# BRTCM: Black Hole Removal Using Threshold and Co-operative Method

Aditya Bakshi, Atul Mishra , Heena Batra

**Abstract**

**A Mobile Ad-hoc Network is an infrastructure-less network and it is prone to various attacks at due to its unique characteristics such as open peer-to-peer network architecture, shared wireless medium, stringent resource constraints, and highly dynamic network topology. Secure communication between the mobile nodes is our prime concern. MANET is vulnerable to various attacks at different layers. One of the attacks is Black Hole attack at network layer. Black Hole attack is a kind of Denial of Service attack in which an adversary node advertise itself as it is having the shortest path from source to destination in order to get selected as a part of the path through which data is to be sent. When data is sent through this node, it just drops the packet or delays it to prevent the communication. In this paper, new algorithm has been proposed to remove Black Hole nodes to provide secure communication between mobile nodes. The proposed algorithm first of all finds out the probable list of malicious node using a threshold method and then it confirms the malicious behavior of node using co-operative method. The proposed algorithm is very efficient to find out the Black Hole node than the existing solutions proposed earlier.**

**Keywords:** MANET, Black Hole attack, AODV, Routing Protocols, Wireless Networks, Security.

IJSER

## 1. INTRODUCTION

### A. MANET

MANET is infrastructure-less wireless network where the mobile nodes communicate with each other without any fixed infrastructure and using radio frequencies in air to transmit and receive data instead of using some physical cables. MANET operates without centralized administration. So the functioning of Ad-hoc networks is dependent on the co-operation between nodes for connectivity and services. Nodes share the responsibility of managing the network and help each other in conveying information about the topology of the network. Hence in addition to acting as hosts, each mobile node behaves as router relay messages for other mobile nodes.

### B. Routing Protocols in MANET

Routing protocols can be divided into proactive or table-driven, reactive or on-demand and hybrid protocols [1]. Proactive protocols are typically table-driven. Each node uses routing tables to store the location information of other nodes in the network and periodically exchange these tables to maintain the fresh and consistent overview of the network. This information is further used to transfer data among various nodes of the network. Examples of this type include Destination Sequence Distance Vector (DSDV) and Cluster-head Gateway Switch Routing (CGSR). Reactive or on-demand protocols, in contrary, do not periodically update the routing information, it initiates a route discovery process which goes from one node to the other until it reaches to the destination or an intermediate node has a route to the destination. The source node then uses this route for data transmission to the destination node. Example of this type includes Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector (AODV) and Temporary Ordered Routing Algorithm (TORA). A hybrid protocol combines the features of both the approaches reactive and proactive. Example of hybrid protocol is Zone Routing Protocol (ZRP).

### C. AODV Routing Protocol

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol is an adaptation of the DSDV protocol for dynamic link conditions [1][3][4]. In AODV, nodes maintain routing table, which contains information about the route to a particular destination. Unlike proactive protocols, these tables are not exchanged periodically. A route is discovered whenever a packet is to be sent by a node. It first checks with its routing table to determine whether information about route to the destination is already available. If so, it uses that route to send the packets to the destination but if a route is not available or the previously entered route is inactivated, then the node initiates a route discovery process. A RREQ (Route REQuest) packet is broadcasted by the node which is received by its neighbors. Every neighbor that receives the RREQ packet first checks if it is the destination for that packet and if so, it prepares and sends back an RREP (Route Reply) packet. If it is not the destination, then it checks if it has got a route to the destination in its routing table. If not, it broadcast the RREQ packet to its neighbors.

If its routing table contains an entry to the destination, then it does the comparison of the 'Destination Sequence' number in its routing table to that present in the RREQ packet. If the destination sequence number present in the routing table is equal to or lessen than the one contained in the RREQ packet, then the node broadcast the request packet further to its neighbors. If the 'Destination Sequence' number in the routing table is higher than the number in the RREQ packet, it means that the route is a 'fresh route' and this route can be used to send the data packets. This intermediate node then prepares and sends a RREP packet to the node through which the RREQ packet is received. The RREP packet reaches to the source through the reverse route. The source node then updates its routing table and selects this route to send data packets. AODV also uses one more type of messages i.e RERR (Route ERRor ) messages. During the process, if link failure is identified by any node it sends a RERR packet to all other nodes that uses this link for their communication to other nodes. This is shown in Fig 1.

