International Journal of Networks and Communications 2020, 10(1): 1-9
DOI: 10.5923/j.ijnc.20201001.01

Design and Implementation of Resilient Cooperative Bait Detection Technique to Curb Cooperative Black Hole Attacks in MANETs Using DSR Protocol

Ephantus Gichuki Mwangi*, Geoffrey Muchiri Muketha, Gabriel Ndungu Kamau

School of Computing and Information Technology, Murang'a University of Technology, Murang'a, Murang'a, Kenya

Abstract Mobile Ad-hoc networks (MANETs) are unique type of wireless networks that are infrastructureless and with no centralised management. Nodes in MANETs act as both routers and hosts. The nodes are free to join and leave the network. Routes are established by use of special routing protocols. Mobility of nodes makes the network topology constantly dynamic. The unique characteristics of MANETs make their security a challenging endeavor. MANETs are prone to a range of security attacks such worm hole, Sybil, black hole, among others. Blackhole is a form of denial of service (DoS) attack. The black hole nodes work in association forming cooperative black hole attacks that drop or redirecting data packets. This compromises the communication process in mission critical areas. The paper proposes a Resilient Cooperative Bait Detection Technique (RCBDT) using DSR protocol to curb collaborative black hole attacks in MANETs. The proposed technique uses source node address as the bait address. Further, RCBDT uses an algorithm that checks nodes energy levels before engaging them in packet transmission. The technique was designed, implemented and simulated in Network Simulator Version 3(NS-3). The proposed technique was compared with Cooperative Bait Detection Scheme (CBDS) and Extended Cooperative Bait Detection Scheme (ECBDS). Simulation results indicate that the proposed technique is superior to benchmark techniques in terms of Packet Delivery Ratio (PDR), End-to-End Delay and Routing Overheads.

Keywords Mobile Ad-hoc Networks, Routing protocol, Network security, Network simulator, Bait detection technique, Cooperative black hole attack

1. Introduction

Mobile Ad-hoc Networks (MANETs) are unique type of wireless networks that are infrastructureless, decentralized and without any management authority. The networks have a dynamic topology since nodes freely join and leave the network at their own will. Globally, whenever the established communication infrastructures are brought down by disasters such as earthquakes, storms, eruptions or even terrorism, there is always a need for immediate intervention with alternative forms of communication. MANETs are the preferred choice of communication in such mission critical operations. Application areas of MANETs range from mission critical situations such as rescue mission, military operations, expeditions such as mountain climbing, vehicular communication, among other areas. Nodes in MANETs cooperate to forward data packets from source to destination using special routing protocols. Every node in a MANET acts as both a router and a host [1], [2].

In MANETs, a node wishing to communicate with other nodes establishes a route using special routing protocols [1], [2]. Several routing protocols have been designed to optimize MANETs routing performance [2], [6]. The major issues involved in designing MANETs routing protocol are dynamic network topology, constrained bandwidth, limited battery power, error prone wireless channel, and node mobility. These unique features of MANETs make most of the security solutions designed for wired networks inappropriate for mobile ad-hoc networks. The dynamic nature of MANETs makes it difficult to establish secure ad-hoc routing protocols [3].

MANETs routing protocols are categorized into three types: reactive routing protocols (on demand), proactive routing protocols (table driven) and hybrid protocols. In reactive routing protocols, routes are created on-demand whenever a source node wishes to send data packets to a destination node. This means that only nodes which participate in active route maintain routing information. Adhoc On-Demand Vector (AODV), Dynamic Source Routing (DSR) and Link Aware Routing (LAR) are some

* Corresponding author:
egmkuc@gmail.com (Ephantus Gichuki Mwangi)
Published online at http://journal.sapub.org/ijnc

of the examples of reactive routing protocols [6]. In proactive protocols, each node maintains complete routing information of the network. Change in the network topology due to nodes mobility leads to automatic updating of routing tables in all the nodes. Some of the examples of proactive routing protocols are Destination Sequenced Distance Vector (DSDV), Global State Routing (GSR) and HSR. Hybrid protocols are as a result of blended features of both proactive and reactive routing protocols [4].

Nodes in MANETs communicate using open wireless medium which paves way for an attacker to easily join and intercept the communication process. Further, the unique features of MANETs have introduced an underlying complex security problem [5], [7]. The issue of constrained network resources and mutual cooperation amongst nodes have made MANETs vulnerable to a range of network security threats that threaten the integrity of communication process [7]. MANETs are susceptible to various denials of service (DoS) routing attacks which make the communication process impossible. [5], [7], [25].

Security of MANETs is essential in preventing the data loss that could be caused by different types of attacks that target these networks. Black-hole attack is one of the popular active attacks that cause harm to the network by dropping data packets between any two communicating nodes that establishes a connection [7]. For instance, when a source node sends Route Request (RREQ) packets over the network in order to establish communication with the destination node, the RREQ packet can be responded to by any node in the network that has a route to destination. The open form of communication in MANETs paves way for malicious nodes to participate in the communication process with malicious intentions. For instance, when the black hole nodes receive the RREQ packet from the source node, they masquerade to be genuine by sending fake RREP packets with the shortest and freshest route to destination. This entices the source node hence making it to select the route with black hole nodes as an optimal choice for data transmission. Once the black hole nodes receive the data packets from source node, instead of forwarding them to the destination node they discard or reroute them. Further, black hole nodes collaborate with each other in order to launch collaborative attacks known as 'cooperative black hole attacks". The cooperative black hole attacks are more harmful to a network than any other form of attack [8], [20].

