An Approach to Defend Against Black hole Attacks in Ad Hoc Networks: Node Clustering AODV Protocol (CAODV)

by

Mnar Saeed Alnaghes
B.Sc, King Abdulaziz University, 2010

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTERS OF APPLIED SCIENCES

in the Department of Electrical and Computer Engineering

An Approach to Defend Against Black hole Attacks in Ad Hoc Networks:
Node Clustering AODV Protocol (CAODV)

by

Mnar Saeed Alnaghes
B.Sc, King Abdulaziz University, 2010

Supervisory Committee

Dr. Fayez Gebali, Co-Supervisor
(Department of Electrical and Computer Engineering)

Dr. Issa Traore, Co-Supervisor
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Fayez Gebali, Co-Supervisor

(Department of Electrical and Computer Engineering)

Dr. Issa Traore, Co-Supervisor

(Department of Electrical and Computer Engineering)

ABSTRACT

The flexibility of Mobile Ad hoc networks (MANET) and its characteristics introduce new security risks. One possible attack is the Black Hole attack which received recent attention. In the Black Hole attack, a malicious node uses the routing protocol to declare itself as having the shortest path to the node whose packets it wants to intercept. It is needed to understand this risk with a view to extract preventive and corrective protections against it. We introduce an approach that could stop this attack from happening in such a network by using an algorithm which controls the communications between nodes and let each node becomes identified and authorized in a group of nodes. In this algorithm, stable nodes, which called leaders, are responsible for routing and forwarding packets from source to destination nodes. This research reviews the black hole attack, and, explains the algorithm that helps throughput to be increased as a consequence.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful,

Alhamdullah that Allah is always close to us, hears us, and gives us the strength in good and bad moments. Alhamdullah for his blessings and guidance.

I would like to thank so many friends who were, and still are, with me during my trip but the list is going to be so long. So, to all of them I say, I really appreciate you and your help. I also would like to deeply thank:

My Parents, and Family, for supporting me in the low moments, and encouraging me to keep going on and achieve all my goals. Thank you so much for making me always happy.

Dr. Fayez Gebali, for his mentoring, support, encouragement, and patience. He supervised me with great suggestions and comments during my study. Thank you Dr. Gebali for your valuable comments and suggestions.

Dr. Issa Traore, and Dr. Yvonne Coady for their time, feedback, direction, and assistance. Thank you for your valuable comments.

The Department of Electrical and Computer Engineering, for the pleasant environment which they always provide.

Ministry of Higher Education in Saudi Arabia, for making this achievement easily possible.

DEDICATION

This work is dedicated to my lovely mother, Mrs. Ramziah A. Zagzog, and father, Dr. Saeed M. Alnughais and all my siblings, for being near supporting me at all stages and for their unconditional love.

Chapter 1

Introduction

1.1 Introduction

Mobile ad hoc networks (MANET) have been an interesting field of research because of the potential use in different situations where the infrastructure support to run a normal network does not exist, for instance, a war zone, a virtual class room, and an isolated remote area, etc. In the domain of mobile ad hoc networks, many researches are attempting to make MANETs less vulnerable to attacks while maintaining the network performance levels. As each node equally and fairly participates in the operation of the network, malicious nodes are difficult to identify. In MANET, nodes communicate with each other without a fixed infrastructure, they provide the connectivity by forwarding packets to other nodes. A node uses routing protocols in order to support its connectivity with other nodes such as ad hoc On-Demand Distance Vector (AODV) which is the protocol that we applied for this project, Dynamic Source Routing (DSR) which is a reactive routing protocol, and, Destination-Sequenced Distance-Vector (DSDV) which is a proactive and table driven routing protocol. All nodes may work as a router as well as a host to forward the network's packets to the other nodes. It is very complicated to provide comprehensive security for ad hoc networks with the desired service quality from all possible threats.

MANET is vulnerable to various kinds of attacks which include active route interfering and denial of service such as black hole, worm hole, rushing attack etc. We considered black hole attacks that appear in MANETs for this research. In the black hole attack, a malicious node advertises itself as having the shortest route to the other nodes. Source node sends the data packets to the malicious node assuming that it is a valid node in the desired path. As a consequence, the network's traffic is absorbed. In addition, nodes are trying constantly to find a route in the network for the destination, which makes the node consume its battery energy in addition to losing packets.

In this thesis, we offer a secure overlay network based approach against the black hole attack in mobile ad hoc networks. We simulated the Black hole attack in mobile ad hoc networks using AODV protocol with some changes that were made in order to fulfill our technique and then evaluated attacks' damages with and without our approach in the network. We also introduce a Markov chain model for black hole attacks in mobile ad hoc network. The results of our model were compared with p-persistent CSMA/CD, slotted ALOHA, pure ALOHA. The performance of our protocol is close to p-persistent CSMA/CD as it is going to be illustrated in the following chapters.

1.2 Objectives and Methodology

The main objective of this work is to design a protocol that can be used to prevent black hole attacks and improve the performance of the mobile ad hoc networks in terms of throughput, packet delivery ratio, normalized routing load, end to end delay and energy consumption. The proposed model is validated by simulation using the network simulator NS2 version 2.35 [25]. The network Simulator 2.35 consists of the collection of all network protocols to simulate lots of existing networks typology

and protocols. However, NS-2.35 does not have any modules to simulate malicious protocols. Thus, we simulated the Black-hole attack by adding some changes into the existing AODV protocol. The protocol is detailed in the following chapters in order to describe important network characteristics that have a significant impact on performance. It is also analyzed using a Markov chains model. We used Matlab software for simulating the analytic model.

1.3 Thesis Organization

The thesis is organized into seven chapters including this introduction. Chapter 2 provides descriptions of Ad hoc On-Demand Distance Vector Routing Protocol and the black-hole attack. Chapter 3 introduces the proposed protocol including some related work in the same area of this research. Chapter 4 shows an overview of network simulation NS-2.35. Chapter 5 details the simulation results of the model. Chapter 6 illustrates an analytic model for the black hole attacks in MANETs using Markov chains. Finally, chapter 7 summarizes the major points which discussed in the thesis.

Chapter 2

Black Hole Attack in AODV Routing Protocol

This chapter provides overviews of mobile ad hoc networks (MANET) and ad hoc On-Demand Distance Vector Routing Protocol (AODV) and explains its needed modifications in NS2. Then, it introduces some details about Black-hole attacks.

2.1 AODV Routing Protocol

2.1.1 An Overview of MANETs

Traditional wired networks are relatively more secure compared to wireless networks. Mobile ad hoc network integrates with multiple wireless systems such as wireless local area network (WLAN), wireless personal area network (WPAN), and wireless metropolitan area network (MAN), in order to improve its performance. It also provides communication between various devices (nodes) using a shared wireless channel. However, unlike more conventional wireless networks, nodes in ad hoc networks communicate without the assistance of a fixed network infrastructure. It uses collab-

orative store-and-forward strategy to provide connectivity beyond transmit/receive range. Nodes within one another's radio range can communicate through wireless links, and, often these nodes are mobile, they dynamically change their locations. This type of networks is suited for situations where rapid network coverage is required or it is highly costly to deploy and manage a network infrastructure.

Mobile ad hoc network is an autonomous collection of devices that communicate with each other over wireless. This type of network is a standalone network and its functionality is established through node cooperation. Nodes which lie within each other's transmission range can communicate directly, and, they can dynamically discover each other. The network path is an open peer to peer connection between the nodes over a common frequency band, there is no fixed infrastructure, as well as, the wireless medium may be shared. The bandwidth could be limited and stringent resource constraints. MANET's features make this network vulnerable to many attacks, thus, securing communications between mobile nodes in a hostile environment is important to prevent malicious nodes from participating in a connection's path.

This type of networks are vulnerable to threats because the wireless medium is insecure, and, the domain of attacks is transient in nature as are the wireless networks themselves. Thus, it needs efficient routing protocol and design particular detection and prevention techniques in order to secure this network. For the importance of securing such a network, there has been much work in intrusion detection systems for traditional wired networks, but, it is a bit different for Ad hoc networks. The key reason is the differences of architectural features, most notably the lack of a fixed infrastructure. The lack of certified or trusted nodes and lack of centralized audit points, such as routers, switches, and gateways, makes it difficult to collect audit data for the entire network. Intrusion detection systems, which monitor system activities and detects intrusions, must work with localized partial information. An IDS

is generally used to complement other security mechanisms, and, it is a complex and difficult task due to ad hoc network's features.

2.1.2 An Overview of AODV Routing Protocol

Ad-hoc On-Demand Distance Vector Routing Protocol (AODV) [29] is a routing protocol used for ad hoc networks in order to find a path to the destination node. It uses an on-demand approach for finding routes, where, a route is established only when it is required by a source node for transmitting data packets. All mobile nodes cooperate together using the routing control messages to set the shortest path to the destination. AODV routing protocol has many features compared to the other routing protocols. AODV makes sure the route to the destination does not lead to a loop and it is the shortest path. It uses a destination sequence number for each route entry. The destination sequence number is generated by the destination when a connection is requested. Using the destination sequence number ensures no loops in the route. It offers low network bandwidth utilization with small size control messages, quick adaptation to dynamic network conditions, low processing and memory overhead.

The control messages are used for establishing a path to the destination, sent using UDP/IP protocols are Route Requests (RREQs), Route Replay (RREPs), and Route Errors (RERRs). Header information of these control messages are explained in [12]. When the source aims to make a connection with the destination, it broadcasts an RREQ message. The RREQ message is propagated from the source, received by intermediate nodes, which are the neighbors of the source node. The intermediate nodes broadcast the RREQ message to their neighbors. This process goes on until the packet is received by destination or an intermediate node that has a fresh route entry for the destination. Fresh route indicates that the intermediate node has a valid route to destination formed a period of time ago and it is lower than the threshold. During

the RREQ packet travels through the network, every intermediate node increases the hop count by one. If more than one RREQ message with the same RREQ ID are received, the node will discard the newly received RREQs, controlling the ID field of the RREQ message. When the destination node or intermediate node that has fresh route to the destination receive the RREQ message, they create an RREP message and also update their routing tables with the last hop count and the sequence number of the destination node. Then, the RREP message is sent to the source node. While the RREQ and the RREP messages are forwarded by intermediate nodes, intermediate nodes update their routing tables and save this route entry for 3 seconds, which is the ACTIVE_ROUTE_TIMEOUT constant value of AODV protocol. Thus, the node knows over which neighbor to reach at the destination. All the default constant values of the AODV protocol are listed in appendix of RFC 3561 [12].

The destination sequence numbers, which are applied to find the latest route to the destination, serve as time stamps and allow nodes to compare how fresh their information on the other node is. Higher sequence number means more precise information and whichever node sends the highest sequence number, its information is considered and route is established over this node by the other nodes. A node increases its own sequence number when it sends any type of routing control message.

2.1.3 AODV Routing Protocol Needed Modifications in NS2

This section explains how AODV routing protocol in NS-2 [13] is modified. The version considered is NS-2.35. The files under consideration are aodv.cc, and aodv.h, which can be found in AODV folder in the NS-2 base directory.

In the main implementation files; aodv.cc and aodv.h, we enabled broadcasting of Hello packets since it is disabled by default, and, we added some statements in order to create black hole nodes By adding the following script into aodv.cc file:

else if (rt && $blackhole == 1$)

assert($rq->rq\_dst == rt->rt\_dst$);

sendReverse($rq->rq\_src$);

$rt->pc\_insert(rt0->rt\_nexthop)$;

$rt0->pc\_insert(rt->rt\_nexthop)$;

Packet::free(p);

These malicious nodes are introduced to drop packets instead of forwarding to next hop.

2.2 Black Hole Attack

In this attack, a malicious node is intent to hinder the path finding process and intercept data packets being sent to the destination node by sending a forged route reply packet (RREP) to a source node that initiates the route discovery in order to include itself in the route.

The malicious node launches this attack by advertising the shortest path and highest destination sequence number to the node that starts the route discovery. Therefore, the black hole attack has two properties. First, the node exploits the mobile ad hoc routing protocol, such as AODV, to advertise itself as having a valid route to a destination node, even though the route is false, with the intention of intercepting packets. Second, the attacker will not forward packets [4].

There are different types of this attack according to AODV routing protocol, firstly, from the source perspective, secondly, from the number of malicious node perspective. The types of black hole attack are shown in Fig. 2.1.

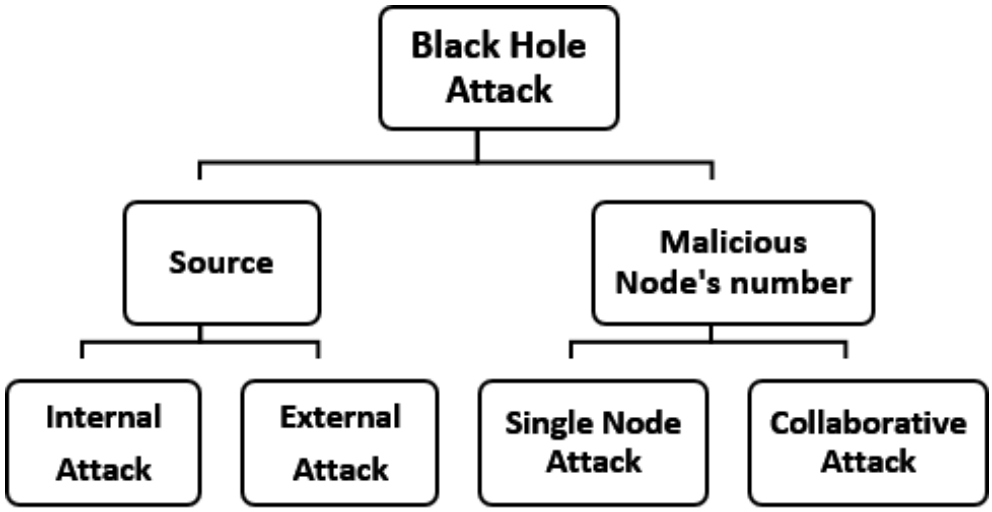

Figure 2.1: Types of Black Hole Attack

2.2.1 Internal Black Hole Attack:

The malicious node makes itself an active data route element by inserting itself in the route between source and destination nodes. Then, it becomes capable of conducting internal attack in network, which is more severe than the external one [5].