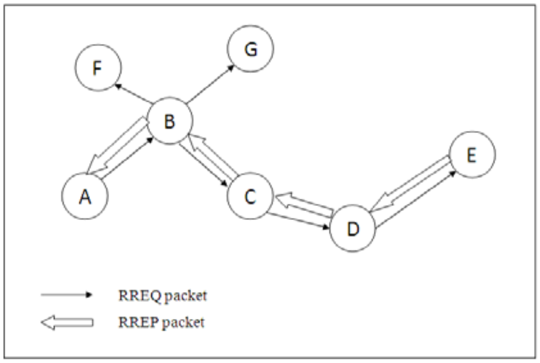

**Fig 1: Propagation of RREQ and RREP packets.**

### D. Black Hole Attack

MANETs are prone to various types of attacks on different layers like physical, MAC and network layer. These are the most important layers while routing the data packet. A Black Hole attack is a network layer attack and it's a kind of denial of service attack. In Black Hole attack, the malicious node (or Black hole node) waits for the source node to initiate the route discovery process and broadcast the RREQ packet. As the malicious node receives the RREQ packet, it immediately generates and sends RREP packet back to source with a very high Dest_seq_no as that of RREQ packet. Dest_seq_no is set to a high number to falsely claim a fresh route to the destination. Now when this RREP packet is received by the source node, it compare the Dest_seq_no of RREP packet with the Dest_seq_no of RREQ packet which is sent by itself. Due to higher Dest_seq_no, the source node assumes that the route to the destination through malicious node is fresh and it decides to send data through this path. As the data packets are sent via this path, malicious node absorbs all the packets and thus behaves like a Black Hole. For Example:

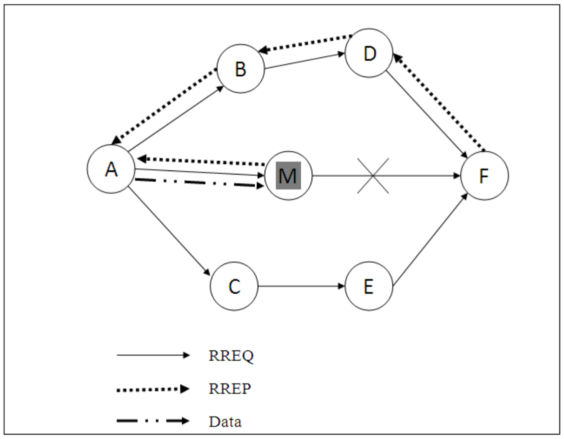

**Fig. 2: Black Hole Attack in MANET**

in Fig. 2 , source node A wants to communicate with node F. First of all, node A broadcasts RREQ to its neighbors to discover the path to destination (node F). All the neighbors i.e node B, C and M receives the RREQ packet and node M without doing any processing generates and send RREP packet with a very high value set as Dest_seq_no. Node A then receive RREP packet from node M ahead of nodes B and C. Node A selects the route via node M to send the data to destination node and start sending packets through this path. Node M then swallows all the packets and thus prevents the communication from source to destination.

In this paper a new algorithm, BRTCM, has been proposed to remove Black Hole nodes to provide secure communication between mobile nodes. The proposed algorithm first of all finds out the probable list of malicious node using a threshold method and then it confirms the malicious behavior of node using co-operative method. Rest of the paper is arranged as follows. Section 3, discusses the core algorithms to find out the list of probable malicious nodes based on certain threshold value and subsequently cooperative method to confirm it. Section 4, discussed the experimental results obtained on the basis of simulated implementation carried out on .Net platform. Section 5, lists some possible future work items.

## 2. BRTCM: Black Hole attack Removal using Threshold and Co-operative Method.

In BRTCM, first of all we will find out the probable list of malicious nodes among the neighbors using threshold method and then after that we will confirm the malicious behavior of the nodes using co-operative method. After confirmation of malicious behavior, nodes are separated from the network.