The existing CBDS and ECBDS techniques suffer from end to end delays due to the fact that they use next hop neighbours' address as the bait address. The techniques take time for a source node to identify and use bait address from the immediate neigbours. Further, CBDS and ECBDS engage genuine nodes in transmission process without checking their energy levels. This makes nodes with energy levels below the threshold level act selfishly. Selfish nodes drop data packets in order to save energy for its sustenance.

Our objective in this paper is to propose a resilient cooperative bait detection technique (RCBDT) using DSR protocol to detect and prevent cooperative black-hole attacks in MANETs. In order to achieve this we used source node's self-address as the bait address; this saves transmission bandwidth, node's energy and time. The bait concept is borrowed from the fishing industry and its purpose is to entice a prey. Further, RCBDT uses an algorithm that checks energy levels for all genuine nodes in MANET before engaging them in any transmission. In case there are nodes whose energy levels are below the threshold, it gives alerts to the source node. In our simulation experiment the bait concept was used by source nodes to lure malicious nodes by sending fake route requests. In return, malicious nodes sent fake route replies which alerted the source node of the presence of malevolent nodes in the network. This triggered the source node to start the process of reverse tracing which detected and eliminated the malicious nodes in the network.

The technique was designed, implemented and simulated in a Linux environment using Network Simulator Version 3 (NS-3). The resilience of the proposed technique was tested alongside two benchmark techniques, namely, the Cooperative Bait Detection Scheme (CBDS) and the Extended Cooperative Bait Detection Scheme (ECBDS).

The rest of the paper is organized as follows; section 2 presents related works, section 3 presents methodology used, section 4 describes the simulation environment, section 5 presents the results and discussions, and section 6 presents the conclusions and future work.

2. Related Works

Abdelshafy and King [6] introduced black hole resisting mechanism (BRM) on AODV routing algorithm to detect and avoid black hole attack in MANET. During the simulation experiment, AODV and BRM AODV routing algorithms were subjected to black hole attacks in order to study their performance. Simulation results showed that BRM-AODV was superior in all network performance metrics over AODV and SAODV routing protocols. The proposed mechanism detected black hole nodes easily regardless of the number of malicious nodes. Further, the results of study showed that BRM can effectively increase the performance of AODV routing algorithms in MANETs. However, BRM AODV was not able to detect collaborative black hole attacks. Additionally, performance metrics such as packet delivery ratio, throughput and routing overhead needs to be enhanced in the new mechanism in order to increase network performance. Reviewed literature indicates that so far no researcher has come up with a modified version of the proposed mechanism.

Ukey [16] proposed a 1-2ACK technique for preventing routing attacks in MANETs. In this technique, all the nodes that form a path for transmitting packets are grouped into sets of three adjacent nodes. When a node sends a packet, it waits for an acknowledgement ACK1 from the Rnode (right node) of its own set and ACK2 from Rnode of the next set. If a node does not receive both of the acknowledgements from both sets, then there exists a malicious node. In this

technique, the need for extra control packets introduces routing overhead as well as end to end delays.

In [17], Hiremani & Jadhao proposed a security technique to prevent cooperative black hole attacks by using modified extended data routing information (MEDRI) table at each node with the routing table of AODV protocol. Simulation results showed that this technique was capable of detecting both consecutive and non-consecutive cooperative black hole attacks. The MEDRI table has the capability of recording and maintaining a history of the previous malicious nodes. This history is used for future discovery of secure paths from source to destination. However, this technique suffers from routing overhead and end to end delay due to the introduction of data packets in the MEDRI table.

In [9], Mistry et al. proposed a security technique in which a source node after receiving the first RREP waits for particular time interval and stores all the RREP's received during that interval. The source node analyses all the RREP's and ignores all the RREP's having a very high sequence number. In this technique, it is observed that there was an increase in the average end to end delay. Further, a heuristic approach was used in deciding the time interval for a node to wait.

Su et al. [10] proposed an anti-black hole technique that uses intrusion detection system (IDS) nodes for the detection of black hole nodes. In this technique, every IDS node estimates the suspicious value of a node based on the difference between the numbers of RREQ's and RREP's forwarded by a node. If the suspicious value of a node goes beyond the threshold value, then the IDS node broadcasts a block message to all nodes on the network in order to work together in mitigating the black hole node. Once a node receives the block message from the IDS, it places the malicious node into its blacklist. In this technique, it was noted that extra nodes had to be placed in the network and every IDS had to sniff the RREQ and RREP's of all nodes, this was an extra overhead for a MANET with many nodes.

Sen et al. [3] proposed a technique in which a node (IN) generating the RREP has to send the Data Route Information (DRI) entry of its next hop neighbour (NHN). The source node then sends FREQ request to the NHN. Further, NHN node replies FREP with DRI entry of IN. The source node cross checks the entries of IN and NHN and if they match then the node is genuine, else IN is malicious. It was observed from this technique that the FREQ and FREP extra control packets are required which increases routing overhead.

Gupta et al. [11] proposed a technique which uses Ad-hoc On-Demand Multipath Distance Vector (OMDV) to provide multiple paths during routes discovery process. The intermediate nodes in the network have multiple paths which lead to the destination node. However, the source node selects only one path among them. Each node in the network maintains a legitimacy of all nodes that are under its neighbourhood. In this technique, nodes try to avoid paths that pass through nodes with legitimacy value less than threshold. This helps in identifying the nodes behaving maliciously, hence avoiding them. This method works fine with one black hole node but dealing with cooperatives black hole nodes would be a tedious undertaking.