2.2.2 External Black Hole Attack

black hole nodes physically remain outside the network and deny access to network traffic or creating congestion in the network by sending bogus packets to denial of service in order to disrupt the performance of the whole network. Thus, this node will not be a part of the connection's path. This black hole node sends a RREP including spoofed destination address field to the nearest available node. This can also be sent directly to the data source node if the route is available. The source node will update its routing table according to the new information received in the route reply. However, it can become an internal attack by controlling an internal malicious node [5].

2.2.3 Single Black Hole Attack:

There is only one malicious node in a zone and the other nodes will be authorized nodes in this attack [9]. The AODV route discovery mechanism is based on Route Request (RREQ)/RREP messages [4]. Source node sends the RREQ message to the close nodes. Either the destination or an intermediate node sends RREP. The RREP is received by source node which is accepted and all further RREPs are discarded. The malicious node uses this feature of AODV and sends RREP without checking its routing table. In this way, a route through black hole node is set up and the malicious node will not forward the data packets. In Fig. 2.2, Node B is malicious. The source node S will try to send a packet to the destination node D and because S declares itself near to the black hole node the packet is going to be send through B. The route is established through this black hole node which consumes all the packets without forwarding them. The black dashed represents the route request packet (RREQ) and the route reply packet (RREP) meanwhile the red solid line represent the final route.

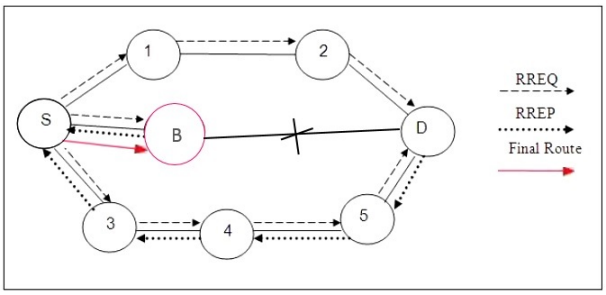

Figure 2.2: Single Black Hole Attack

2.2.4 Collaborative Black Hole Attack:

The network has more than one malicious node act as a group. It is also known as Black Hole Attack with malicious nodes [7]. When multiple black hole nodes are acting in coordination with each other, the first black hole node refers to one of its team mates as the next hop [4]. In Fig. 2.3, Nodes B and C are two black hole nodes,

S is the source node, and, node D is the destination node. In this attack, B refers to C as the next hop as same as node C refers to B as the next hop, as depicted. Once the packet is sent through either C or B, as expected since the source node is nearby them, the packet will not be forwarded. Thus, this attack affects the information of the routing tables when more than one malicious nodes collaborate together. As an example, in Fig. 2.3, it is not necessarily node B drops the packets, it could forward it to node C and then the packet is dropped.

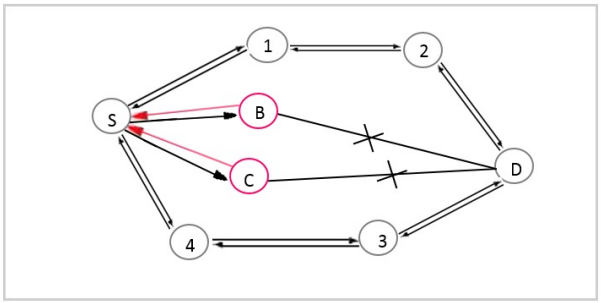

Figure 2.3: Collaborative Black Hole Attack

Recently, lots of researchers tried to face the serious risk of this attack by offering many types of solutions. By summarizing the ideas of existing solution we will find that there are three different levels to negate a black hole attack; preventative, incentive and detective-corrective [3]. The preventative defense stops the malicious nodes from participating in packet forwarding. The incentive attempts to enhance communication among the active nodes by using an economic model. Detective-corrective seeks to reveal the identity of the malicious node and to exclude it from participating in the network.

Chapter 3

Proposed Node Clustering Protocol

This chapter covers the details of our clustered-AODV-based approach. It includes a background and some related work. Followed by, the description of the clustering structure of our method. Then, the security operations provided to authenticate all nodes within the network.

3.1 Related Work

Kaur et al. [7] proposed a black hole detection tactic and alive nodes' detection methodology which was based on Artificial Neural Networks (ANN). That approach was helpful in minimizing the damage in reliable routing procedure. They used computation which was based on perceptron model to spot black hole and alive nodes. Saini and Vinod Saroha [9] proposed an algorithm which was based on fuzzy logic to detect a black hole in a MANET. They also analyzed packet loss rate, packet loss, packet delay and bit rate. Das et al. [4] proposed an algorithm which focused on analysis and improves the security of AODV protocol. Their algorithm can detect and remove Black hole nodes from MANET at the beginning of a connection.

Their paper also provided a practical study which shows effects of black hole attack. Dokurer et al. [2] presented a solution to conquer black hole which enhanced the network performance by 19% in the presence of black hole. They introduced a new protocol called BLACKHOLEAODV. Then, they implemented a new protocol called IDSAODV. They compare the results with and without black hole in the network. Gerhards-Padilla et al. [1] presented a Topology Graph based Anomaly Detection to detect Black hole Attack in Tactical MANET. To obtain the network topology information, they used well established techniques.

3.2 The Proposed Protocol

The focus of this work is providing security to an existing MANET systems using the data-link layer in order to prevent the black-hole attacks. Additionally, offering a technique which is presented by an algorithm for avoiding the black hole attacks from damaging communications in such a network. In this proposed scheme the considered protocol is ad hoc On-Demand Distance Vector (AODV); which is an on-demand routing protocol that discovers a route only when there is demand from mobile node. In this scheme, leaders are assumed to be all trusted. Grouping algorithm is not periodic which reduces updates, computation and communication cost in system. And, also, the proposed model, in Fig 3.1, uses assigned trust value and shared secret key technologies to raise the security level and make sure that all nodes are authenticated and trusted.

In the proposed algorithm the black-hole attack is prevented under the following assumptions:

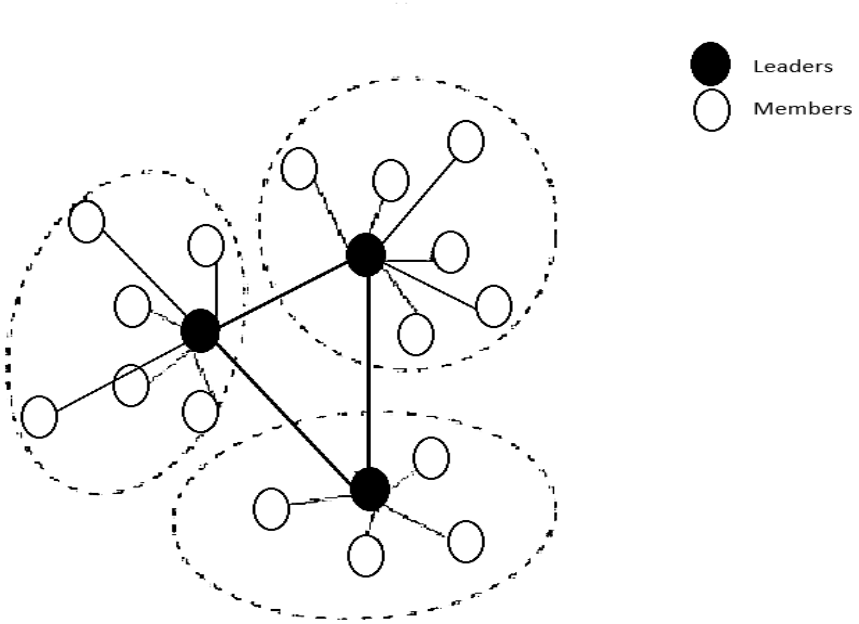

Figure 3.1: Proposed Model

1. Hierarchy: Network nodes are divided into two categories which are fixed and normal. Normal nodes will get a unique signature and an information table, both of them are from its leader. Before leaders certify a node, the leader has to make sure that node is authenticated by applying the authentication protocol. (Section 4.3)

2. Restriction: The dual transmit range of leader nodes is larger than the normal nodes. The reason is that any communication between normal nodes is possible only through leaders thus it is impossible for a malicious node to intercept a connection path with the source node.

3. Routing Protocol: The leader node controls the routing table and also the details of all its group of normal nodes. It also maintains details about other leaders and their addresses. Leader node's address is already involved in routing and it has stored in every packet for verification by the other leaders.

4. Registration: Once a certain node joins a group, it is going to maintain a signature and a table which contains information about its leader node's address and the common identifier generated by its leader node.

5. Communication: If a source node needs to deliver packets, it sends Route Request message to the group leader node, the leader uses its common identifier to verify the packets. And then, it checks if the destination node is a member in the same group in order to send all packets directly. If not, then, the leader sends Route Request message to all other leaders. Then, they will use their common identifier to verify the destination node then do the packets transferring.

6. Verification: Destination node checks if the packet has been transmitted from its leader node or from any other node using the identifier, after verification process is over it accepts the packet.

The communication between nodes in the same group is done by using the same transferring route, destination node sends RREP to the source node (Fig. 3.2).

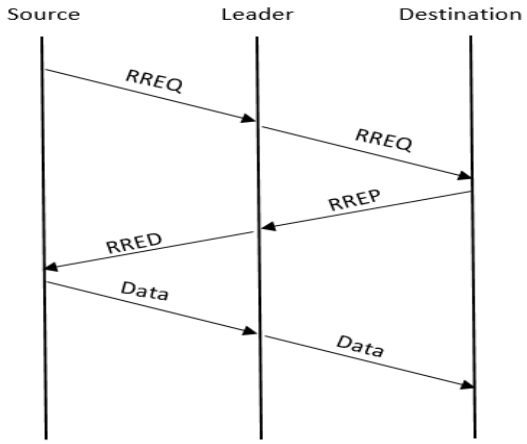

Figure 3.2: RREP Packet from Destination to Source

If any verified node moves from its group to another group, it will not subjugate to the registration process as it is already registered in the network.

3.3 Nodes Authentication Methods

3.3.1 Nodes Trust Value By Leaders

We have used Watchdog technique [15] where a node assigns trust values to its neighbors while it sends packets through them. Each leader node has a neighbor table which stores its neighbor's ID, expiration time, status and how many times a hello message was lost. Simply, suppose we have a node #1 sends a packet to another node #2 for forwarding to some next hop node #3. Node #1 can listen to the traffic of node #2 due to the broadcasting nature of wireless networks. Node #1 is often able to tell weather node #2 transmits the packet or not. By this technique, node #1 is eligible to track node #2's behavior if it drops any packet that was intended to forward to node #3 or not. Thus, node #1, which is the leader in our case, assigns a trust value to node #2 as the ratio of no. of packets forwarded by node #2 to the no. of packets received by node #2 from node #1. Generally, the trust assigned by node #i to node #j, declared as V_{ij} , is given by

$$V_{ij} = \frac{\text{No. of forwarded Packets of node \#j}}{\text{No. of received Packets of node \#j from node \#i}}$$

This scheme is a semi-distributed as the leader assigns to and maintains trust values of its neighbors only. It does not have the trust values of the nodes even two hops apart as long as they do not become its immediate neighbors.

3.3.2 Key Management

All group leaders in network are assigned unique id's. Each group leader has a public/private key pair and a secure hash function. Since black hole attacks are not targeting the traffic, we do not consider any encryption or decryption techniques

for data traffic. We define two types of keys in this network: a symmetric key which shared between group leaders and their member nodes. And, the other key is shared by all group leaders in this network. Group leaders generate key (k) which is shared between them and their members at the time when a node joins the group. K is the function of NODE_ID and a secret randomly generated number (R) by group leader.

$$K \ = \ f \ (\ NODE\_ID \ , \ R \ )$$

where f is a secure hash function, NODE_ID is assigned node id and R is a secret number generated by the groups leader. In addition, groups' leaders will share a key to securely communicate using Group Diffie-Hellman key agreement protocol [16]. This protocol helps to securely exchange private keys over a public channel whereas the shared secret key is not known for public. Let P be a large prime number, and let g be a primitive element. Both P and g are publicly known. In the Diffie-Hellman key agreement protocol, two parties leader (1) and leader (2) exchange public keys X and Y, where X,Y $\in R[1, P-1]$ are their private keys, and then compute the shared secret S . A session key is then derived from S usually by hashing. The key can be formed by broadcasting key related messages for all leaders in several steps as follows:

1. A leader (i) selects its private key (S_i), computes its public key by g^{S_i} .
2. The leader (i) broadcasts the first round broadcast by sending $(g^{S_i}, Leader(i))$.
3. The leader then selects another secret number R_i.
4. After the first round broadcast messages are received, the leader computes $g^{S_i}g^{S_j}$ for all $j \neq i$, and encrypts R_i with each of g^{S_i} g^{S_j} respectively, then bundles the following parameters into one packet.

$$((R_{ig^{S_i}g^{S_0}}, Leader(0)),\ (R_{ig^{S_i}g^{S_1}}, Leader(1)),\ ....,\ (R_{ig^{S_i}g^{S_i}}, Leader(i)),\ ....,$$
$$(R_{ig^{S_i}g^{S_{n-1}}}, Leader(n-1)))$$

5. the leader then broadcasts the packet, this time is the second round broadcasting.
6. After receiving the second round broadcast messages, the leader is able now to compute the group key as $K_G = f(R_0, , R_n 1)$ where f is a predefined function which is an XOR function.

On the other hand, if a new node wants to join a group, it has to send a request to the group's leader. This request may be captured by a malicious node showing as group leader to the new node. Thus, it is important for both nodes to authenticate each other. The new node and the group leader can authenticate each other using challenge-response protocol. In this protocol, a new node has to send a challenge to the group's leader and this leader has to provide a valid response to prove its sincerity according to the next steps.