### 3.1. Finding the probable list of Malicious Node

In normal AODV, route Discovery process is initiated by broadcasting a Route Request (RREQ) packet to its neighbors. Each neighbor node either responds the RREQ by sending a Route Reply (RREP) back to the source node or rebroadcasts the RREQ to its own neighbors. The node that receives the RREP packet first checks the value of sequence number in its routing table. The RREP packet is accepted if it has RREP_seq_no higher than the one in routing table. Our solution does an additional check to find out whether the RREP_seq_no is higher than the calculated threshold value. The threshold value is dynamically updated in every time interval. The nodes with higher RREP_seq_no are added to the black list.

**Calculation of Threshold Value**

- The Threshold Value is dynamically updated using the data collected in the time interval.
- The time interval to update the threshold value is as soon as a node receives a RREP packet.

#### 3.1.1. Rule used to calculate Threshold value

**Notations:**

**Thr :** Threshold Value .
**Dest_seq_no( RT ) :** Destination sequence number in Routing Table of that node.
**Dest_seq_no( RREP_pkt) :** Destination sequence number in RREP packet received from neighbor node.

**Avg: Average.**

IJSER

1) **Thr = [ Dest_seq_no( RREP_pkt(1) ) - Dest_seq_no( RT ) ] + [ Dest_seq_no( RREP_pkt(2) ) - Dest_seq_no( RT ) ]……… +Dest_seq_no( RREP_pkt(n) ) - Dest_seq_no( RT ) ]/n**

   **Where n is total number of neighbor nodes from which node has received RREP pkts.**

   **OR**

2) **Thr = Avg (differences(Dest_seq_no( RREP_pkt(i),Dest_seq_no( RT))**

   **Where i varies from 1 to n.**

**Fig. 3: Rule used to find the Threshold value**

After comparison, if the value of RREP_seq_no is found to be higher than the threshold value, the node is suspected to be malicious and this node will be added to the black list. Black list is the list of suspicious nodes which may act as black hole in network. Fig 3 provides a set of rules to compute threshold value for the deduction of probable malicious node.

### 3.2.1 Proposed Algorithm to find out the elements of Black List

**Algorithm 1**

**Notations::**

**RREQ_pkt : Route Request packet, NL : Neighbor List, t0 : Stores the initial value of time, WT : Waiting Time, Dest_seq_no : Destination sequence number , RREP_pkt : Route Reply packet , RREP_table : Route Reply table,**

**Thr : Threshold Value**

**Black List( Node_id )**

```
Begin
For ( Node_id )
  {
     Send RREQ_pkt to every  x Є NL
     t0 = get ( current time value)
     Set timer ( t0+WT )
     Receive RREP_pkt ( packet P )
     till timer expires
     Store P.Dest_seq_no and  corresponding node_id in RREP_table
     After timer expires
     While ( RREP_table is not empty)
          {
            Set Dest_seq_no from table
            If ( Dest_seq_no >= Thr )
              {
                 Put corresponding Node_id  in Black List
              }
          }
  }
End
```

IJSER

### 3.3.1.Explanation of Algorithm 1: (Black List ( Node_id)):

1) This algorithm takes Node_id as an argument whose neighbor nodes are to be examined for their behavior.
2) That node (whose Node_id is passed as an argument) will then send RREQ_pkt to all its neighbors in Neighbor List.
3) Current time value is noted and a timer is set to time value equal to the addition of current time and waiting time value.
4) Node will collect all RREP_pkt till the timer expires.
5) Alongwith reception of RREP_pkt, node will maintain RREP_table which maintains the sender address of RREP_pkt and Dest_seq_no in RREP_pkt.
6) When the timer get expire, we take out the nodes in RREP_table one by one and compare its Dest_seq_no with the Threshold Value.
7) If the value of Dest_seq_no is greater than Threshold value then put the corresponding Node_id in Black List.
8) Otherwise keep the entries for that node in NL.

