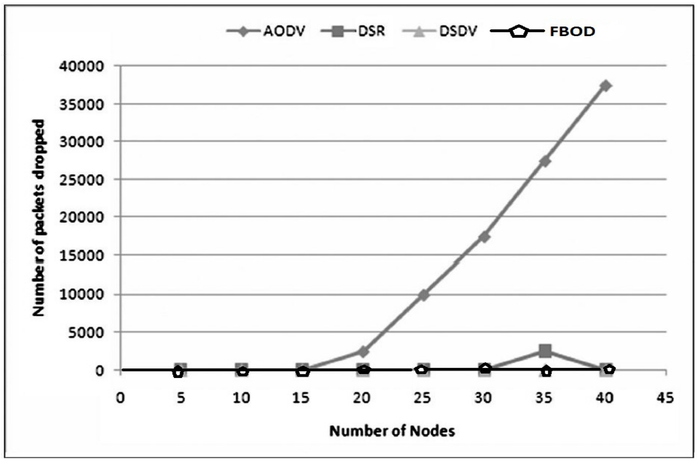


Figure 35.3: Dropped Packets for AODV, DSR, DSDV, FBOD

XI. CONCLUSION

This is a very light weight scheme with minimum computational overheads. In DSDV, we need to maintain a routing table. AODV has a lot of overhead while discovering routes, which clogs the network for sending data packets to desired destination. Not only does no such complicacy exist in our scheme, but it also has some of their benefits. Like AODV it is an on-demand routing scheme and the physical hardware support needed to implement it is substantially low which increases its scalability. This scheme also has added features so as to nullify some of the security threats which cause faults in the MANET networks.

REFERENCES

[1] Hongmei Deng, Wei Li, and Dharma P. Agrawal, “Routing Security in Wireless Ad Hoc Networks”, IEEE Communications Magazine October 2002.

[2] V. Ramesh, Dr. P. Subbaiah, N. Koteswar Rao, M. Janardhana Raju, “Performance Comparison and Analysis of DSDV and AODV for MANET”, V. Ramesh et al. / (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 02, 2010, 183-188.

[3] Huaizhi Li Zhenliu Chen Xiangyang Qin, “Secure Routing in Wired Networks and Wireless Ad Hoc Networks” IEEE, 2004.

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

[5] 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.

[6] S. Matri, T. J. Giuli, K. Lai and M. Baker , Mittigating Routing misbehaviour in Mobile Ad Hoc Networks. *Proceedings of the 6th annual internaional conference on Mobile Computing and Networking (MOBICOM), Boston, Massachusetts, United States, 2000, 255-265.*

[7] H. Yang, J. Shu, X. Meng, and S. Lu, “SCAN: Self-organized network-layer security in mobile ad hoc networks,” IEEE Journal on Selected Areas in Communications, vol. 24, issue 2, pp. 261-273, February 2006.

[8] F. Anjum, Anup K. Ghosh, Nada Golmie, Paul Kolodzy, Radha Poovendran, Rajeev Shorey, D. Lee, J-Sac, “Security in Wireless Ad hoc Networks”, IEEE journal on selected areas in communications, vol. 24, no. 2, February 2006.

[9] Oscar F. Gonzalez, Michael Howarth, and George Pavlou, Detection of Packet Forwarding Misbehavior in Mobile Ad-Hoc Networks. Center for Communications Systems Research, University of Surrey, Guildford, UK. Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on May 21, 2007.

[10] Nik os Komninos, Dimitris Vergados, Christos Douligeris, "Layered security design for mobile ad hoc networks" journal computers & security 25, 2006 , pp. 121 – 130.

[11] Haiyun Luo, Petros Zerfos, Jiejun Kong, Songwu Lu, Lixia Zhang, "Self-securing Ad Hoc Wireless Networks", 7th IEEE Symp. on Comp. and Communications (ISCC), Taormina, 2002.

[12] H. A. Wen, C. L. Lin, and T. Hwang, "Provably Secure Authenticated Key Exchange Protocols for Low Power Computing Clients," Computers and Security, vol. 25, pp. 106-113, 2006.

[13] Anand Patwardhan, Jim Parker, Michaela Iorga. Anupam Joshi, "Tom Karygiannis, Secure Routing and Intrusion Detection in Ad Hoc Networks" 3rd International Conference on Perv asive Computing and Communications (PerCom 2005), Kauai Island, Hawaii.

[14] Angel R. Otero, Carlos E. Otero and Abrar Qureshi, "A Multi-Criteria Evaluation of Information Security Controls Using Boolean Features", International Journal of etwprk Security & its application (IJNSA), Vol. 2, No. 4, October 2010.

[15] Bing Wua, Jie Wua, Eduardo B. Fernandeza, Mohammad Ilyasa, Spyros Magliveras, "Secure and efficient key management in mobile ad hoc networks" Journal of Network and Computer Applications 30 (2007) 937–954.

[16] J. Nam, S. Cho, S. Kim, and D. Won, "Simple and Efficient Group Key Agreement Based on Factoring" Proc. Int'l Conf. Computational Science and Its Applications (ICCSA '04), pp. 645-654, 2004.

[17] 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.

[18] 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.

[19] "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.

[20] P. Papadimitratos and Z.J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks" Proc . IEEE Workshop on Security and Assurance in Ad Hoc Networks, IEEE Press, 2003, pp. 27–31.

[21] [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.

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

[23] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 17 February 2003.

[24] J. Parker, J. L. Undercoffer, J. Pinkston, and A. Joshi., "On Intrusion Detection in Mobile Ad Hoc Network s". In 23rd IEEE International Performance Computing and Communications Conference Workshop on Information Ass urance. IEEE, April 2004.

[25] Huaizhi Li, Mukesh Singha, "Trust Management in Distributed Systems" IEEE Computer Society February 2007.

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

[27] S. Holeman, G. Manimaran, J. Dav, and A. Chakrabarti, "Differentially secure multicasting and its implementation methods", Computers & Security, Vol 21, No. 8, pp 736-749, 2002.

[28] 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.

[29] Anuj K. Gupta, Dr. Harsh Sadawarti, Dr. Anil K. Verma, "Performance Analysius of AODV, DSR & TORA Routing Protocols", IACSIT International Journal of Engineering & Technology, Vol. 2, No. 2, April 2010, ISSN: 1793 - 8236.

[30] R. Balakrishan, S. Jayabalan, Dr. U. Rajeswar Rao, Dr. T. K. Basak. Dr. V. Cyrilraj, "Performance Issues on AODV and DSDV for MNAETS", Jounnal Theoritical and Applied Information Technoilogy.

[31] Sapna S. Kaushik & P. R. Deshmukh. "Comparison of effectiveness of AODV, DSDV and DSR routing protocols is mobile Ad hoc networks", International Journal of Information Technology and Knowledge Management, July – December 2009, volume 2, No. 2, pp. 499-502.

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

[33] Nor Surayati Mohamad Usop, Azizol Abdullah, Ahmad Faisal Amri Abidin, "Performance Evaluation of AODV, DSDV & DSR Routing Protocol in Grid Environment", IJCSNS International Journal of Computer Science and Netwprk Security, Vol. 9, No. 7, July 2009.