1. The leader chooses two large prime numbers (P) and (Q) and calculates N using the following equation. N is publicly announced in the group.

$$N\ =\ P \times\ Q$$

2. The leader selects a random secret number (S) and calculates Y that is publicly announced within the group using the following equation.

$$Y\ =\ S^2\ mod\ N\ where\ (\ 1\ <\ S\ <\ N\ )$$

3. The group's leader selects R which is a random number such that $1<R<N$ and then calculates X by the equation:

$$X = R^2 \ mod \ N$$

4. The leader sends the parameters N, Y, and X to the new node. After that, the new node sends a challenge (G) to group leader. Group leader calculates Z by the following equation, then, sends it to the node.

$$Z = RS^G \ mod \ N$$

5. The new node calculates M

$$M = XY^G \ mod \ N$$

6. The new node matches M value with Z^2. If both values are the same, group leader is successfully authenticated.

Now, the new node can join the group and share a key with group leader securely after successful mutual authentication. Group's leader generates NODE_ID and a symmetric key which generated by the previously mentioned function (f). Then, the leader updates group members list and sends it to the members of the group.

All communications are done through groups' leaders. If a node receives RREQ from other node, this RREQ will be discarded. If node #1 wants to send a RREQ to another node #2 which is out of its group, node #1 sends its request to its leader which will send a request RREQ to destination's group leader. The new leader then sends RREQ message to destination node maintaining a route from itself to destination. Finally, destination node replies back to its group leader by sending route reply RREP

message towards source node. Verification of nodes and message integrity is using digital signatures and hash code of messages. In our proposed scheme, RREQ carries hash code of parameters that are parts of RREQ so any kind of modification in RREQ would change this hash code. RREQ has a destination sequence number to avoid loop and it also has Digital Signature of all nodes in route from source to destination. RREQ packet will not go in infinite loop because lifetime message will be decreased by 1 on every broadcast.

$$RREQ(SourceNode,\ DestinationNode,\ SequenceNumber,\ <Route>,$$
$$Node's\ Digital\ Signature,\ Node's\ Hash\ Code)$$

$$Node's\ Hash\ Code = H(SourceNode, DestinationNode, SequenceNumber)$$

3.4 Pseudo-code of the Proposed Protocol

After explaining our protocol, we would like to introduce the Algorithm 1 which shows the mechanism of assigning nodes to the leaders in the network. All these steps are done according to the previously mentioned assumptions.

From Algorithm 1, the chance for a malicious node to get involved in a route between the source and the destination nodes is very low, as it is not included in the group of the leaders. Moreover, the throughput, which represents the average number of message transmitted in a noted time, will be improved. As it is known, malicious nodes in black hole attacks affect network throughput dramatically. By using the mentioned technique, the black hole attack is easily prevented during the route discovery process between the source and destination node. Thus, the routing communication process will be more efficient between nodes and the packets are safe to suitably reach the destination node.

Algorithm 1 The Pseudo-code of Our Protocol

Initialize Leader nodes and Normal nodes
Assign node to group #Network Establishment
for i=0 To N **do**
  read current
  Nodes Maintaining Trust Values;
  if (Nodes within the range of LEADER_Node) **then**
    Transmit common identifier;
  else
    The node is related to another group;
  end if
end for
Assign node to a leader: #For verifying a normal node, the node has to send its details to the group's leader
for each node n **do**
  LEADER_Node will ask the node for its details
  if (The node sent its details to the LEADER_Node) **then**
    The Node is verified with its key by the LEADER_Node;
    The Node is informed with routing table of authenticated nodes by the LEADER_Node;
    Node = Normal Authenticated Node;
  else
    The Node is not verified;
    Node = Unauthenticated Node;
  end if
end for
Communication:
Source node forwards RREQ to the LED_Node;
if (destination and source nodes are within the same group) **then**
  Forwards RREQ to Destination node;
else
  Current LEADER_Node Forwards RREQ to other LEADER_Node;
  LEADER_Node forwards RREQ to Destination node;
end if
Verification:
Destination node checkes RREQ;
if (RREQ is from LEADER_Node) **then**
  RREQ is accepted;
else
  RREQ is discarded;
end if

Chapter 4

Network Simulator NS-2

We have used Network Simulator NS-2.35 in order to test the scenario of our scheme which includes black hole nodes in mobile ad hoc networks (MANETs). To do the simulation, we added changes into AODV protocol to include malicious nodes that drop data packets. In this chapter, we present some details about the network simulator NS-2.

4.1 An overview of Network Simulator NS-2

NS-2 [13] is a discreet event network simulator used for networking research. It is an open-source simulation tool running on Unix-like operating systems. It supports simulation of routing, multi-cast protocols and IP protocols, such as UDP and TCP over wired and wireless networks.

NS-2 is written in Object Tool Command (Otcl) and C++ languages in order to disconnect the control and data path implementations. This simulator supports a corresponding hierarchy within the Otcl interpreter as same as a class hierarchy in C++. NS-2 uses two languages because the simulator has two different tasks to do, which have different requirements[13]. C++ is used to implement protocols. Also,

generally, it is used for such cases where every packet of a flow has to be processed. Moreover, Otcl is suitable for configuration and setup. Otcl runs quite slowly, but it can be modified very easily. The ready-made C++ objects can be controlled from the Otcl level.

It is important to understand how Otcl works in order to get correct results. Briefly, in tcl, values can be assigned to variables and these values can be further used, new files can be opened for reading by using the command open, new procedures can be defined with the proc command, and a sub-process can be created with the exec command. Exec command is also used when the one wants to call a tcl-script from within another tcl-script[13].

There are some steps that need to be done in order to be able to simulate any scenario. A new simulator object must be created at the beginning of the script. The simulator object has member functions that enable creating nodes and links, connecting agents etc. All these functions can be found from the class Simulator in NS-2. Then, the topology should be set up and created in order to run the simulation. The topology consists of links and groups of nodes. After the simulation, the data has to be collected in trace files to be able to calculate the results. The traces enable recording of packets whenever an event such as packet drop or arrival occurs in a queue or a link. After topology is created, agents are configured etc., the start and stop of the simulation and other events have to be scheduled with the commands run and finish [25].

The member function of the Simulator class, called node creates nodes. If the node is not a router but an end system, traffic agents such as TCP, and traffic sources such as FTP, must be set up that means sources need to be attached to the agents and the agents to the nodes, respectively. In NS-2 [13], the most common agents used are UDP and TCP agents and the most common applications and traffic

sources provided by ns2 are Application/FTP, Application/Traffic/CBR, Application/Traffic/Exponential, and Application/Traffic/Trace.

Links are required to complete the topology. In NS-2 [13], the output queue of a node is implemented as part of the link, so when creating links the user also has to define the queue-type. There are a duplex-link and simplex links in NS-2. In the link [13], the packet is entered the queue. Then, it is either dropped, passed to the Null Agent and freed there, or dequeued and passed to the Delay object which simulates the link delay. Finally, the time to live (TTL) value is calculated and updated. In addition, traffic sinks are important as well. An UDP sink is defined in the class Agent/Null and a TCP sink is defined in the class Agent/TCPSink. TCP Sinks are a subclass of Agents that implement a receiver for TCP packets. The simulator only implements one-way TCP connections. And, Null sinks are a subclass of Agents that implement a traffic sink. If the information flows are about to be ended without processing, the TCP and UDP sources have to be connected with traffic sinks [25].

If we consider NS-2 as a black box with inputs and outputs, the OTcl scripts are its inputs and its outputs are trace files. Trace files are two types; namtrace and trace. These traces can be used for network animations using network animator (NAM) and/or analysis using Xgraph/GNUplot. NAM is a Tcl/TK based animation tool, which supports topology layout, packet level animation, and various data inspection tools. Using NAM, the network topology and the packet movement are easy to follow [13].

4.2 Installation

This tool is a free simulation tool, which would be found in [25]. It is supported on various platforms including UNIX , Windows, and Mac systems.

NS2 source codes are distributed in two forms: the all-in-one suite, which we have used, and the component-wise. With the all-in-one package, users get all the required components along with some optional components. This is basically a recommended choice for the beginners. This package provides an install script which configures the NS2 environment and creates NS2 executable file using the make utility.

The current all-in-one suite consists of the following main components [25]:

- NS release 2.35.
- Tcl/Tk release 8.5.8.
- OTcl release 1.14.
- TclCL release 1.20.

and the following are the optional components [25]:

- NAM release 1.15: NAM is an animation tool for viewing network simulation traces and packet traces.
- Zlib version 1.2.3: This is the required library for NAM.
- Xgraph version 12.2: This is a data plotter with interactive buttons for zooming, printing, and selecting display options.

In addition, the component-wise approach is used to separately download the above pieces and install them individually. This option saves considerable amount of downloading time and memory space. However, it could be completed for the beginners and is therefore recommended only for experienced users.

4.3 Simulation Steps

The general simulation steps can be tailored to fit with the NS2 framework. In this section, we introduce the main simulation steps for NS2.

4.3.1 Step 1: Simulation Design

To simulate a network, the user has to design the simulation. In this step, the users should determine all the needed details for the simulation purposes, network configuration, assumptions such as the network routing protocol assumptions, the performance measures, and the type of expected results.

4.3.2 Step 2: Implementing The Simulation's Design

After designing the simulation, the user should implement the design that he planned for his network. Implementing the design consists of two phases:

- Network Configuration Phase:

  Network components, such as node, TCP and UDP, are created and configured in this phase according to the simulation design. Also, the events such as data transfer are scheduled to start at a defined time.

- Simulation Phase:

  This phase starts the simulation which was configured in the Network Configuration Phase. It maintains the simulation clock and executes events chronologically. This phase usually runs until the simulation clock reaches a threshold value specified in the previous phase.

As mentioned before, it is convenient to define a simulation scenario in a Tcl scripting file and feed the file as an input argument of an NS2 invocation.

4.3.3 Step 3: Post-simulation Processing

After implementing the design a trace file, as mentioned previously, will be generated. The trace file records all the details of packets passing through network checkpoints. Using the trace file, the performance of the network can be calculated.

The main tasks in these steps include verifying the integrity of the program and evaluating the performance of the simulated network.

Chapter 5

Simulation and Results

In this Chapter, we present our simulation and its results. The performance analysis of the proposed approach is done as mentioned previously by NS-2.35. NS-2.35 was installed on Windows 8.1 operating system using Virtual Box with Fedora 21 environment.

5.1 Simulation Parameters and Setup

The simulation networks considered here consists of 10 to 30 nodes including 2 to 6 malicious nodes deployed in a field of 600 x 600 square meters. Some nodes are set in promiscuous mode. Watchdog [22] is also implemented to calculate node's trust value as it is mentioned in section 4.3.1. Each node is randomly placed in the area. After a defined pause time, a path is set up between predefined pairs of source and destination. Node's movement is set up using a random mobility model. After each 15 seconds of simulation, the nodes are moved to new positions randomly. The transmission range was set at 250m and the data rate is 2 Mbps. The user traffic model is Constant Bit Rate (CBR) traffic. CBR packet size is chosen to be 512 bytes long. We used the energy model as used by [14]. Each simulation is run for 200 seconds. Five independent simulations are taken for a particular scenario combining the normal

nodes and malicious nodes. The results are averaged. The specification of network is given in Table 1 meanwhile Table 2 summarize our simulation parameters.

Table 5.1: Network Specifications

Parameters	Value
Channel Type	Channel/WirelessChannel
MAC Layer	802.11
Network Interface	Physical/Wirlessphy
Radio Propagation Model	Propagation/TwoRayGround
Mobility Model	Random Waypoint
Traffic Model	CBR
Routing Protocol	AODV

Table 5.2: Simulation Parameters

Parameters	Value	Unit
Simulation Area	600 x 600	Square Meters
Number of Nodes	10 to 30	Nodes
Node Speed	15	Meter/Second
Simulation Time	200	S
Packet Size	512	bytes
Data Rate	2	Megabits/Second
Transmission Range$_{Leader}$	250	Meter
Transmission Range$_{Member}$	50	Meter

5.2 Measured Metrics

We have focused on Five aspects in order to evaluate our scheme. First, throughput which is the number of packets successfully delivered from the source to destination. Second, end to end delay which is the average time taken by the packet to be delivered

to the destination node. Third, packet delivery ratio which is the proportion of the number of delivered data packet to the destination. Fourth, normalized routing load which is the number of routing packets per data packets delivered at the destination. Finally, energy consumption which represents how much energy is consumed during our simulation. Then, our Clustered AODV based scheme (CAODV) was compared to the original AODV.

5.2.1 Throughput

Throughput (Th) is the average rate of successful data packets received at destination. It is calculated by taking the number of bytes received by the sink. Received bytes number is multiplied by 8, and divided by sample time.

$$Th = \frac{RB \times 8}{T}$$

where RB is the received bytes and T is the time.

5.2.2 Packet End-to-End Delay

Packets are transmitting from source node to destination node. The difference between send times and received times is and calculated. All delays due to route discovery, propagation, transfer time and queuing are included in the delay metric. It is calculated by subtracting the initial transmitting time of packets from the last packet received time (lastPktTime_), then dividing the result by the total number of packets sent that sampling interval.

$$ETE\ Delay = \frac{LPR_{Time} - PS_{Time}}{TPR}$$

Where LPR_{Time} is the time of last packet received, PS_{Time} is the initial time of

transmitting the packet and TPR is the total number of received packets.

5.2.3 Packet Delivery Ratio

Packet Delivery Ratio (PDR) is defined as the ratio of the number of packets sent by the source node and received by the destination node.

$$PDR = \frac{TPR}{TPS}$$

Where TPR is the total number of received packets and TPS is total number of sent packets.

5.2.4 Normalized Routing Load

Normalized Routing Load (NRL) is the ratio of the total number of sent routing packet including forwarded routing packets, which are generated by routing protocol during simulation, per data packet. It is calculated by dividing the total number of transmitted routing packets over the total number of data packets received.

$$NRL = \frac{TTP}{TPR}$$

Where TTP is the total number of transmitted routing packets and TPR is the total number of received packets.