### 3.2. Co-operative detection to confirm the malicious behavior of nodes in Black List

Once the list of possible black hole nodes is maintained with the help of Threshold Value, the cooperative detection procedure is activated. The cooperative detection procedure is initiated by the initial detection node, which proceeds by first broadcasting and notifying all the one-hop neighbors of the possible suspicious node to cooperatively participate in the decision process confirming that the node in question is indeed a malicious one.

### 3.1.1. Proposed Algorithm for Co-operative detection:

**Algorithm 2**

**Notations:**

**IN : Initial detection node (source node) , SN : Suspicious Node , DN : Destination Node ,IN1H : IN's 1 Hop Neighbor node list , SN1H : SN's 1 Hop Neighbor node list, VT : Voter Table, WT : Waiting Time.**

```
Begin
Find out the common nodes of IN1H and SN1H, i.e IN1H ^ SN1H and store it in list called INSN1H i.e the list of nodes which are at 1 hop distance to both ,IN as well as SN.
IN Broadcasts cooperative detection message to all nodes of INSN1H list.
For each x Є INSN1H
   {
      Broadcast RREQ_pkt ( with DN being set to IN )
     Upon receiving RREP_pkts
If( received RREP_pkt is from SN )
   {
            Send a check packet(CP) to IN via this
           route and a notification of this CP to IN
   }
         Else
```

```
        {
              Discard the RREP_pkt( as it doesn't
        require further processing)
        }
    }
IN waits for two WT to collect the Notification
packets from nodes of INSN1H list.
IN put Node_id of sender of Notification_pkt to VT's
Voter field.
IN waits for three WT for CP from SN.
    {
       If IN receives CP from SN
          {
           Mark "False" to the "Suspicious Value"
           field of VT and send
           notification packet to cooperative node.
          }
       Else
          {
           Mark "True" to the "Suspicious Value"
           field of VT.
          }
    }
If (All the entries of "Suspicious Value" field are
"True")
  {
   It's a Black Hole in the network.
  }
If (All the entries of "Suspicious Value" field are
"False")
  {
   It's not a malicious node, remove it from Black
   List.
  }
End
```

**Explanation of Proposed Algorithm for Co-operative detection:**

1) First of all find out the neighbors of first suspicious called SN (1) node .
2) Now find out the common nodes in the two neighbor lists , one is list of nodes which are at distance of 1 hop from source node ( also called initial detection node) , this list is called IN1H and other is the neighbor list ( SN1H ) of SN (1) or first suspicious node by taking the intersection of both the lists. We call the final list as INSN1H, i.e list of nodes which are at distance of both the nodes, IN as well as SN.
3) After finding out the nodes with 1 hop distance from both, the source and the first suspicious node in last step, source node will broadcast the cooperative detection message to all nodes in INSN1H list.
4) Now every node in INSN1H list will broadcast the RREQ_pkt with destination node being set to IN ( initial detection node ).
5) Upon receiving the RREP_pkts from different nodes do the following :

   5.1) If the RREP_pkt is from SN then
   - Send check packet intentionally to IN via SN.
   - Send Notification of this check packet to IN directly (as its at only 1 hop distance from IN )

   5.2) Else, if the packet is not from SN simply discard the packet as it doesn't require any further processing.

6) IN will wait till two Waiting Time ( pre-defined) time to collect the notification packets from nodes in INSN1H list.
7) IN will also maintain a table called Voter Table ( denoted as VT ). VT consists of 2 fields: "Voter" and "Suspicious Value". The format of the voter table is shown in Table 1.
8) Put the Node_id of sender of notification packet to "Voter" field of VT.
9) IN waits for three WT for check packet from SN. If IN receive check packet from SN , mark " False" to the "Suspicious Value" field and send Notification to cooperative node that CP is received.
10) After waiting for three WT for check packet, if IN doesn't receive any check packet from SN, mark "True" to the "Suspicious Value" field of VT.

**Table 1 : Format of Voter table:**

| Voter | Suspicious Value |
|---|---|
| 2 | True |
| 3 | True |
| 4 | True |
| 5 | True |

If all the entries in VT's Suspicious Value are TRUE then its surely a malicious node and ALARM packet is sent to all its neighbors to warn them against malicious node.