In [12], Saha et al. presented a Two-Level Secure Re-routing (TSR), a novel routing architecture for MANETs which is attack resilient. TSR employs a two-level approach that uses Local Supervision (LS) and Congestion Window Surveillance (CWS) modules to detect network attacks at the transport layer. TSR then responds to these attacks using the Alternate Route Finder (ARF) module that executes re-routing at the network layer. Simulation analysis showed that TSR is resilient against a variety of insider attacks as well as protocol-compliant attacks. This architecture can also be used in controlling black hole nodes as they are a variant of DoS attacks. However, LS and CWS modules introduced routing overhead during data transmission.

In [13], Bhosle proposed a technique based on watchdog and pathrater mechanism. In this technique, each node maintains two tables: pending packet table and node rating table. Every node stores packet forwarded in the pending packet table and overhears its neighbours. If the neighbouring node sends the packet in the forward direction, then the value of the packet forwarded in node rating table is incremented. Further, if the packet is dropped, then that value is decremented. If the value of dropped packets in the node rating table goes beyond a threshold value, then that node is considered to be malicious. This technique requires extra memory space to store multiple tables. Further, extra time is incurred for frequently monitoring of the two tables. This technique suffers from routing overhead due to the two tables introduced.

Thachil [14] presented a technique in which every node performs overhearing of neighbouring nodes and calculates their trust value. Each node keeps a copy of a packet in the cache before forwarding it and then overhears the packets forwarded by the neighbouring nodes. If a packet forwarded by the neighbouring node matches with the packet in the cache then the sending node believes that the neighbouring node is genuine; otherwise its trust value is decremented. Each node maintains a trust value that is updated dynamically and if the trust value of a node goes beyond threshold that node is considered to be malicious. In this technique, it was observed that routing overhead at a node level increased due to the fact that a node had to keep copies of packets in its cache and had to overhear all its neighbours.

In [15], Bindra et al. proposed a security technique using AODV protocol that detect and prevent black hole and gray hole attacks. The technique maintains an extended data routing information (EDRI) table at each node in addition to the routing table of AODV protocol. The EDRI table is an extension of DRI Table and is able to identify cooperative black nodes in MANETs. Further, the technique can discover secure paths from source to destination by avoiding multiple black hole nodes acting in cooperation. Limitation of this technique is that malicious nodes have to be in sequence while acting in cooperation for them to be discovered by the

algorithm. Additionally, routing overhead is experienced due to the many packets introduced in the EDRI table. Further, the algorithm needs to be optimized for efficient usage.

Gaikwad & Ragha [18] proposed a technique which uses cooperative cluster agents (CCAs) to detect and avoid cooperative black hole attacks in MANETs. In this technique, DRI and SRT-RRT tables are used as input to CCAs. Simulation results showed that the technique successfully detected black hole and cooperative black hole nodes in MANETs. Further, the technique identified secure routing path from source to destination by avoiding the black hole nodes. The new technique was compared with the standard AODV protocol and proved to be more superior in terms of throughput, packet delivery ratio and end to end delays. However, this technique experiences routing overhead due to the introduction of DRI and SRT-RRT tables. Additionally, packet delivery ratio and throughput need to be further improved to hit the optimum levels.

In [19], Dumne and Manjaramkar proposed a hybrid defence architectures known as Cooperative Bait Detection Scheme (CBDS) based upon DSR mechanism. This scheme uses proactive and reactive defence architectures to detect malicious nodes that launch collaborative black hole attacks. Simulation results show that CBDS using AODV performs better than DSR protocol and CBDS using DSR in terms of throughput and packet delivery ratio. From the above results, CBDS using AODV was considered as a better alternative because it reduced routing overhead. However, the new technique didn't perform better than CBDS using AODV in terms of throughput and packet delivery ratio. This gives room for enhancement of the new technique in order to improve performance efficiency. Further, introduction of reverse tracing technique led to the introduction of end to end delay in data transmission.

Emimajuliet & Thirilogasundari [20] proposed a Modified Cooperative Bait Detection Scheme (MCBDS) for defending collaborative attacks caused by black hole and jellyfish. Simulation results indicated that MCBDS along with DSDV protocol performs better than the DSR and 2ACK scheme. However, this scheme suffers from routing overhead compared to DSR protocol. A hybrid technique needs to be explored which would be a combination of MCBDS with other techniques in order to effectively secure routing of packets.

3. Methodology

The first sub section describes in detail the design of Resilient Cooperative Bait Detection Technique using a flowchart. Next sub section describes the algorithms used to implement the technique. Further, next sub section describes the simulation environment. Additionally, the next sub section discusses the results of simulation of RCBDT technique in NS-3 and comparison with benchmark techniques. Finally, the last sub section gives the conclusion and future work.

3.1. Proposed Resilient Cooperative Bait Detection Technique

The proposed RCBDT uses a four key phases in its operation. The phases include; a) Initial Self-Address Bait Phase, b) Reverse Tracing Phase, c) Reactive Defense Phase, d) Refreshing phase.

a) Initial Self-Address Bait phase

The phase uses address of the source node (self address) as the bait address. This is opposed to initial bait phase of CBDS and ECBDS (used as benchmark techniques) which uses the address of one hop neighbour as its bait address. The source node sends bait RREQ with its own address as the destination address and waits for a reply from other nodes in the network. Any node that sends RREP packet is considered as malicious. This triggers the reverse tracing program as indicated in the next phase.