5.2.5 Energy Consumption

The network nodes includes Energy/Model in order to be able to calculate power consumption [14]. Energy consumption (EC) is the consumed power by nodes during the simulation. It is calculated by subtracting the value of residual energy of nodes

at the end of the simulation from the total energy.

$$EC = TE_{Node} - RE_{Node}$$

Where TE_{Node} is the total node's energy and RE_{Node} is the remaining energy of the node.

5.3 Results and Analysis

After executing the tcl scripts, we get a trace files which have extension (.tr) and a NAM file which have extension (.nam). Then, we have used AWK scripts which have extension (.awk) in order to extract the necessary information for calculating the results from trace files, and, then, plot its graphs.

5.3.1 Throughput

Figure 5.1 shows the average throughput (Th) of original AODV protocol and the clustered-AODV based protocol (CAODV) as well. The unit of throughput from our simulation is kilobyte per second (Kbps), then, we convert it to the unit packet per second (pps). X-axis in this figure represents number of nodes meanwhile Y-axis represents the packets in pps. The average throughput of CAODV is higher than the average throughput of the normal AODV as illustrated on the graph, which indicates that CAODV has a higher performance than the regular AODV.

5.3.2 End to end delay

The average end to end packet delay (EPD) for AODV and CAODV are shown in Figure 5.2. Here, X-axis represents the number of nodes and Y-axis represents time

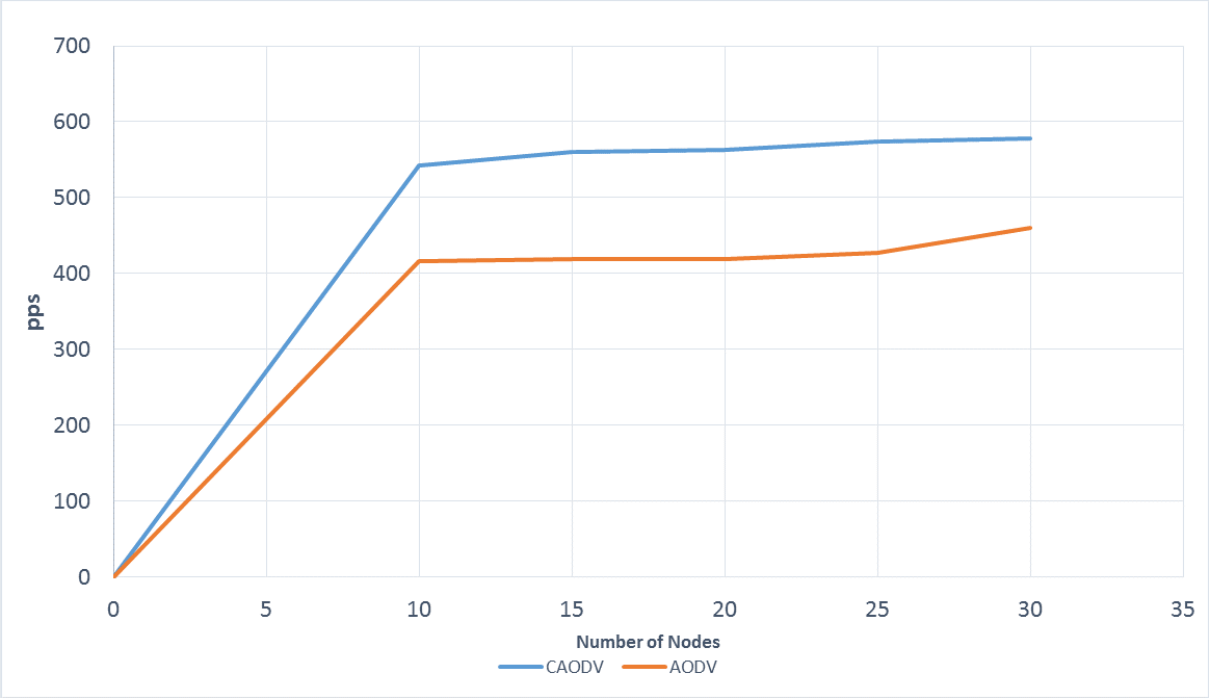

Figure 5.1: Average Throughput in pps for AODV and CAODV

delay in Ms. As it is appeared, AODV has higher end to end delay than CAODV. There are many reasons for delaying plackets but the main reason could be that the number of participating nodes in the network in AODV is more than CADOV.

5.3.3 Packet Delivery Ratio

In Figure 5.3, the average of packet delivery ratio (PDR) for both original AODV and our scheme CAODV are printed. X-axis represents the number of nodes and Y-axis represents packet delivery rates. As it is shown, the difference in the delivery ratios increases as the number of nodes increases. Additionally, AODV has lower PDR than CDAOV. The performance gained by our scheme is due to two main reasons, first, the effective use of AODV control messages in CAODV, and second, its localized route discovery.

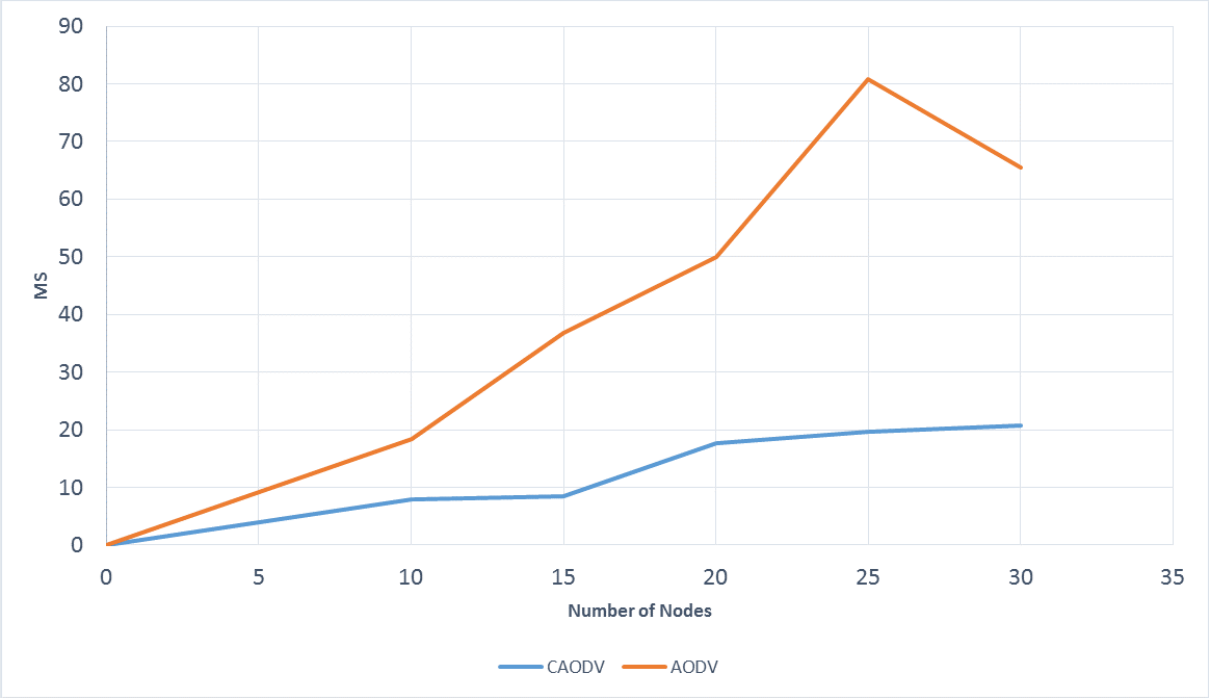

Figure 5.2: Average EPD for AODV and CAODV

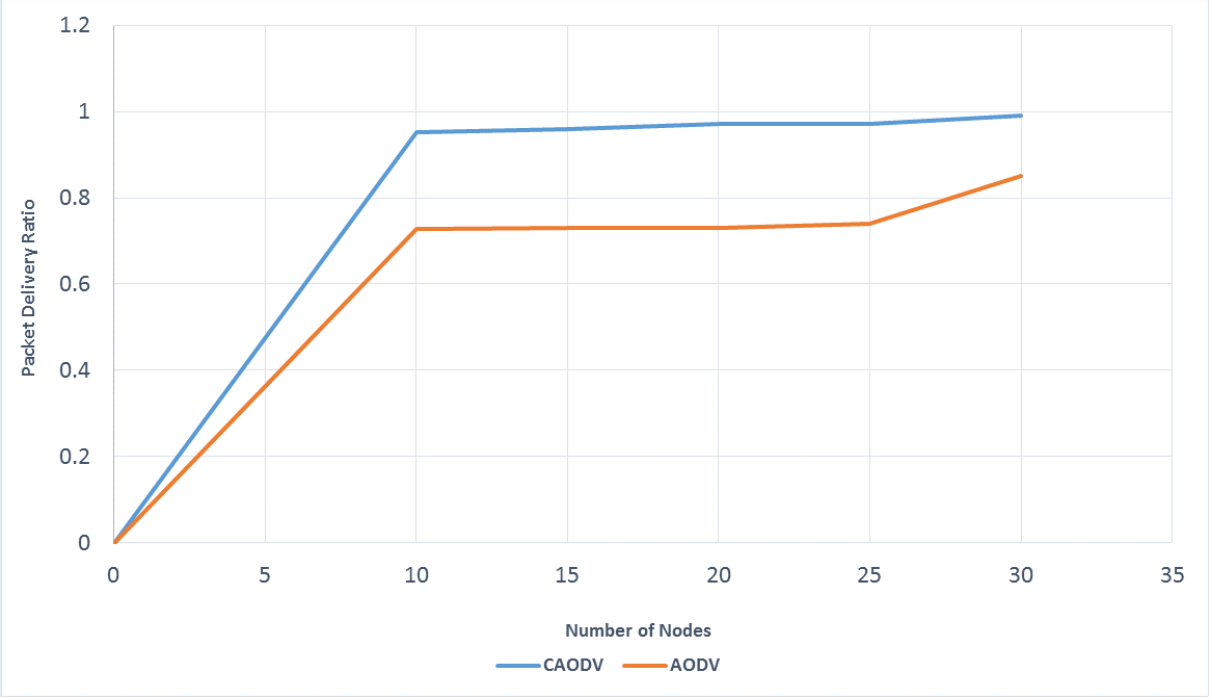

Figure 5.3: Average PDR for AODV and CAODV

5.3.4 Normalized Routing Load

Figure 5.4 represents the average normalized routing load (NRL) for AODV as well as CAODV. In this graph, X-axis identifies the number of nodes and Y-axis identifies the ration of routing load. It is illustrated that the NRL of our scheme CAODV is less than AODV, which indicates that the number of received packets in CAODV is higher than AODV, that's because CAODV generates less control traffic overhead due to its localized and distributed control traffic handling.

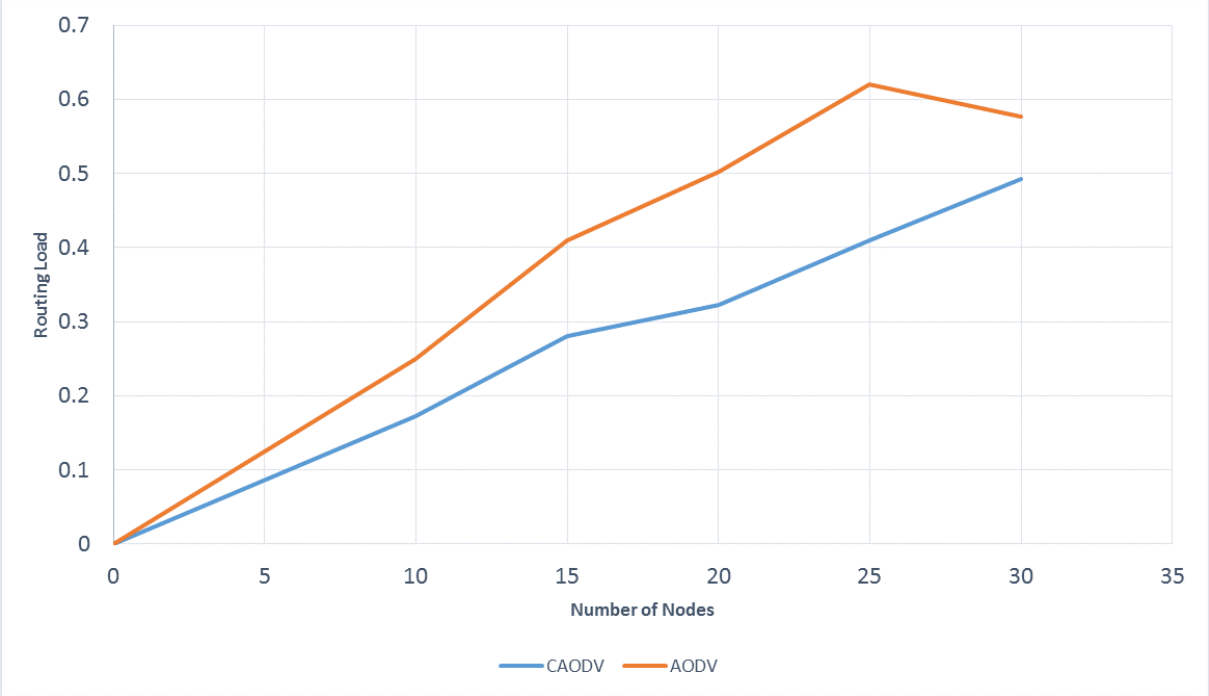

Figure 5.4: Average NRL for AODV and CAODV

5.3.5 Energy Consumption

In Figure 5.5, the average of energy consumption for AODV and CAODV. In this figure, X-axis represents energy consumption in Joules and Y-axis represents the number of nodes. The results show that nodes in CAODV consumed less energy than

Nodes in normal AODV. The reason of the energy reduction in CAODV protocol is that we eliminate the unnecessary connections between nodes.