11) If all the entries in VT's Suspicious Value are FALSE then its not a malicious node, remove it from Black List.
12) If some values are TRUE and some are False in VT's Suspicious Field then decision is taken on the basis of voter's count in favour and against that particular node.

Fig 4 shows the flow chart of cooperative detection process.

IJSER

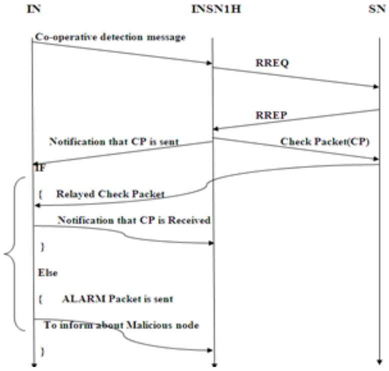
IN
INSN1H
SN
Co-operative detection message
RREQ
RREP
Notification that CP is sent
Check Packet(CP)
IF
{ Relayed Check Packet
Notification that CP is Received
}
Else
{ ALARM Packet is sent
To inform about Malicious node
}

**Fig. 4: Flow chart for co-operative detection**

## 4. RESULTS & CONCLUSION

In this paper, algorithms to remove Black Hole attack have been proposed and analysis is done experimentally by considering different scenarios. Implementation is carried out on .NET platform for various scenarios. Performance of the ADOV protocol, in terms of packet delivery ratio, is analyzed when the network is under Black Hole attack. The packet delivery ratio was measured in the range of 5% - 25% of actual packets delivered. Whereas, on the application of the proposed algorithm, BRTCM, when the network is put under the Black hole attack, the packet delivery ratio has been increased up-to 80-90% as shown in Figure 5. This increase in PDR is at cost of delay and routing overhead but increase in performance is more compared to increase in delay and overhead.

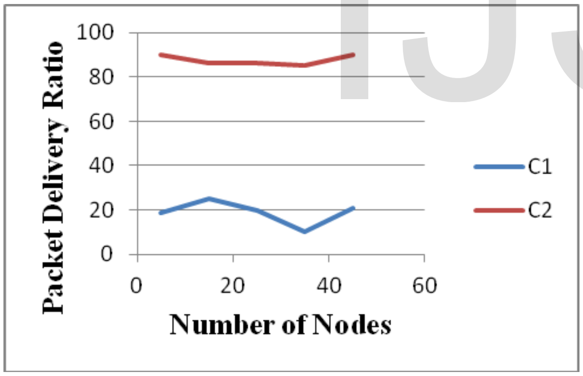

**Fig. 5: Performance of proposed algorithm(BRTCM)**

**Legend:**
**X-axis:Number of nodes**
**Y-axis: Packet Delivery Ratio in %age(PDR)**
**C1:Normal AODV under Black Hole Attack**
**C2:Using BRTCM**

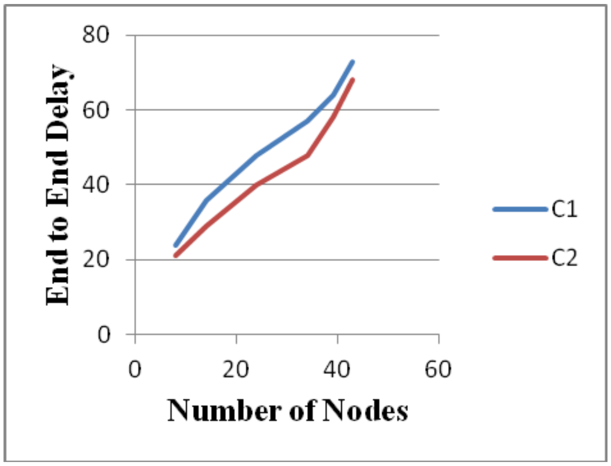

**Fig. 6: The effect of proposed algorithm(BRTCM) on End to End delay:**

**Legend:**
**X-axis:Number of nodes**
**Y-axis: End to End delay in ms**
**C2:Normal AODV under Black Hole Attack C1:Using BRTCM**

As shown in Figure 6, when BRTCM is used in network under black hole attack, End-to-End delay got increased due to extra processing that is involved in proposed algorithm to detect and remove Black Hole Nodes. Furthermore, End-to-end delay increases as number of nodes are increased.