Using self address as the bait address makes the source node to save its battery power and which could have been used when communicating with one hop step neighbour in order to generate the bait address. Further, this also saves time as no engagements are involved between source node and its one hop step neighbours, hence improving network efficiency.

b) Reverse Tracing Phase

In this phase, the reverse tracing program would be started to detect the routes with malicious nodes. If the routes were secure, no node could have sent a RREP packet due to the fact that the source node had broadcasted its own address (self address) as the bait address. When malicious nodes receive a RREQ, they respond to the source node with fake RREP packets. This triggers the reverse tracing program which tries to identify the dubious paths and exact location of the malicious nodes through the route replies (RREPs). The reverse tracing program then forms a set (Nd) of all the nodes that sent back the fake RREPs and saves them under malicious nodes alarmed list in the cache. The source node uses this set (Nd) to form a malicious node detected list (considered as the black hole list) and then sends an alarm to all other nodes in the network about the existence of the malicious nodes. The malicious nodes detected list helps other nodes to establish temporary a set of trusted routes in the network which are saved in their caches.

$$Nd = \{n1, n2, n3, ..., nm\} \quad (1)$$

This phase saves a lot of node's battery power and memory space as no set difference operation is computed (like in the case of ECBDS) in order to identify the malicious nodes. In ECBDS, when the node received RREP, it would perform a set difference operation between the address List P = {n1,…, nk....nm...,nr} recorded in RREP and saved RREQ' Kk = {n1,…. nk} before caching the routing of receiving nodes, and consequently obtain P-Kk = Kk' {nk+1,…nm…,nr}. This process took a lot of node resources (battery power and memory space) hence limiting its ability to participate in subsequent data transmission processes.

c) Reactive Defense Phase

In this phase, all the nodes in the malicious node detected list (black hole list) are deactivated by setting their life-bit bit to zero (sleep mode) to prevent them from further activation. Further, this information is broadcasted to all other nodes in the network. This mode makes the malicious nodes not to participate in any network operation during the time of data transmission.

d) Refreshing Phase

In this phase, nodes' route caches are refreshed. The invalid routes and broken links are deleted. The newly established temporary trusted routes are saved in the nodes caches. Further, the newly recorded routes in the cache are prioritized and used to determine the optimal route to channel data packets based on current status of the network. These routes remain valid as long as there are no broken links or no gratuitous routes established. Additionally, the life-bit of nodes classified as genuine is incremented by one and information circulated to all other nodes in the network. These nodes are allowed to participate in all network operations as long as their battery power is above the threshold level. Figure 1 shows a flowchart of the proposed design [22].

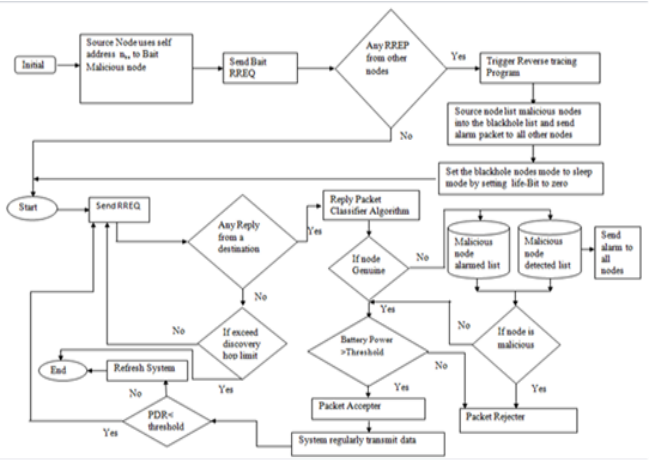

Figure 1. Flowchart of the Proposed Resilient Cooperative Bait Detection Technique

3.2. Algorithm for the Proposed Resilient Cooperative Bait Detection Technique

The purpose of this algorithm is to describe step by step process through which RCBDT baits, identifies and eliminates the malicious nodes in the network.

a) RCBDT Algorithm

The input of RCBDT algorithm is the bait RREQ. In this algorithm, the source node broadcasts its own address as the bait address over the MANET. If there are malicious nodes in the network, they send fake RREP packets to the source node. This triggers a reverse tracing program which locates the exact routes with malicious nodes. Reverse tracing program identifies the exact position malicious nodes.

```
Start [Algorithm]
  Source Node sends Bait RREQ with its address as the Bait Address
     If (RREPs from other nodes) {
        Trigger Reverse Tracing Program
        List all nodes that sent RREPs as Malicious nodes in the Black hole
        list
        Send Alarm Packets to all other nodes with the blackhole list
        Set Blackhole_Node _Modes to Sleep_Mode
      Set Life-bit to Zero
      Goto Start Transmission_Process   // Start symbol in the flowchart
      Send RREQ
     Check RREP from destination
   }
  else
       {
         Do {
                If (Nodes Battery_power>Threshold) {
              Accept node for transmission
              Start Transmission_Process   // Indicated by Start symbol in
             the flowchart
                If ( PDR>Threshold_PDR){
              Compelete_Data_transmission by routing all data packets
              Acknowledge Successful end of data transmission
              Refresh System
              End transmission_process }    // End Data transmission
              process
                      else{
                         Send FRREQ // Further_Route Request for
                          establishing another route
                            If (RREP from Destination node) {
                            Start Data Transmission Process
                          Call Packet_Classifier_Algorithm ()
                         }
               else {
                     If (Node Exceed Discover_Hop_Limit) {
                   End Data_Transmission_Process   // Indicated by End
                  symbol in the flowchart
                    }
                    else
                       {
                          Send_FRREQ   // Send Further_Route Request
                        }
                  }
       } while (No RREPs Received from other Nodes)
   }
End [Algorithm]
```