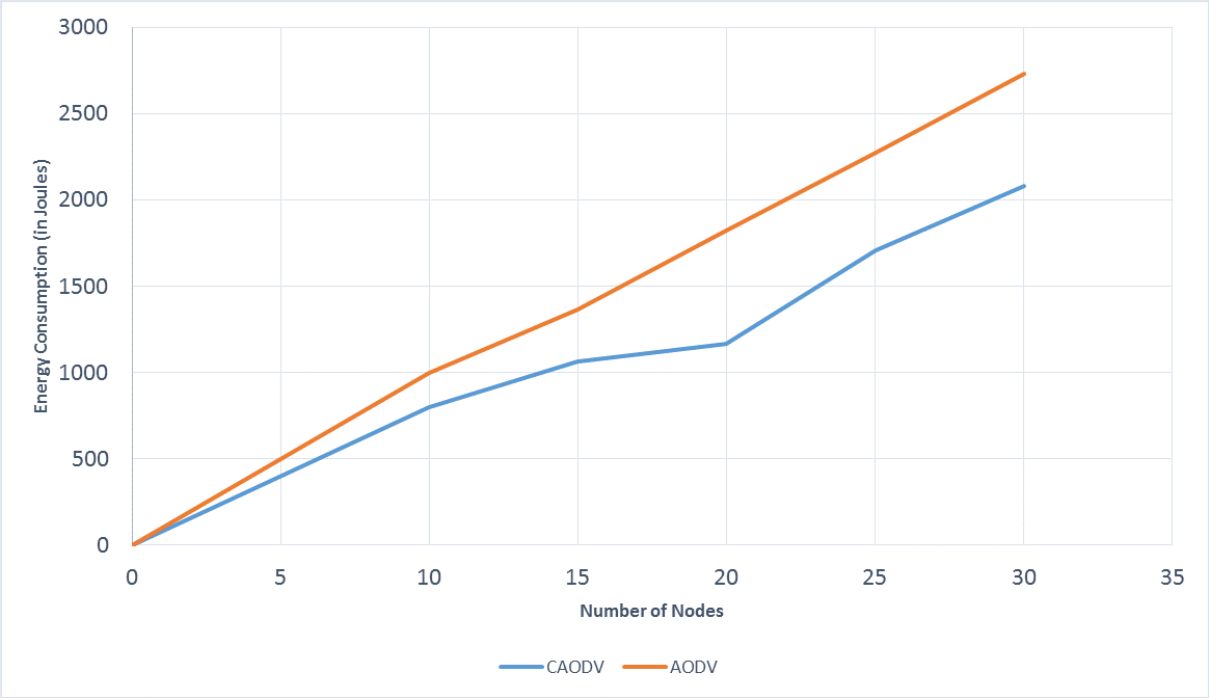

Figure 5.5: Average EC for AODV and CAODV

Chapter 6

IEEE 802.11:Markov chains model for blackhole attacks using RTS/CTS for Ad-hoc networks

In this chapter, we present a stochastic model to analyze the effects of the black hole attacks which is a well known attack in the mobile ad hoc environment. In black hole attack [18], [19], and [17], the black hole node uses its routing protocol to advertise that it has the shortest path to the destination node as we mentioned in chapter 2. Therefore, an attacker will always have the situation in replying to the route request and thus attract all the traffic on the network and intercept the data packet.

6.1 Markov chains for MANET

Markov chain is a special case of the Markov process whose development can be treated as a series of transitions between certain states. It is the most commonly used stochastic techniques for predicting the performance of various infrastructure facilities. Markov-chain models are based on the concept of probabilistic cumulative damage, which predicts changes of component condition over multiple transition. Due

to a node's communication is a time variant, thus, a better option to characterize this communication is Markov chains. This stochastic process has a limited number of states and whose transition between them is based on the probability of an event. It is a mathematical model for a process which moves step by step through various states, it can describe efficiently the characteristics of the system. In addition, the probability that the process moves from any given state to any other particular state is always the same, regardless the history of the process.

6.2 The black hole modeling scheme using RTS/CTS

The IEEE 802.11 [24] is the most widely used medium access control protocol for wireless local area networks (LANs). IEEE 802.11 wireless LAN standard is used for infrastructure as well as ad hoc networks. In IEEE 802.11 based mobile ad hoc networks (MANET), multi-cast packets are generally forwarded as one hop broadcast; mainly to reach all the multi-cast members in the neighborhood. The mobile ad hoc network does not have a central controller. Instead, each node in the network attempts to access the shared medium on its own. Multiple simultaneous but spatially separated transmissions are possible in the ad hoc network. Thus, the request to send/ clear to send (RTS/CTS) mechanism is a widely used technique for packet transmission between nodes in IEEE 802.11 Medium Access Control (MAC) protocol which helps to avoid packet collisions in order to achieve high throughput. In the RTS/CTS protocol, the source informs the destination of its intention to exchange data by issuing an RTS packet, and the destination confirms this with a CTS packet, after which the source sends the payload (DATA) packet. All other nodes that recover the RTS or CTS packets are unable to transmit during some specified time interval in order to facilitate a successful RTS-CTS-DATA cycle.

Moreover, MANETs are vulnerable for many types of attacks and the malicious

nodes use several techniques to illegally increase their throughput and capture the channel at the expense of other normal nodes [19]. In IEEE 802.11, A node is considered malicious when it deviates from the IEEE 802.11 MAC Standard [24]. It is believed that ad hoc network problems are all related in somehow to malicious nodes that effectively damage the functionality of the network [31]. Thus, there is a need to have security mechanisms to protect these networks.

We employ a set of assumptions in order to derive a simple mathematical model for the effect of the black hole attack in MANETs using RTS/CTS mechanism. The RTS/CTS mechanism is used to reduce energy consumption due to collisions. We note that the time a station requires to determining if a collision occurred is the duration of the RTS/CTS packets. This is definitely shorter than the data duration. Thus, by using the RTS/CTS mechanism hopefully the performance will be higher compared to the other protocols. The assumptions we employ are:

1. The current state of the user depends only on its immediate past history and we can model the user using Markov chain analysis.

2. The behavior of one user is considered in this protocol.

3. The states of the Markov chain represent the states of the user: idle, waiting, requesting to send, transmitting, and collided.

4. There are N equal priority users and N_b black hole nodes in the network.

5. We replace the waiting backoff window with a transmitting probability p when the channel is sensed idle.

6. The duration of one time step in the contention window is roughly taken equal to the propagation delay plus the time it takes a station to sense the presence of a carrier. This time is called the Distributed Inter-frame Spacing (DIFS).

7. The Markov chain time step is taken equal to the DIFS period.

8. The ratio of frame transmission delay to contention window delay is $n > 1$.

9. All frames have equal lengths such that a frame takes n time steps to be transmitted.

10. The transmission range of a node is R.

11. The nodes are distributed in a square area of dimension LXL.

12. Probability that an idle user receives a frame for transmission during a frame period is a.

13. Probability that a black-hole node is located within the transmission range of a node is α which is calculated by the following equation:

$$\alpha = \frac{\pi R^2}{L^2} \tag{6.1}$$

.

14. A station can have at most one message waiting for transmission.

15. Collided users employ a random backoff strategy with transmit probability γ when the channel is sensed idle.

In Fig. 6.1 [21], a station with a packet to send will first send a request to send packet (RTS) when it senses the channel is free for a minimum of DIFS time. If the RTS packet is successfully received without suffering collisions, the intended receiver will issue a clear to send packet (CTS). After this, the sender will commend to send the data and wait for an acknowledgment (ACK) for the receiver [21].

Fig. 6.2 shows the state transition diagram for the IEEE 802.11 user when our protocol is used. There are several good transmitting states because the time required for transmitting one frame is bigger than the propagation delay p. There is also only

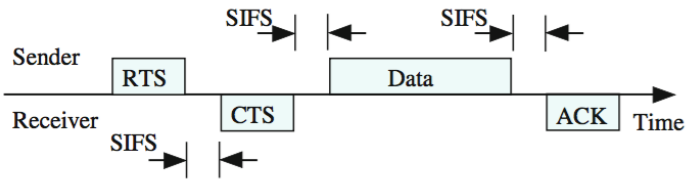

Figure 6.1: Four-way handshaking RTS/CTS (MACA) protocol for IEEE 802.11

one collided state since the user will not transmit after a collision is appeared.

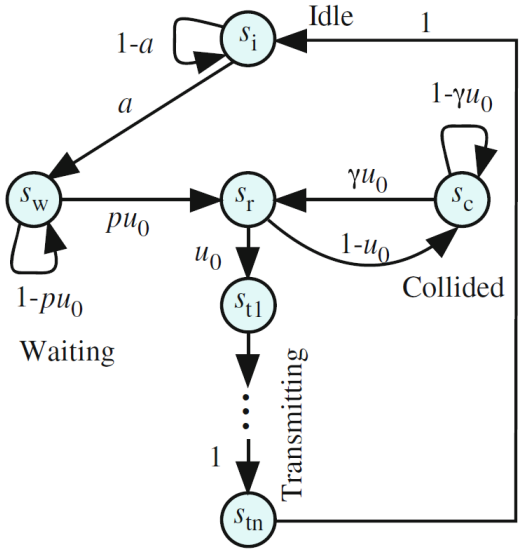

Figure 6.2: State transition diagram for the tagged user of our model

The probability u_0 in Fig. 6.2 denotes the probability that all $N-1$ users, apart from the tagged user, will not transmit when the channel is free. The probability that a user will not start transmission even when the channel is sensed free is given by:

$$P_{idle} = s_i + (1 - u_0 p)s_w + (1 - u_0\gamma)s_c \tag{6.2}$$

The probability all the other users will not start transmission is given by:

$$u_0 = [s_i + (1-p)s_w + (1-\gamma)s_c]^{N-1} \tag{6.3}$$

We organize the distribution vector at equilibrium as follows:

$$s = [s_i, s_w, s_r, s_c, s_{t_1}, s_{t_2}, ..., s_{t_n}]^t \tag{6.4}$$

The corresponding transition matrix of the channel for the case when $n = 3$ is given by:

$$P = \begin{pmatrix} a & 1-pu_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & pu_0 & 0 & \gamma u_0 & 0 & 0 & 0 \\ 0 & 0 & 1-u_0 & 1-\gamma u_0 & 0 & 0 & 0 \\ 0 & 0 & u_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{6.5}$$

At equilibrium the distribution vector is calculated by solving the following two equations

$$Ps = s \tag{6.6}$$

$$\sum s = 1 \tag{6.7}$$

The terms in P depend on the state vector components. This constitutes a highly nonlinear set of equations. The solution for s is obtained through several techniques such as iterative techniques as in the following pseudo code:

Algorithm 2 The iterative technique to obtain s

1: **Input** a, N, n, p, **and** γ.
2: **Initialize** a trial value for the state vector s:

$s_i = s_w = s_r = s_t = rand/6;$
$s_c = 1 - (si + sw + sr + st);$
$a\_mod = \frac{a}{nN};$
$abs\_error = 1;$
$error\_limit = 0.0001;$
$f = 0.1;$
$iteration = 0;$
$max\_iteration = 1000;$

The iterations is started by obtaining the probability u_0:

3: **while** $(abs\_error > error\_limit)$ AND $(iteration < max\_iteration)$ **do**
4: $p_{user\_idle} = s_i + (1-p) \times s_w + (1-\gamma) \times s_c;$
5: $u_0 = {p_{user\_idle}}^{N-1};$
6: $u_{inv} = \frac{1}{u_0};$
7: $D = 1 + \frac{a\_mod}{(u_0 \times p)} + a\_mod \times u_{inv} + \frac{1-u_0}{u_0 \times \gamma} \times a\_mod \times u_{inv} + n \times a\_mod;$
8: $D_{inv} = \frac{1}{D};$

Calculate state vector ($\vec{s}$) after the value of u_0 is substituted:

9: $P\vec{s} = \vec{s};$
10: $\sum \vec{s} = 1;$

Calculate the error:

11: $\vec{e} = \vec{s_{calc}} - \vec{s};$

Update $\vec{s}$, calculate the root mean square error, and update the iteration:

12: $\vec{s} = \vec{s} + f\vec{e};$
13: $abs\_error = abs(e_i) + abs(e_w) + abs(e_r) + abs(e_c) + abs(e_t);$
14: $iteration = iteration + 1;$
15: **end while**

6.3 Results and Performance

To obtain the performance of our mathematical model, we have calculated the single hop distance, the average number of hopes for a packet to travel from the source to the destination, and the probability for the packet to be forwarded to its destination through good nodes. In order to calculate the single hop distance between nodes, we have to find out the density of the distributed node in the network. To obtain that we calculated the node density (ρ) in units of node per unit area which is given by the following equation:

$$\rho = \frac{N}{L^2} \qquad (user/unit\ area) \tag{6.8}$$

The linear density of users (ρ_a) is given by the equation:

$$\rho_a = \frac{\sqrt{N}}{L} \tag{6.9}$$

Therefore, the average single hop distance (l_d) between users is obtained by the following equation:

$$l_d = \frac{L}{\sqrt{N}} \qquad (distance/user) \tag{6.10}$$

Fig. 6.3 shows the average single hop distance between users in this network.

