Hindawi
Security and Communication Networks
Volume 2019, Article ID 8249108, 12 pages
https://doi.org/10.1155/2019/8249108

Research Article

Avoiding and Isolating Flooding Attack by Enhancing AODV MANET Protocol (AIF_AODV)

Mahmoud Abu Zant and Adwan Yasin

Computer Science Department, Arab American University, State of Palestine

Correspondence should be addressed to Adwan Yasin; adwan.yasin@aaup.edu

Received 16 October 2018; Revised 9 January 2019; Accepted 4 February 2019; Published 28 March 2019

Academic Editor: Prosanta Gope

Security in MANET is an essential task in preventing the harm that could be caused by malicious nodes in the network. Flooding attack is one of DoS attacks that aim to exhaust the network resources by flooding the network with a lot of fake packets and messages. There are different forms of Flooding attacks, and the most common form is the request one. Request Flooding attack keeps flooding the network with a lot of requests to the fake nodes that do not exist in the network. In this research, we presented a new enhanced AODV protocol AIF AODV that can detect and isolate flooding nodes in the network. NS-2.35 is used to simulate and to prove the efficiency of the proposed technique. The results of the enhanced protocol in terms of Throughput, End to End Delay, PDF, ARE, and NRL are very close to the native AODV without Flooding attack. The comparisons with other models showed that the proposed model AIF_AODV has a better Throughput characteristic.

1. Introduction

Network technology is moving towards changing the wired connection between nodes in the network to a wireless connection which makes the network more flexible. There are two types of wireless networks, infrastructure networks and infrastructureless networks. In infrastructure networks, nodes depend on a central node to coordinate the communication between them. But in infrastructureless networks, nodes depend on themselves to coordinate the communication process. Mobile Ad Hoc Network (MANET) is an infrastructureless network that connects mobile nodes via wireless links like radio and microwave signals [1, 2]. In MANET, each node has a limited coverage that can transmit or receive packets within it, and nodes that are located within each other's coverage can communicate directly without the help of other nodes, but when there are two nodes that cannot reach each other directly, they request the help of the other nodes to work as a bridge and forward packets to the distant destination. The control of MANET is hard because the network topology is not fixed and changes frequently. There are three types of routing protocols: proactive, reactive, and hybrid. These protocols are adapted to the changing topology of MANET and appropriate in finding the optimal path between any two nodes that want to communicate with each other. Reactive (on-demand) routing protocols are a type of MANET routing protocol where nodes do not exchange information about other nodes except when a route is needed [3]. In general, MANET has different properties such as autonomous behavior, bandwidth, energy, dynamic topology, and security; see Figure 1. 'Autonomous' means that there is no centralized unit to control the communication between nodes. Bandwidth in MANET is very low compared to wired networks. Energy in MANET is limited because it depends on the battery as a source of energy in most cases. Dynamic topology in MANET is caused by the mobility and the randomized movement of nodes, which keep changing the topology of the network. Because of the above mentioned properties, the nodes and data in MANET are vulnerable to a variety of threats and attacks [4]. Therefore routing protocols in MANET should be provided with algorithms and techniques that detect and prevent these attacks in order to preserve these properties.

(1.1) Problem statement: MANET is threatened by different types of attacks, and the security of MANET