Algorithm 1. RCDBT Algorithm

The malicious nodes are then listed in the blackhole list. Additionally, an alarm signal with the blackhole list is sent to all other nodes in the network. Further, all the blackhole nodes are set to sleep mode by setting their life-bit to zero; this means that they can't participate in any data transmission activities. The fact that the life-bit of malicious node is set to zero; means they cannot be reactivated during the duration of data transmission. Otherwise, in case there

are no RREPs received from other nodes, the process of data transmission is started having eliminated all the black holes nodes in the network. The source node sends a RREQ through the network, if there is a RREP from a destination node, packet classifier algorithm is activated in order to determine whether RREP is genuine or not. Otherwise, if a node exceeds discovery hop limit, the transmission process is ended; if not so, a Further Route request (FRREQ) is sent over the network. Algorithm 1 shows a step by step procedure of the RCBDT algorithm.

b) Packet Classifier Algorithm

The inputs of Packet Classifier Algorithm are RREPs from intermediary nodes that have routes to the destination node. The purpose of this is to determine the genuineness of nodes sending the RREPs. Packet Classifier Algorithm compares the destination address in RREQ packet sent by source node with the destination address in the RREPs sent by intermediary or destination node. If the address matches, the node is classified as genuine; otherwise the node is classified as malicious and listed under malicious node alarmed list/ malicious node detected list. Further, an alarm with malicious nodes list is circulated to all other nodes in MANET. Additionally, all malicious nodes are rejected from participating from the communication process [27]. Algorithm 2 shows a detailed procedure of the Packet Classifier Algorithm.

```
Packet_Classifier_Algorithm () {
   Capture Destination Address from node(s) that sent RREP(s)
   If (Dest_Address==(Dest_Addess_In_RREQ of Source_Node))
         {
           Classify Node as Gennuine_Node
           Check Genuine_Node_ Battery_Power // by calling
           Energy_Aware_Algorithm ()
          }
   else {
          Consider Node that sent RREP as Malicious_Node
          Register Malicious_Nodes to Malicious_Nodes_Alarmed_List
          AppendMalicious_Nodes_Alarmed_List to
           Malicious_Nodes_Detected_List
          Send larm_Signal to All Nodes with
         Malicious_Nodes_Detected_List
          Record Malicious_Nodes_Alarmed_List and
         Malicious_Nodes_Detected_List into Nodes_Cache
        Rejected node that sent RREP Packet
     }
}
```

Algorithm 2. Packet Classifier Algorithm

c) Energy Aware Algorithm

The energy aware routing algorithm receives all the nodes that sent genuine packets, their energy levels are tested for them to be allowed to participate in the transmission process [21]. Nodes with energy levels above the threshold levels are allowed to participate in the data transmission process while the rest are rejected [24], [26]. The accepted nodes are allowed to participate in transmitting all the data packets to the destination node, if at the end of the transmission process the system meets the Packet delivery Ratio (PDR) threshold; the transmission process ends successfully[28]. Otherwise, packets delivered are discarded and transmission process started afresh. Algorithm 3 shows a detailed process of energy aware routing algorithm [21], [22], [23].

```
Energy_Aware_Algorithm ()
 {
   Accept all Genuine_Nodes as inputs
   If (Gennuine_Node_Battery_Power>Nodes_Threshold_Battery_Power)
     {
      Accept thisNode && its RREP Packet
      List thisNode as Safe_Node
      Enlist all Safe_Node in Trusted _Nodes _Register
      Form Tempolary_Trusted_ Routes using Trusted_Nodes_Register
      Register Tempolary_Trusted_ Routes in Nodes_Caches
      Select Freshest Trusted_Route and Trasmit_Data Packets
      }
   else {
         Reject nodes
         }
}
```

Algorithm 3. Energy Aware Algorithm

4. Simulation Environment

Table 1. Simulation Experiment Parameters

Parameter	Value
Channel Type	Wireless Channel
Simulation Time	400 seconds
Number of nodes	50
MAC type	802.11
Routing Technique	RCBDT
Routing Protocol	DSR
Movement Model	Random Way Point
Traffic model	Constant Bit Rate (CBR)
Receiving Antenna	Omnidirectional Antenna
Transport layer protocol	User datagram protocol (UDP)
Radio Transmission range	250 meters
Packet size:	512 bytes
Sending frequency	4 packets/second
Simulation Area	1500*1000 meters
Node speed	1-10 meters/second
Number of black hole nodes	2,4,6

To compare the effectiveness of the proposed RCBDT technique, simulation environment was setup in NS-3 Simulator. Simulation area was set in a rectangular pane measuring 1500 by 1000 meters. Fifty genuine mobile nodes were installed. Further, two, four and six blackhole nodes were installed in our three simulation scenarios respectively. The black hole nodes used simple attack model to entice

other nodes in the network. Channel of communication among nodes was set to User Datagram Protocol (UDP). DSR protocol was set as the routing protocol for all the nodes in the network. In order for the nodes to manoeuvre within the simulation area, propagation model was set to Radom Way Point (RWP) model. The nodes were configured using radio waves in a manner that could enable them to receive signals from all directions using omnidirectional antenna. Constant Bit Rate (CBR) traffic model with a packet size of 512 bytes and sending rate of 4 packets/second was set to handle packet traffic. The simulation time for each scenario was set to 400 seconds. Finally, nodes' transmission range was set to a radius of radio range of 250 meters. Table 1 is a summary of the simulation parameters.