The average number of hopes (h_a) is given by the following equation:

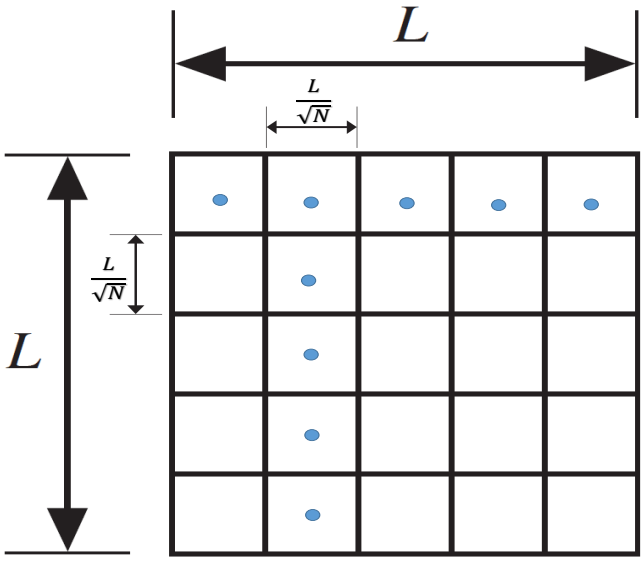

Figure 6.3: The average single hop distance (l_d) between users

$$h_a = \frac{L}{l_d} = \frac{L\sqrt{N}}{L} = \sqrt{N} \tag{6.11}$$

Additionally, the probability for a packet to be sent through good nodes is calculated through the equation:

$$P_{good} = (1-\alpha)^{N_b} \tag{6.12}$$

where P_{good} is the probability that a packet to be sent through good nodes and N_b is the number of black hole nodes in the network.

Table 6.1 illustrates the values of the model parameters that have been used in

this simulation.

Table 6.1: Simulation Parameters

Parameters	Value
N	10
N_b	2
n	20
R	250
L	600
γ	0.01
α	0.55
p	0.05

The user throughput (Th) can be calculated by the following equation:

$$\begin{aligned} Th &= nNs_t(P_{good})^{h_a} \\ &= nNs_t(1-\alpha)^{\sqrt{N}N_b} \end{aligned} \tag{6.13}$$

where P_{good} is the probability that a packet to be sent through good nodes, h_a is the average number of hops for a packet to be delivered, and N_b is the number of black hole nodes in the network.

Fig. 6.4 shows the user throughput of our model versus the average input traffic when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01. The dashed black line is the throughput of the multi-hop network, the black line is the throughput of the single hop network the blue line represents the throughput of $p-$persistent CSMA/CD, the green line shows the throughput of slotted ALOHA, and the red line is the throughput of pure ALOHA.

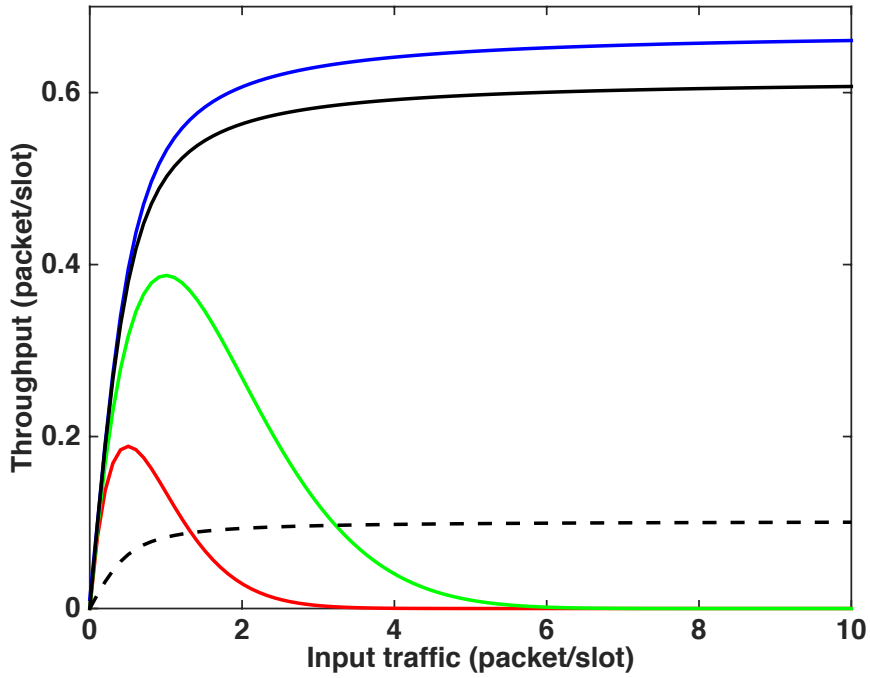

Figure 6.4: User throughput for the model versus the average input traffic when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01. The dashed black line is throughput of the multi-hop network, the black line is the throughput of the single hop network, the blue line is the throughput of p-persistent CSMA/CD, the green line is the throughput of slotted ALOHA, and the red line is the throughput of pure ALOHA.

The packet in the multi-hop networks has a higher probability to be intercepted by black hole nodes; which affect the user throughput. Another issue that may affect user throughput is that the users with a frame to transmit have to wait before accessing the channel. Additionally, changing the value of p has a little effect on the throughput as well. The throughput of both CSMA/CD and our protocol is reduced when the value of γ is increased from 0.01 to 0.05.

The user access probability is obtained by:

$$p_a = \frac{Th}{Na} \tag{6.14}$$

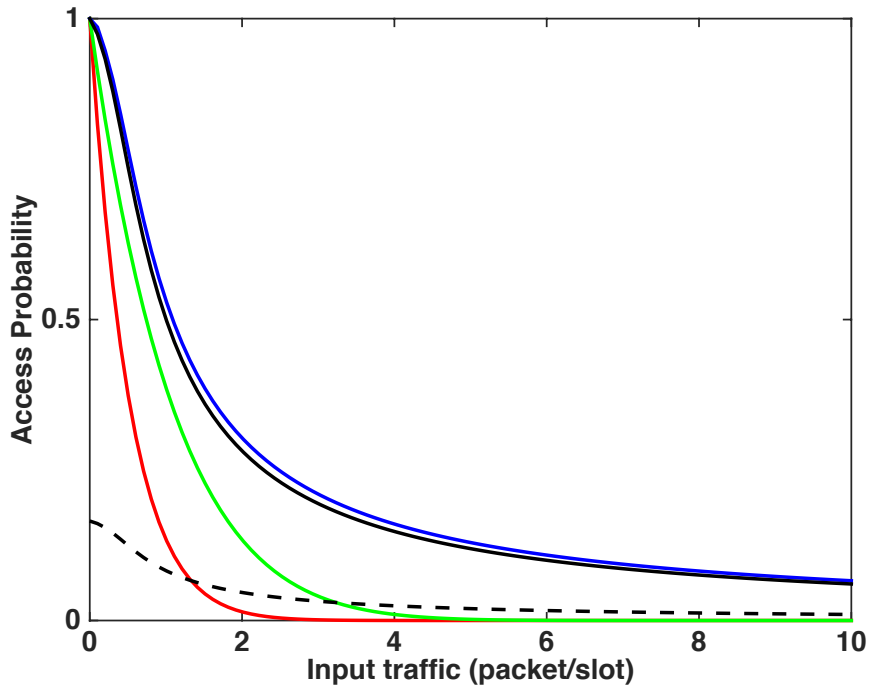

Figure 6.5: Access probability for the model when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01 of the multi-hop network (the dashed black line) compared to the access probabilities of the $p-$persistent CSMA/CD (the blue line), the single hop network (the black line), slotted ALOHA (the green line), and pure ALOHA (the red line).

Fig 6.5 shows the access probability of our model when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01. The dashed black line is the access probability of the multi-hop network, the blue line is the access probability of $p-$persistent CSMA/CD, the black line is the access probability of the single hop network, the green line is the access probability of slotted ALOHA, and the red line is the access probability of pure ALOHA. It is shown that the access probability of our model in multi-hop network is the lowest.

The average number of attempts for a successful transmission is calculated by the following equation:

$$\begin{aligned} n_a &= \sum_{i=1}^{\infty} i((1-p_a)^i p_a) \\ &= \frac{p_a}{1-p_a} \end{aligned} \tag{6.15}$$

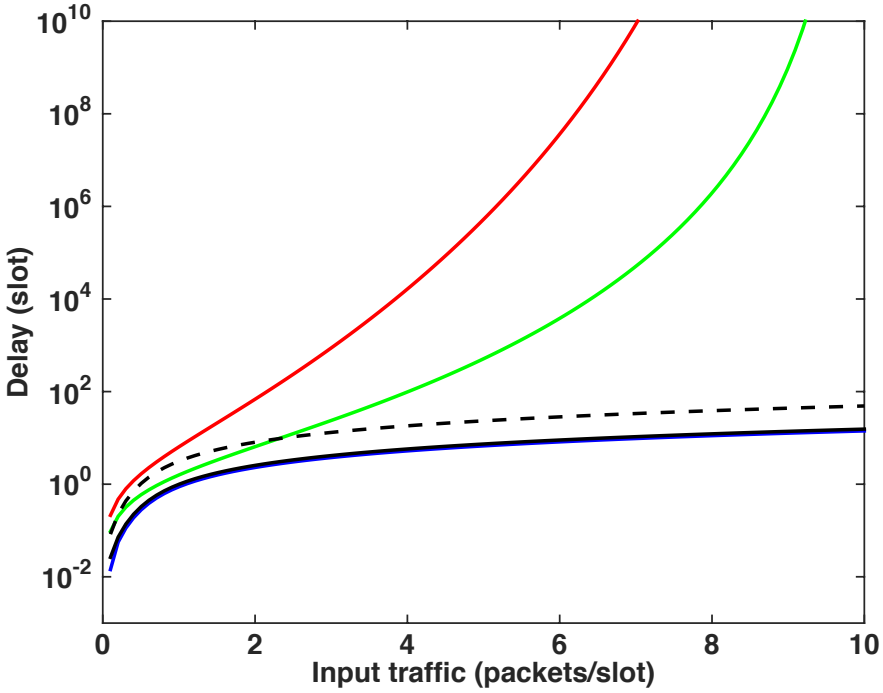

Figure 6.6: The frame delay when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01 of the multi-hop network (the dashed black line) compared to the frame delay of the single hop network (the black line), $p-$persistent CSMA/CD (the blue line), slotted ALOHA (the green line), and pure ALOHA (the red line).

Fig. 6.6 shows the delay of our model when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01. The dashed black line shows the delay of the multi-hop network, the black line shows the delay of the single hop network, the blue line is the delay of

p-persistent CSMA/CD, the green line is the delay of slotted ALOHA, and the red line is the delay of pure ALOHA. The delay of our protocol is a bit higher than the delay of the $p-$persistent CSMA/CD protocol but less than slotted ALOHA and pure ALOHA .

The average energy required to transmit a frame is obtained by the following equation:

$$\begin{aligned} E &= E_0\sqrt{N}\sum_{i=1}^{\infty}(i+1)(1-p_a)^i p_a \\ &= \frac{E_0\sqrt{N}}{p_a} \end{aligned} \tag{6.16}$$

where E_0 is the energy required to send the one frame. In dB, the above equation can be written as

$$\frac{E}{E_0\sqrt{N}} = -\ 10\ log_{10}\ p_a \quad dB \tag{6.17}$$

Fig. 6.7 shows the average energy required to transmit a packet for our protocol when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01. The dashed black line is the energy of the multi-hop network, the black line is the energy of the single hope network, the blue line is the energy of $p-$persistent CSMA/CD, the green line is the energy of slotted ALOHA, and the red line is the energy of pure ALOHA. The average energy of the multi-hop network model is higher than the average energy of the $p-$persistent CSMA/CD protocol but less than slotted ALOHA and pure ALOHA.

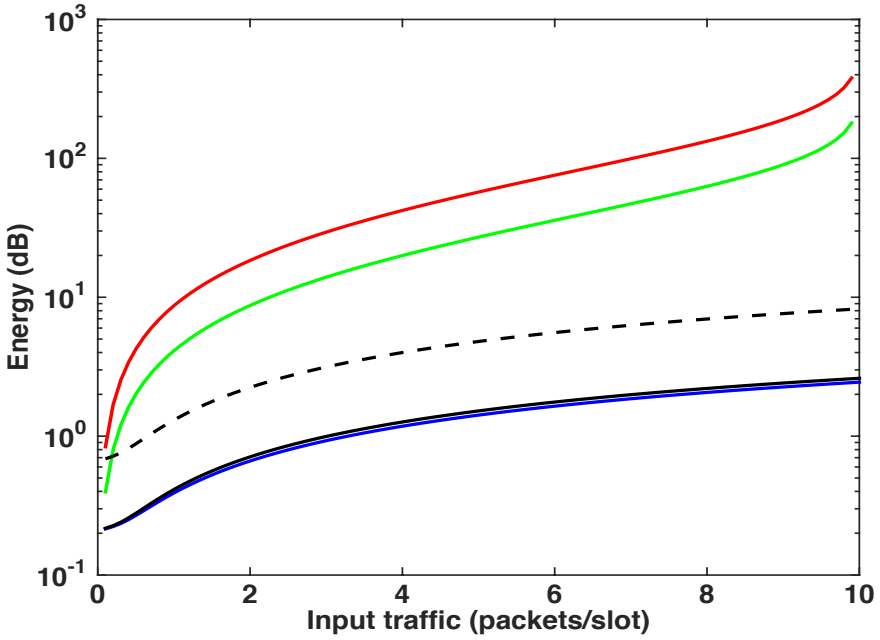

Figure 6.7: The average energy for our model when $n = 20$, $N = 10$, $p = 0.05$, $N_b = 2$ and γ=0.01 of the multi-hop network (the dashed black line) compared to $p-$persistent CSMA/CD (the blue line), the single hop network (the black line), slotted ALOHA (the green line), and pure ALOHA (the red line).

Then, we have simulated the effect of the number of black hole nodes (N_b) on the network performance in order to compare the results. Fig. 6.8 shows the effect of the number of the black hole nodes that are located within the range of the network on the average throughput of the network. It shows that the throughput has been affected according to the number of black hole nodes.

Fig. 6.9 illustrates the effect of the number of black hole nodes (N_b) which are located within the range of the network on the average access probability of this network. It also shows that N_b has high impact on the network access probability.