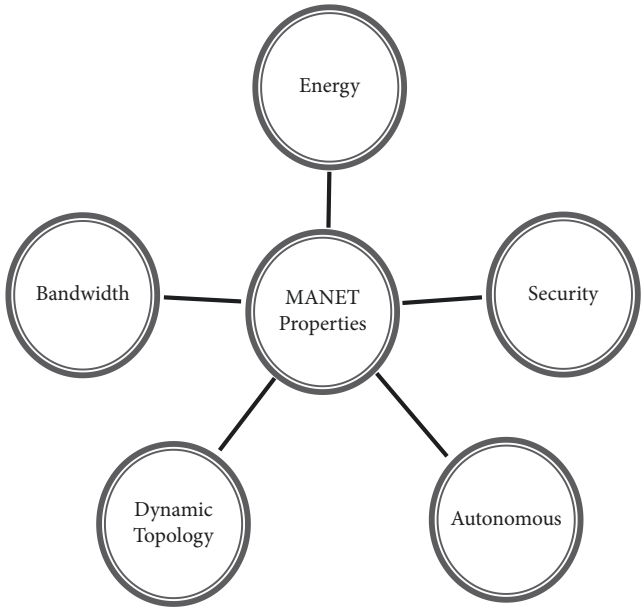

Figure 1: MANET properties.

is important to expose and to prevent the attacks. The Flooding attack, for example, is considered to be one of the Denial of Service (DoS) attacks that threatens the network operations and aims to congest the network with false packets in order to effect the communication between nodes in the network. The native AODV is an on-demand routing protocol, which finds the shortest possible path between nodes in the network, but it lacks a mechanism to detect and prevent the Flooding attack.

(1.2) Our contribution: In this article, we studied Flooding attack and its effect on the network, and we enhanced the AODV to resist Flooding attack with two algorithms, Flooding Avoidance and Attacker Isolation algorithm, since both of them work on avoiding the effect of Flooding attack and preventing its harm in the network. The main ideas of our proposed model are to avoid congesting the network with fake request packets in order to eliminate their bad effects on the network and this is achieved by granting the nodes in the network the capability to decide whether the request is received from an attacker node or from a normal node (self-decision), which helps to avoid false judgment on nodes by putting them in a suspicious list before judging them. Finally, we aimed to make the received requests limit value dynamic by making it change according to the number of neighbor nodes (connectivity). We tested our proposed model, which we call AIF_AODV, by simulating it in different scenarios in which the average maximum speed equals 15 mps, the terrain coordination is 850x850 m, and the flooding interval is 16 requests per second while the number of nodes varies from 20 to 80. The results of the simulation showed the effectiveness of the enhanced AODV in detecting the attacker nodes in a low mobility scenarios network under different performance metrics.

2. Background

2.1. Attacks on MANET. There are two types of attacks in MANET: Passive attacks and Active attacks. In Passive attacks, the attacker nodes only gather information and data about other nodes in the network without affecting the network operations. Examples of these Passive attacks are Monitoring, Eavesdropping, and Traffic Analysis. On the other hand, in Active attacks, the attacker nodes aim to affect the network operation by dropping, modifying, and delaying packets or by altering the path of the packets. Examples of Active attacks are Sybil attack, Wormhole attack, Spoofing attack, Black-Holes and Gray-Holes attack, and Flooding attack [5]. Figure 2 shows the most popular attacks in MANET.

2.2. Flooding Attack. It is an Active attack type that floods the network with the protocol main messages in order to affect the network operation and to consume its resources such as energy and bandwidth. There are several forms of Flooding attack: Hello Flooding, RREQ Flooding, Data Flooding, Error Flooding, and SYN Flooding [6].

(A) Hello Flooding: In this form, the attacker node has a powerful transmitter that has a higher range than the normal nodes. This attacker keeps broadcasting Hello messages convincing other nodes that he is adjacent and a neighbor to them. As a result normal nodes keep forwarding packets to the attacker node hoping to deliver it to the destination node because it has a higher power than any other normal node in the network.

(B) RREQ Flooding: In this form, the attacker node keeps flooding the network with requests (RREQs) for random nodes' IDs that do not exist in the network. Normal nodes keep forwarding these RREQs hoping to find a path of fake nodes. Figure 3 shows RREQ Flooding attack in MANET.

(C) Data Flooding: also called Sleep Deprivation Attack. In this form, two attacker nodes in the network start to transmit an enormous amount of fake data to each other in a high sending rate in order to consume the energy of each normal node that is a part of the path between the two attacker nodes.