Theoretically we can say that proposed algorithms will result into the better packet delivery ratio but at the cost of increase in End-to-End delay because of processing required in our proposed algorithm.

From the above discussion it can be concluded that security is the main concern for providing secure communication between the nodes participating in MANET. One of the security threats can be caused by a malicious node which is part of MANET. The communication should be secure from such malicious node so that cooperation of the network should not be compromised. Malicious node can attack on all the layers of the protocol stack by changing their position slightly from initial position to the other position and delay or drop packet forwarding. One of the malicious nodes can be a Black Hole node. Black hole node can absorb the packets passing through itself in such a way that sending node will assume that packets have reached the destination.

## 5. FUTURE SCOPE

The proposed algorithm is efficient in detection of Black Hole nodes and its removal from network but improvement can be done in mainly two directions as follows:

- **End-to-End Delay:** Due to the processing involved in our proposed algorithm, end to end delay will get increased. Further improvement can be done to decrease the end to end delay alongwith the successful removal of Black Hole nodes.

- **Routing Overhead:** In our proposed algorithm, control packets like alarm packet, notification packets and

IJSER

check packets results in increase of routing overhead. Improvement can be done to reduce the transfer of packets involved hence to decrease the routing overhead involved.

## REFERENCES

[1]Poongothai T. and Jayarajan K., "A non cooperative game approach for intrusion detection in Mobile Adhoc networks", Internatonal conference of computing, communication and networking (ICCC), 18-20 Dec 2008, St. Thomas, VI, pp 1-4.

[2] DjamelDjenouri and LyesKhelladi, "A survey of security issues in mobile ad hoc and sensor network", IEEE communications Surveys and Tutorials journal,Volume 7, Number 4, 2005, pp 2-29.

[3] Michele Nogueria Lima, AldriLuiz dos Santos and Guy pujolle, "A Survey of Survivability in Mobile Ad Hoc Networks", IEEE Communications Surveys and Tutorials COMSUR), Volume 11, Number 1, 2009, pp 1-3.

[4] NishuGarg and R.P Mahapatra, "MANET Security Issues", International journal of Computer Science and Network Security (IJCSNS), Volume 9, Number 8, 2009, pp. 241-246.

[5] Wenjia Li and Anupamjoshi, "Security Issues in Mobile Ad hoc Networks – A Survey". Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore Country, 2006.

[6] Bing Wu, Jianmin Chen, Jie Wu and MihaelaCardei, "A Survey on Attacks and countermeasures in Mobile Ad Hoc Networks", Wireless/Mobile Network security, ch-12,2006.

[7] HesiriWeerasinghe "Preventing Cooperative Black Hole Attacks in Mobile Ad Hoc Networks: Simulation Implementation and Evaluation", Proceedings of the Future Generation Communication and Networking, Volume 2, 2007, pp 362-367.

[8] Mehdi Medadian, M.H. Yektaie and A.M.Rahmani, " Combat with Black Hole Attack in AODV routing protocol in MANET", First Asian Himalayas International Conference on Internet (AH-IC12009), 3-5th Nov, 2009.

[9] Bo sung Yong, Guan Jianchen and Udo W. Pooch, " Detecting Black-hole Attack in Mobile Ad hoc Networks", The Institution of Electrical Engineers (IEE), Volume 5, Number 6, 2003, pp 490-495.

**Authors:**

**Aditya Bakshi is currently working as Assistant Professor, Deptt. of Computer Science & Engineering, Lovely Professional University, Jalandhar, Punjab, India**

**Atul Mishra is currently working as Associate Professor, Deptt. of Computer Engineering, YMCA University of Science & Technology Faridabad, India**

**Heena Batra is currently working as software engineer in Tech. Mahindra, Noida, India**

IJSER