Fig. 6.10 shows the effect of the number of black hole nodes (N_b) which are located within the range of the network on the average network frame delay. It represents

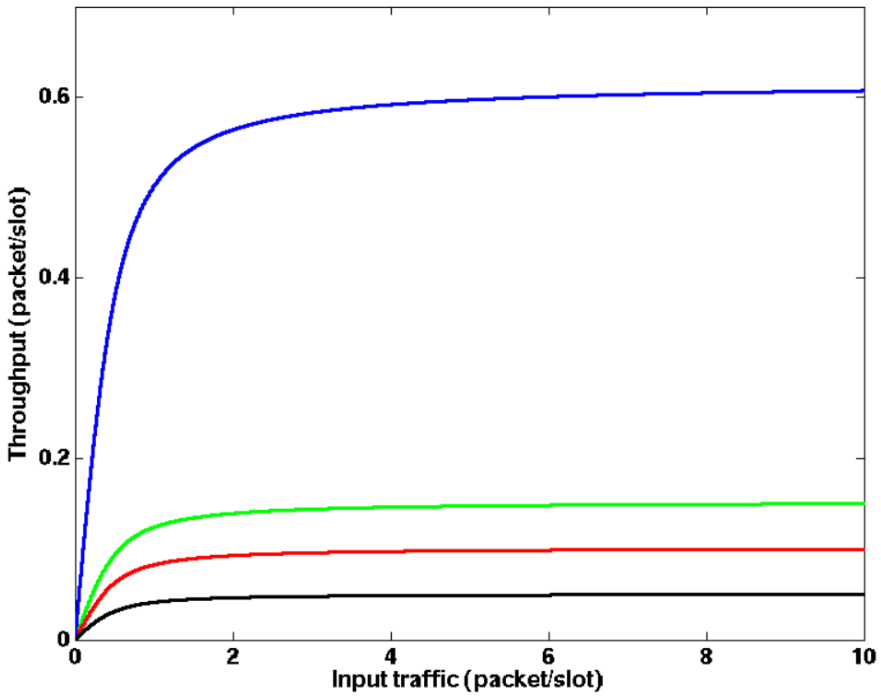

Figure 6.8: The effect of the number of black hole nodes on the average throughput of the multi-hop ad hoc network model. The black line represents the network throughput with three black hole nodes. The red line represents the throughput of the network with two black hole nodes. The green line represents the throughput of the network with one black hole node. The blue line shows the throughput of the network without any black hole nodes.

that the delay increases when there are N_b within the range of the network.

Fig. 6.11 illustrates the effect of the number of black hole nodes (N_b) which are located within the range of the network on the average energy consumption. It shows that the energy is less consumed when there is no black hole nodes in the network.

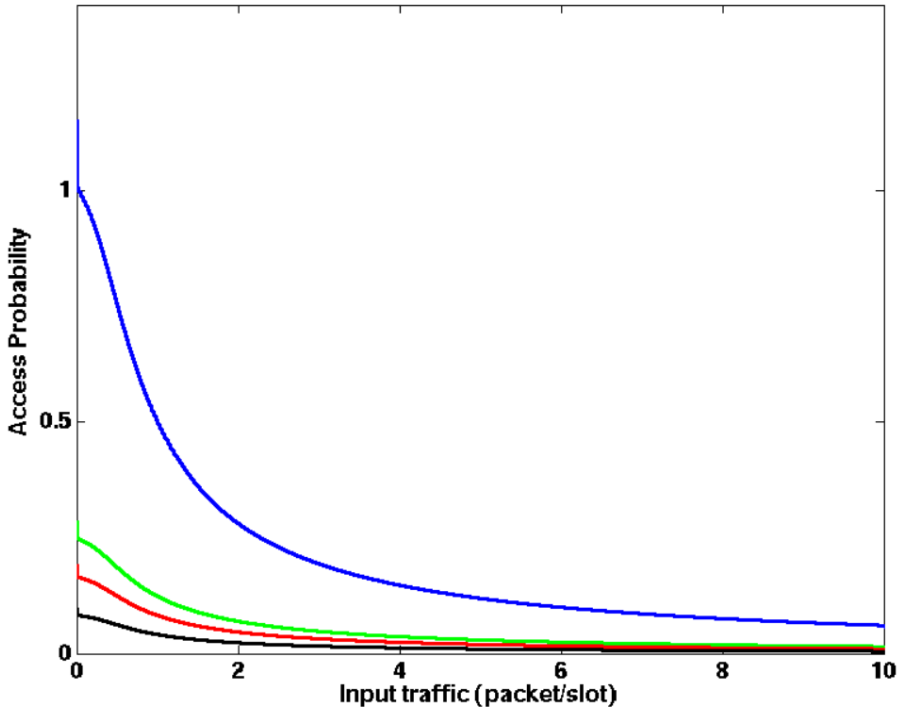

Figure 6.9: The effect of the number of black hole nodes on the average access probability of the multi-hop ad hoc network model. The black line represents the network access probability with three black hole nodes. The red line represents the network access probability with two black hole nodes. The green line represents the network access probability with one black hole node. The blue line shows the network access probability without any black hole nodes.

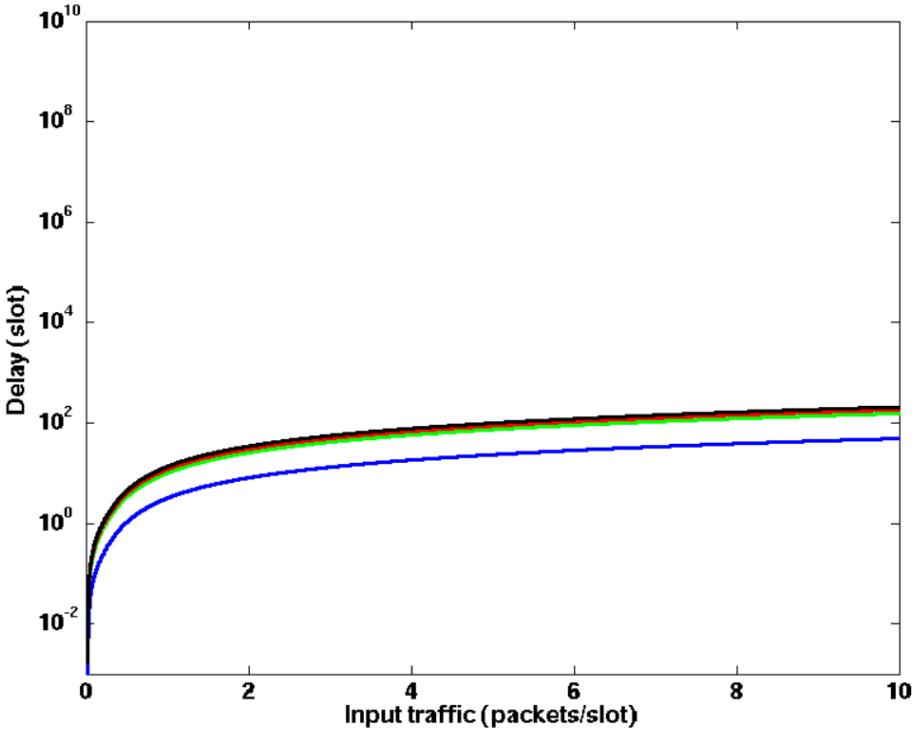

Figure 6.10: The effect of the number of black hole nodes on the average delay of the multi-hop ad hoc network model. The black line represents the network delay with three black hole nodes. The red line represents the network delay with two black hole nodes. The green line represents the network delay with one black hole node. The blue line shows the network delay without any black hole nodes.

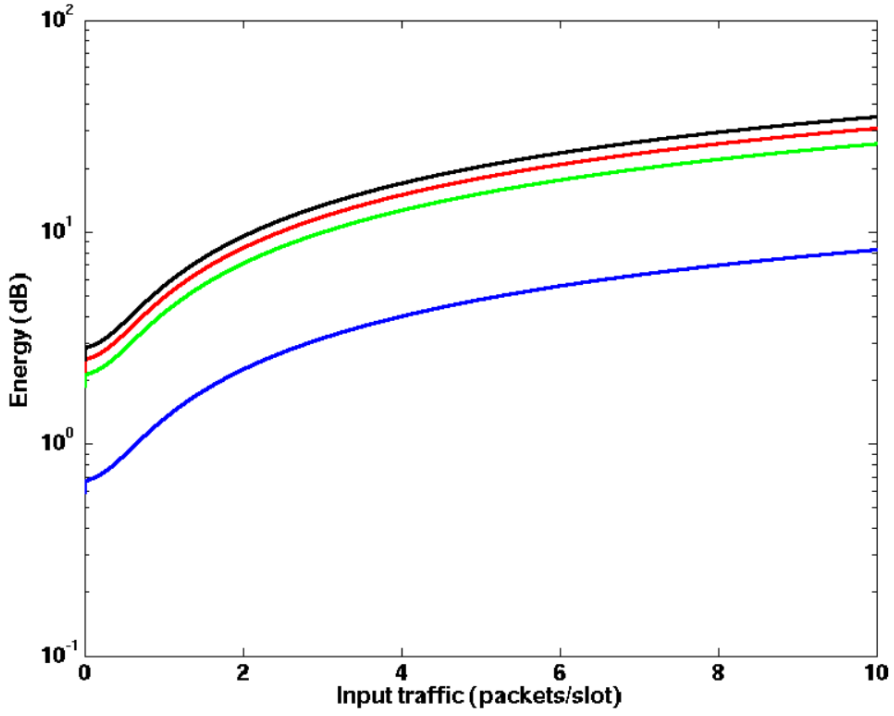

Figure 6.11: The effect of the number of black hole nodes on the average energy of the multi-hop ad hoc network model. The black line represents the network energy with three black hole nodes. The red line represents the network energy with two black hole nodes. The green line represents the network energy with one black hole node. The blue line shows the network energy without any black hole nodes.

Chapter 7

Discussion, Conclusion and Future Work

In this chapter, the results' contributions and limitations are presented based on the analyses conducted in previous chapters. Additionally, this chapter highlights some avenues for possible future research based on this study.

7.1 Discussion

7.1.1 Contributions

1. **A secured protocol against Black-hole attacks in MANETs:**
   In this protocol, we divide the nodes in this network into two groups; leaders group and normal nodes group. Any communications between two normal nodes are controlled and the forwarding packets have to be sent only through the groups' leaders.

2. **The Simulation of the secured protocol by network simulator NS2:**
   We simulate and evaluate the proposed protocol using the network simulator

NS-2.35. We present all the results from the simulation in Chapter 5.

3. **A Markov chain model for the black-hole effects using RTS/CTS:** In this model, we mathematically analyze the effects of the black hole attack in MANETs using a stochastic model. This model describe the characteristics of the system efficiently and therefore they can be solved.

7.1.2 Limitations

Various limitations might exist in this work. The primary limitation is that our analytical and numerical models should have been built differently where we can define some relationships between them in order to compare both of their results. In the node clustering AODV model, we should have considered the expected number of nodes in the range of 600×600 mobile ad hoc networks before design the model as in [33]. In Markov chains model, the location of the nodes should have been taken into account in order to accurately calculate the average number of hops. These limitations should be under consideration in the future work to get accurate results.

7.2 Conclusion

Due to MANET's characteristics, this network has become very attractive in many aspects as same as it has lots of attacks scenarios, one of them is the black hole attack which is a type of denial of service attack. In this work the problem of black hole attacks in Mobile Ad hoc Network has been addressed and a secure protocol has been proposed in order to avoid this type of attacks. The main idea is controlling the communications between nodes within a network. Routing and forwarding packets will be only through the leader nodes. Additionally, once a normal node is joined a group in the network, it will get a special key with information table from the leader of its group. This algorithm is less complex and assumed to improve network's

throughput.

In addition, we also proposed a Markov chain model for the black hole attack effects using RTS/CTS mechanism. We consider randomly the behavior of one user in this model. The simulation for this model has been done by Matlab. The results of our protocol were compared with $p-$persistent CSMA/CD, slotted ALOHA, and pure ALOHA. Our model performance has the lowest throughput due to the black hole nodes.

7.3 Future Work

Our proposed clustering architecture is evaluated using simulation experiments. The simulation results show that the algorithm builds stable clusters with low communication overhead due to its localized and reactive nature. The findings also show that the energy consumption is low comparing to the regular AODV protocol. In the future, this method will be applied for different types of routing attacks such as gray-hole attack in order to test its effectiveness.

Appendix A

Nam Screenshots

In this section, there are some examples from the simulation results that were obtained by NS-2.35. We present some pictures of the network topology with some explanations.

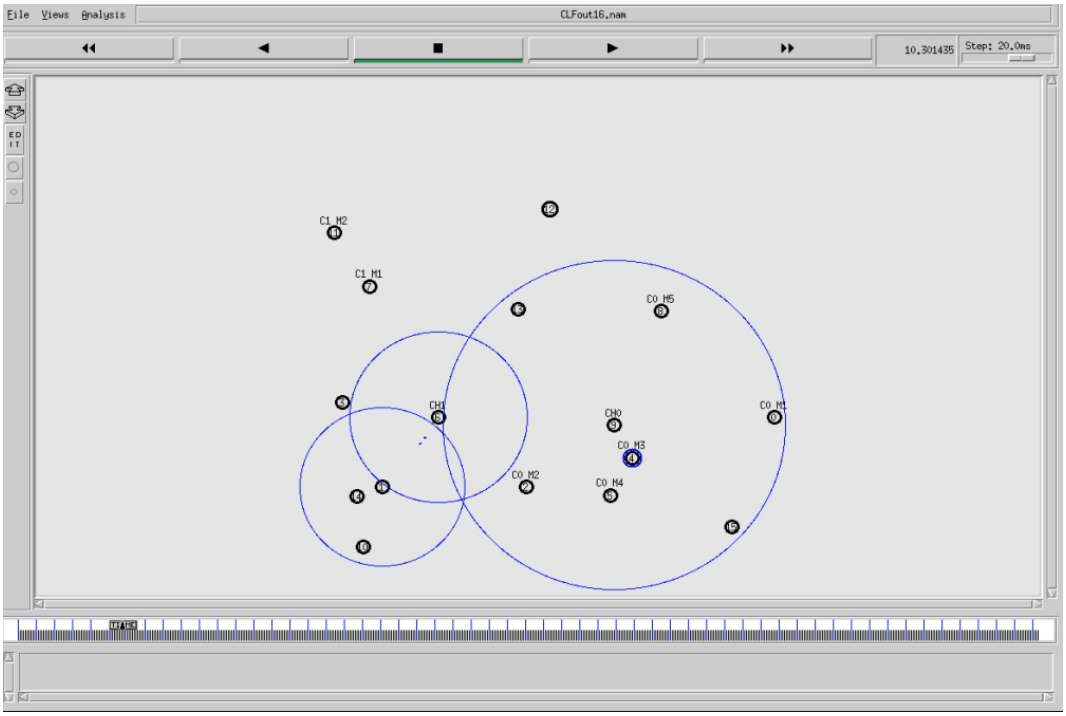

Figure A.1: Clusters are being Formed

After the simulation, the trace file is obtained, this file can be animated by Nam [13]. By generating a Nam file, users are able to follow the network topology and see the packet movement; i.e., how a packet, which is represented by a small rectangle, is moved between different nodes along links. Users also can direct Nam on how to display network components.