(D) SYN Flooding: In this form, the attacker node consumes normal nodes memory by continuously sending an enormous amount of synchronization packets to the victim node.

(E) Error Flooding: In this form, the attacker node should be a part of the path between any two nodes transmitting data to each other or near to them. The attacker node then keeps flooding error messages

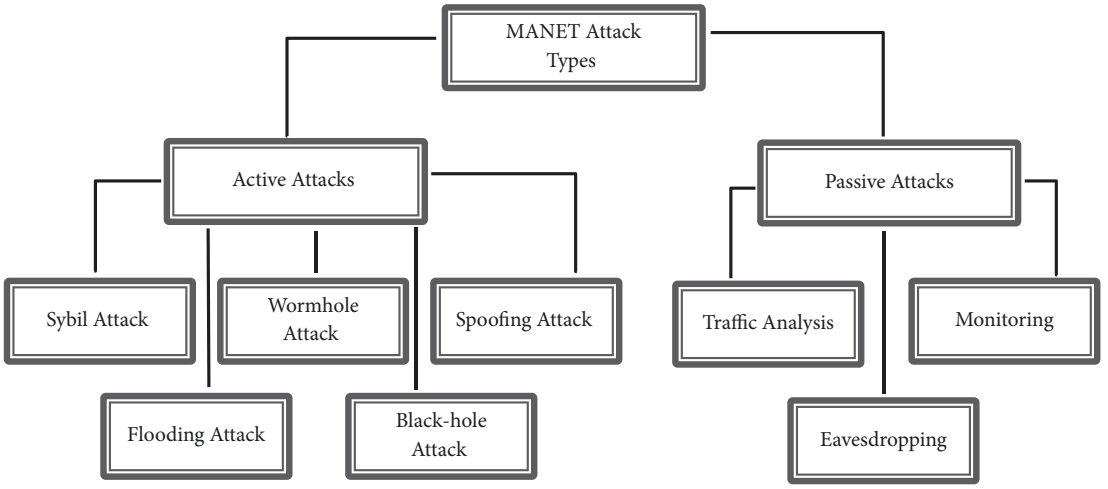

Figure 2: Attacks in MANET [5].