5. Results and Discussions

NS-3 Simulator was used to simulate the proposed RCBDT technique in Linux environment. Data generated by the Simulator was saved as text files of extension ".dat". The text files were then executed using Gnuplot software in order to generate the output. The generated output of RCBDT technique was compared against CBDS and ECBDS technique as chosen benchmarks. Packet Delivery Ratio, End-to-End Delay and Routing Overhead were used as the basis of our performance metrics. Figure 2 shows the simulation environment of RCBDT technique. The dots in red show the distribution of mobile nodes across the simulation area.

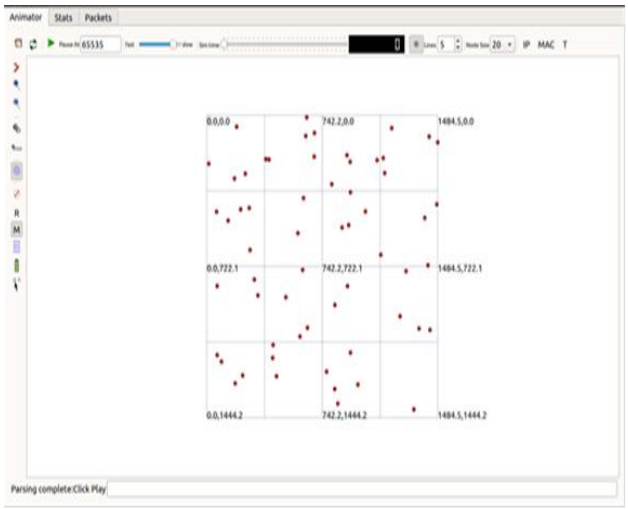

Figure 2. Simulation Interface for RCBDT Technique

a) Packet Delivery Ratio

This is the ratio of the total number of packets received by the destination node to the total number of packets sent by the source node.

$$PDR = \sum_{i=1}^{n} \frac{pktd}{pkts} \quad (2)$$

Where 'pktd' represents the total number of packets received at the destination node, while 'pkts' is the total number of packets transmitted by the source node to the destination node. Packet Delivery Ratio versus Pause Time for the three techniques was compared in the presence of cooperative blackhole nodes. From our analysis, as indicated in figure 3, RCBDT had a highest Packet Delivery Ratio compared to the benchmark schemes. The RCBDT technique had the highest Packet Delivery Ratio of 95%, while ECBDS and CBDS had 91% and 83% respectively. This implies that RCBDT technique is superior to the benchmark techniques due to the fact that it does not lose many packets to the adversary nodes during the transmission process.

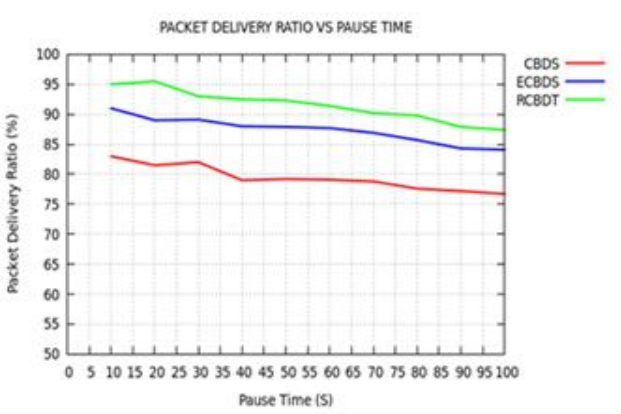

Figure 3. Packet Delivery Ratio versus Pause Time

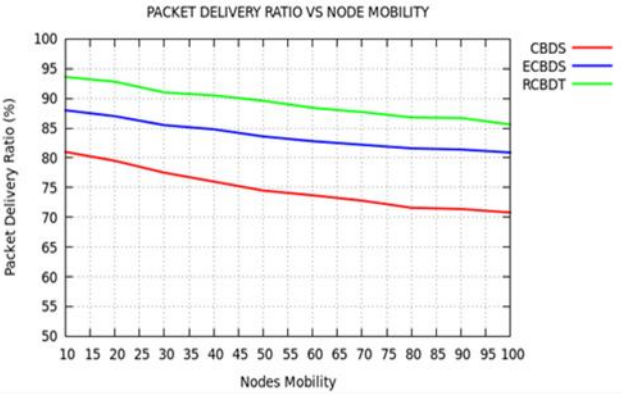

Figure 4. Packet Delivery Ratio versus Nodes Mobility

The Packet Delivery Ratio against nodes' mobility of RCBDT technique was compared with the benchmark techniques. As indicated in figure 4, it was noted that RCBDT had a highest Packet Delivery Ratio in the presence of cooperative blackhole nodes. This implies that RCDBT is robust enough to withstand higher nodes mobility during packet delivery, detect malicious nodes and maintain higher Packet Delivery Ratio than the benchmark techniques. From our findings, RCBDT had the highest Packet Delivery Ratio of 94%, while the benchmark techniques; ECBDS and CBDS had 88% and 81% respectively.

b) Routing Overhead

This is defined as the ratio of the total number of control packets transmitted to the destination node to the total number of data packets transmitted.

$$RO = \sum_{i=1}^{n} \frac{cpktd}{dpktd} \quad (3)$$

Where 'cpktd' is the total number of control packets sent to the destination node while 'dpktd' is the total number of data packets sent to the destination node. Results from our analysis as indicated in figure 5 show that on average RCBDT has a lower routing overhead in relation to CBDS and ECBDS used as the benchmark. This implies that RCBDT is more efficient in terms of bandwidth utilization during data transmission in the presence of cooperative black hole nodes. From our findings as indicated in figure 5, RCBDT had the lowest Routing Overhead of below 8% while ECBDS and CBDS had 15% and 19% respectively, an indication that most of the assigned bandwidth goes to the data packets than the control packets when compared to benchmark techniques.