In Fig. A.1, In this figure, the clusters are started to be formed. The figure shows the network topology. The nodes are randomly placed in the field of 600m 600m area. As described previously, the nodes use wireless channels following Two Ray Ground Reflection propagation model and transmits packets with Omni-directional Antennas. The traffic between the nodes is considered to be constant bit rate (CBR) and the routing agent is AODV.

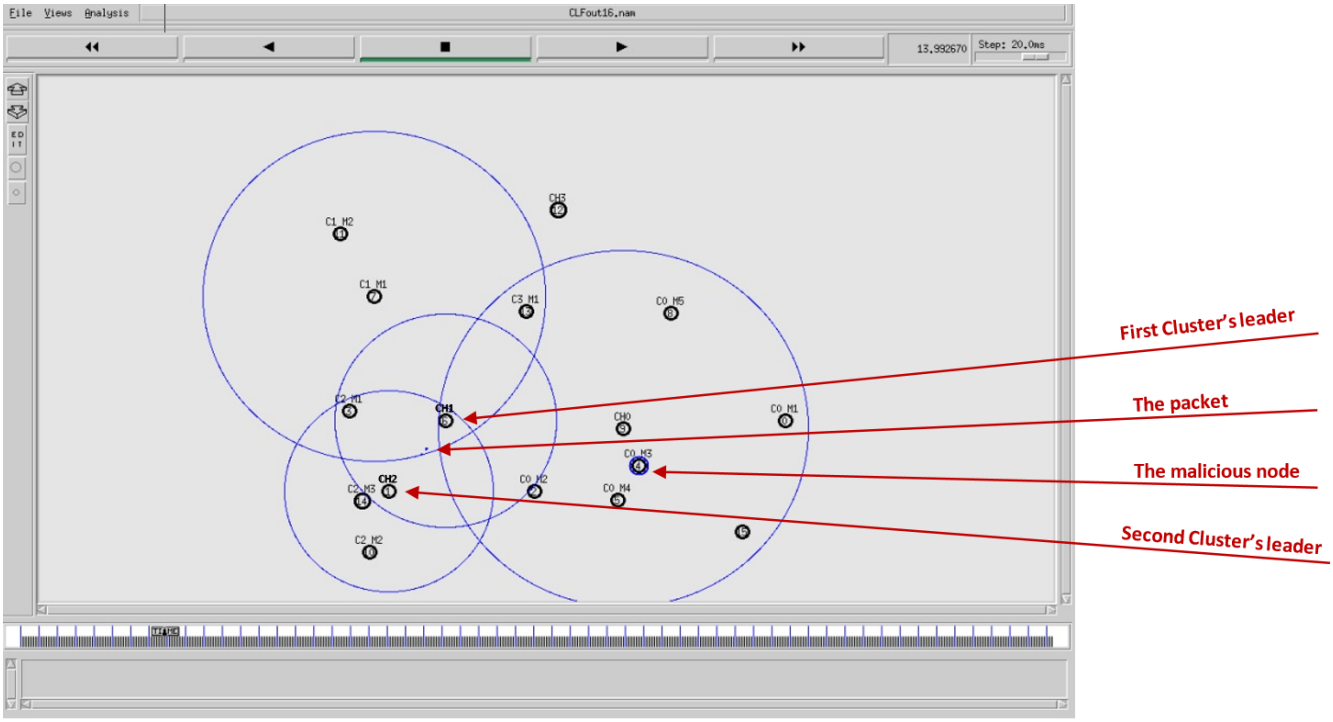

Figure A.2: A packet is sent from cluster #2 to cluster #1

In Fig A.2, malicious nodes are colored blue and good nodes are colored black. Node member #3 in the cluster #2 sends a packet to the node member #1 in the cluster #1 through clusters' leaders (node #1 and node #6). This network has only one black hole node which is node member #4 in the cluster #0.

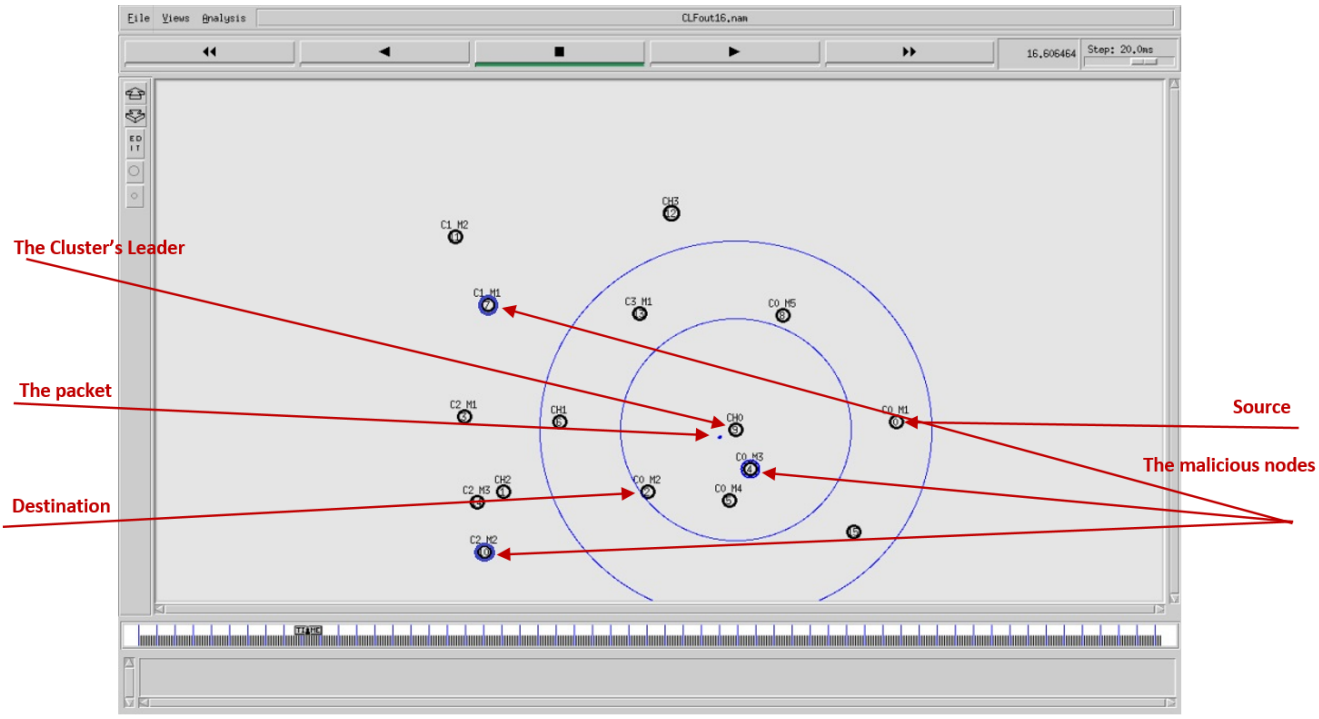

Figure A.3: A packet is sent from node to node within one cluster

Fig. A.3 shows that node member # 4 in the cluster #0 sends a packet to node member #2 within the same cluster through the cluster's leader which is the node #9. This network has three black hole nodes, which are node #4 in cluster #0, node #7 in the cluster #1, and node #10 in the cluster #2.

Bibliography

[1] E. Gerhards-padilla, N. Aschenbruck, P. Martini, *Detecting Black Hole Attacks in Tactical MANETs using Topology Graphs Network*, 32nd IEEE Conference on Local Computer Networks, 2007.

[2] S. Dokurer, Y. Erten, C. Acar, *Performance analysis of ad-hoc networks under black hole attacks*, IEEE, 2007.

[3] S. Djahel, F. bdesselam, Z. Zhang, *Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges*, IEEE Communications Surveys and Tutorials, 2011.

[4] R. Das, B. Purkayastha, P. Das, *Security Measures for Black Hole Attack in MANET: An Approach*, International Journal of Engineering Science and Technology (IJEST), 2011.

[5] T. Santhamurthy, *A Comparative Study of Multi-Hop wireless Ad-Hoc Network Routing Protocols*, International Journal of Computer Science (IJCS), 2011.

[6] R. Karpaga, R. Brinda, P. Chandrasekar, *Detection and Removal of Co-Operative Black Hole Attack in Manet*, International Journal of Computer Applications (IJCA), 2012.

[7] R. Kaur, A. Kaur, *Blackhole Detection in MANETs Using Artficial Neural Networks*, International Journal For Technological Research in Engineering (IJTRE), 2014.

[8] K. Gupta, M. Gujral, Nidhi, *Secure Detection Technique Against Blackhole Attack For Zone Routing Protocol in MANETS*, International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2013.

[9] S. Saini, V. Saroha, *Analysis and Detection of Black Hole Attack in MANET*, International Journal of Science and Research (IJSR), 2013.

[10] A. Sherif, M. Elsabrouty, Amin Shoukry, *A Novel Taxonomy of Black-hole Attack Detection Techniques in Mobile Ad-Hoc Network*, 16th IEEE International Conference on Computational Science and Engineering, 2013.

[11] G. Wahane,and A. Kanthe, *Technique for Detection of Cooperative Black Hole Attack In MANET*, International Organization of Scientific Research (IOSR) Journal of Computer Science, 2014.

[12] C. Perkins, E. Belding-Royer, S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, The Internet Society, 2003.

[13] T. Henderson, *The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns*, 2011.

[14] J. Cano, P. Manzoni, *A Performance Comparison of Energy Consumption for Mobile Ad Hoc Networks Routing Protocols*, 8th International Symposium MASCOTS, 2000.

[15] S. Marti, T. Giuli, K. Lai, M. Baker, *Mitigating routing misbehavior in mobile ad hoc networks*, International conference on mobile computing and networking, 2000.

[16] X. Zou, B. Ramamurthy, *A Simple Group Diffie-Hellman Key Agreement Protocol without Member Serialization*, Computational and Information Science, 2004.

[17] D. Eastlake, *US Secure Hash Algorithms (SHA and HMAC-SHA), http://tools.ietf.org/html/rfc4634*.

[18] S. Ramaswamy, *Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks*, Department of Computer Science, IACC 258 North Dakota State University, Fargo, ND 58105.

[19] F. Tseng, *A survey of black hole attacks in wireless mobile ad hoc networks*, Human-centric Computing and Information Sciences 2011.

[20] E. Khin,*IMPACT OF BLACK HOLE ATTACK ON AODV ROUTING PROTOCOL*, International Journal of Information Technology, Modeling and Computing (IJITMC), 2014.

[21] F. Gebali, *Analysis of Computer Networks*. Second Edition, Springer, 2015.

[22] X. Xue, J. Leneutre, L. Chen, J. Ben-Othman. *A secured watchdog for ad hoc networks*. International Journal of Computer Science and Network Security (IJCSNS), 2006.

[23] Y. Wang, *A Tutorial of 802.11 Implementation in ns-2, http://www.winlab.rutgers.edu/ zhibinwu/pdf/tr_ns802_11.pdf*, 2013.

[24] IEEE Standard 802.11 Part 11: *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, 1999.

[25] T. Issariyakul, E. Hossain, *Introduction to Network Simulator 2 (NS2)*, Springer, 2011.

[26] S. Behzad, S. Jamali, *A Survey over Black hole Attack Detection in Mobile Ad hoc Network*, International Journal of Computer Science and Network Security (IJCSNS), ,2015.

[27] J. Wang, *ns-2 Tutorial (1), http://www.cs.virginia.edu/ cs757/slidespdf/ cs757-ns2-tutorial1.pdf*, Multimedia Networking Group, The Department of Computer Science, UVA, 2004.

[28] J. Wang, *ns-2 Tutorial Exercise, http://www.cs.virginia.edu/ cs757/ slidespdf/cs757-ns2-tutorial-exercise.pdf*, Multimedia Networking Group, The Department of Computer Science, UVA.

[29] D. Ahmed, *Multicasting in Ad Hoc Networks*, University of Ottawa, 2005.

[30] M. Elboukhari, M. Azizi, A. Azizi,*Impact Analysis of Black Hole Attacks on Mobile Ad Hoc Networks Performance*, International Journal of Grid Computing & Applications (IJGCA), 2015.

[31] A. Mishr, *Security and Quality of Service in Ad Hoc Wireless Networks*, Cambridge University Press, 2015.

[32] M. Alnaghes, F. Gebali, *A Survey on Some Currently Existing Intrusion Detection Systems for Mobile Ad Hoc Networks*, 2nd International Conference on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC), 2015.

[33] M. Tamilarasi, T. Palanivelu, B.Rajan, S. Das, *Node Optimization in MANETs for Maximum Throughput using On-Demand Routing Protocols*, 11th National Conference on Communications, 2005.

[34] M. Alnaghes, F. Gebali, *A Graph Theoretic Modeling for Securing Link Layer in Mobile Ad Hoc Networks*, International Journal of Engineering Science and Innovative Technology (IJESIT), 2015.