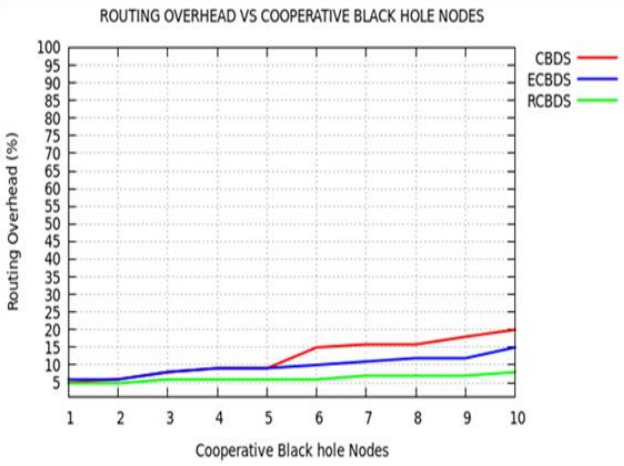

Figure 5. Routing Overhead versus Black hole Nodes

c) End to End Delay

The End to End Delay metric is a measure of the average time taken for a packet to be transmitted from source to destination. The results of the simulation are presented in figure 6.

$$ED = \sum_{i=1}^{n} \frac{dly}{dpktd} \quad (4)$$

Where 'dly' is the total time delay of packets received by the destination node and 'dpktd' the total number of packets received by the destination node. Finally, EED denotes the average end-to-end delay of the transmission process. From figure 6, RCBDT had a lower end-to-end delay compared to the benchmark techniques. This implies that RCBDT has a higher turn-around time in terms of RREQs and RREPs during data transmission. On average as the number of nodes increased in the network, RCBDT had an End-to End Delay of below 1.2 seconds compared to ECBDS and CBDS which had an average of below 1.3 and 1.8 seconds respectively. Therefore RCBDT is more efficient in terms of end to end delay management.

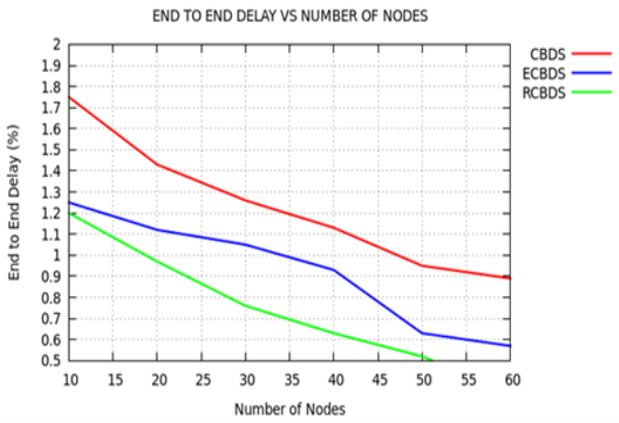

Figure 6. End to End Delay versus Number of Nodes

6. Conclusions and Future Work

MANETs are a unique type of wireless networks; their flexibility and ease of deployment have attracted a lot of attention in industrial application. However, MANETs are prone to a range of security threats due to their unique characteristics. Security is a key feature in any communication system. Guaranteeing security in MANETs is today's one of the biggest challenge. In this paper, we proposed a Resilient Cooperative Bait Technique (RCBDT) against cooperative black hole attacks in MANETs. Simulation results indicated that the proposed RCBDT technique is superior to both CBDS and ECBDS used as benchmark techniques in terms of Packet Delivery Ratio, End to End Delay and Routing Overhead used as performance metrics. This implies that the proposed RCBDT is a resilient and robust technique in MANETs' communications. The technique can withstand malicious attacks such as cooperative black hole nodes and still maintain better performance in any MANET communication environment compared to benchmark techniques.

As part of our future work, we intend to improve RCBDT technique by incorporating the aspect of trust component amongst nodes. This will further improve the effectiveness of the technique in mitigating cooperative black hole attacks with higher efficiency, improved packet delivery ratio, reduced end to end delays and minimal routing overheads.

REFERENCES

[1] Rutvij, H., Jhaveri, J., Sankita, P., and Jinwala, C. D., 2012, A Novel Solution for Gray hole Attack in AODV Based MANETs, In Proc. of Third International Conference on Advances in Communication, Network and Computing: Springer, 60-67.

[2] Boukerche, A., Turgut, B., Aydin, N., Ahmad, M., B¨ol¨oni, L. and Turgut, D., 2011, Routing protocols in Ad-hoc networks: a survey of Computer Networks, 55(13), 3032–3080.

[3] Jeenat, S. and Tasnuva, A., 2017, Securing AOMDV Protocol in Mobile Ad-hoc Network with Elliptic Curve Cryptography, International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, 539-543.

[4] Sagar, R. D., Chatur, P. N. and Nikhil, B. B., 2016, AODV-Based Secure Routing Against Black hole Attack in MANET, IEEE International Conference on Recent Trends in Electronics Information Communication Technology, IEEE, 319-326.

[5] Soufiene, D., Farid, N. and Zonghua, Z., 2011, Mitigating Packet Dropping Problem in Mobile Ad-hoc Networks: Proposals and Challenges", IEEE Communications Surveys & Tutorials, 13(4), 658 - 672.

[6] Abdelshafy, M. A. and King, P. J. B., 2016, Resisting Black hole Attacks on MANETs, 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 1048 - 1053.

[7] Sukanesh, R. Edsor, E. and Aarthylakshmi, M., 2016, Energy Efficient Malicious Node Detection Scheme in Wireless Networks, IEEE, 307-312.

[8] Sen, J., Koilakonda, S. and Ukil, A., 2011, A mechanism for detection of Co-operative Black hole attack in Mobile Ad-hoc networks, Second International Conference on Intelligent Systems, Modeling and Simulation, IEEE, 338-343.

[9] Mistry, N. Jinwala, D. C. and Zaveri, M., 2010, Improving AODV Protocol against Black hole Attacks, International Multiconference of Engineers and Computer Scientists, 2,(6), 1-6.

[10] Su, M-Y., Chiang, K-L., and Liao, W-C., 2010, Mitigation of Black-Hole Nodes in Mobile Ad-hoc Networks. International Symposium on Parallel and Distributed Processing with Applications, IEEE, DOI: 10.1109/ISPA.2010.74, 105-113.

[11] Gupta, S., Kar, S. and Dhararaja, S., 2011, BAAP: Black hole Attack Avoidance Protocol for Wireless Network", International Conference on Computer & Communication Technology (ICCCT), IEEE, 1-6.

[12] Saha, H. N., Bhattacharyya, D., Bandhyopadhyay, A. K. and Banerjee, P. K., 2012, Two-level Secure Re-routing (TSR) in Mobile Ad-hoc Networks, IEEE, 119-122, DOI 10.1109/MNCApps.2012.31.

[13] Bhosle, A. A., Thosar, T. P. and Mehatre, S., 2012, Black-Hole and Wormhole Attack in Routing Protocol AODV in MANET, International Journal of Computer Science, Engineering and Applications (IJCSEA), 2(1), 45-54.

[14] Thachil, F. and Shet, K. C., 2012, A trust based approach for AODV protocol to mitigate Black hole attack in MANET, International Conference on Computing Sciences, IEEE, 312-325.

[15] Bindra, G. S., Kapoor, A. Narang, A. and Agrawal, A., 2012, Detection and Removal of Co-operative Black hole and Gray hole Attacks in MANETs, IEEE, 3(11), 207-212.

[16] Ukey, A. S. A., Chawla, M. and Singh, V. P., 2013, I-2ACK: Preventing Routing Misbehavior in Mobile Ad-hoc Networks, International Journal of Computer Applications (0975 – 8887), 62(12), 345-353.

[17] Hiremani, V. A. and Jadhao, M. M., 2013, Eliminating Co-operative Black hole and Gray hole Attacks Using Modified EDRI Table in MANET, IEEE, 944-948, DOI: 10.1109/ICGCE.2013.6823571.

[18] Gaikwad, V. and Ragha, L., 2015, Security Agents for Detecting and Avoiding Cooperative Black hole Attacks in MANET, International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), IEEE, 306-311.

[19] Dumne, P. R. and Manjaramkar, A., 2016, Cooperative Bait Detection Scheme to prevent Collaborative Black hole or Gray hole Attacks by Malicious Nodes in MANETs, 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), IEEE, 486-490.

[20] Emimajuliet, P. and Thirilogasundari, V., 2016, Defending Collaborative Attacks in MANETs Using Modified Cooperative Bait Detection Scheme, International Conference On Information Communication And Embedded System (ICICES), ISSN: 978-1-5090-2552-7, 819-826.

[21] Allard, G., Minet, P., Nguyen, D. Q. and Shresta, N., 2006, Evaluation of the energy consumption in MANET", Adhoc-Now, Ottawa, Canada, 41-51.

[22] Bheemalingaiah, M. Naidu, M. M. and Rao, D.S., 2017, Energy aware Clustered based Multipath Routing in Mobile Ad-hoc Networks, *International Journal of Communications, Network and System Sciences, 2(5),* 1-24.

[23] Cao, L., Dahlberg, T. and Wang, Y., 2007, Performance evaluation of energy efficient Ad-hoc routing protocols, *Proc. IPCCC, IEEE,* 306-313.

[24] Rango, F., Guerriero, F., and Fazio, P., 2012, Link-Stability and Energy aware Routing Protocol in Distributed Wireless Networks. *Journal of IEEE Transaction on Parallel and Distributed Systems, 347-362.*

[25] Dorri, A., Kamel, S. R., and Kheyrkhah, E., 2015, Security Challenges in Mobile Ad-hoc Networks: A Survey, *International Journal of Computer Science & Engineering Survey (IJCSES), 6*(1), 15-29, DOI: 10.5121/ijcses.2015.6102.

[26] Guo, Z. and Malakooti, B., 2007, Energy Aware Proactive MANET Routing with Prediction on Energy Consumption, *International Conference on Wireless Algorithms, Systems and Applications*, IEEE, 287-292, DOI: 10.1109/WASA.2007.151.

[27] Shabbir, A., Khalid, F., Shaheed, S.M., Abbas, J. and Zia-Ul-Haq, M., 2015, Security: A Core Issue in Mobile Ad-hoc Networks. *Journal of Computer and Communications, 3(3),* 41-66, http://dx.doi.org/10.4236/jcc.2015.312005.

[28] Toh, C. K., 2001, Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad-hoc Networks, Communication Magazine, *10th International Conference on Practical Applications of Agents and Multi-Agent Systems,* IEEE, 39(6), 174-186.