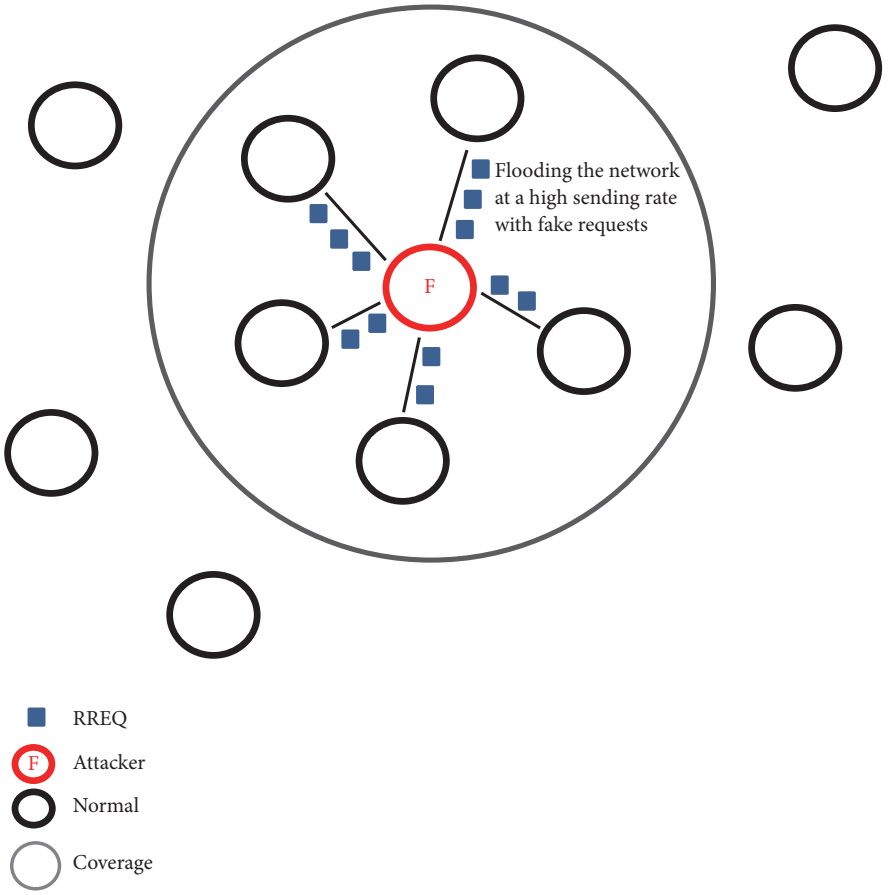

Figure 3: RREQ Flooding in MANET.

(RERRs) to randomly selected nodes within its range. This will lead to interruptions of the transmission process between those nodes because they think that one of the nodes that forwards their packet is unreadable so they start the discovery phase again.

2.3. AODV. Ad Hoc On-Demand Distance Vector (AODV) is one of the most commonly used reactive routing protocols. It uses two types of messages, Route Request (RREQ) message and Route Reply (RREP) message, to discover paths between nodes. Route Error (RERR) is used to maintain and recover these paths. Hello message is used to notify neighbor nodes

about a node's existence. AODV is vulnerable to different types of attacks that may affect its performance under different performance metrics. Flooding attack takes the advantage of the main messages in AODV and uses them to affect the network operation by flooding the network with these messages [7].

3. Related Work

In this section, we will discuss anti-DoS attack techniques and the most known anti-Flooding attack techniques, especially RREQ Flooding as it is considered the most popular form and has the highest impact on the network. Also, we will discuss some limitations for some of the techniques.

Opinder Singh et al. [8] developed a new model called SAODV to detect and isolate the RREQ Flooding attack in MANET. SAODV uses a statistical threshold to detect the attacker node, which depends on two parameters: the mean number of RREQs (MRREQs) made by different nodes in the network and the mean deviation from the mean of all RREQs (MDRREQs). After computing these two parameters, the value of the threshold is set. Any node that sends a number of RREQs higher than the threshold is considered as an attacker node and an alarm will be broadcasted to isolate this node. The results of SAODV showed a high Throughput that is near to the native AODV and a low delay that is also near to the native AODV.

T. Pandikumar et al. [9] proposed a model that prevents the RREQ Flooding attack in MANET. The proposed model employs a Dynamic Profile Based Detection Scheme (DPDS) to detect the attacker node. Each node records the number of sent requests and the number of received requests in order to compute the average of RREQs which is used to compute RATE_LIMIT. The value of RATE_LIMIT is then used to determine the threshold value, and any node sending a number of RREQs exceeding this threshold is isolated and considered as an attacker node. This model decreases the Packet Loss Ratio (PLR) for two different scenarios compared to the native AODV under attack.

Sheetal Jatthap et al. [10] proposed a technique to detect and isolate RREQ Flooding attacker nodes based on their energy. The proposed technique analyzes a node's energy consumption in the network without an attack and then analyzes a node's energy consumption after an attack. The analysis process is performed to determine max and min energy threshold. If the node's energy is equal to or less than the min energy threshold, then the node is dead. And if the sender node has a higher energy than the max threshold, it is considered as an attacker node and is then added to the blacklist in order to isolate it and to avoid communication with it. The results showed a lower protocol power consumption and a lower node power consumption compared to the native AODV under attack.

D. Srinivasa Rao et al. [11] proposed a technique to avoid the RREQ Flooding attack in MANET. The proposed technique depends on dividing the network into clusters to avoid any RREQ Flooding because only cluster head nodes are allowed to broadcast RREQs in the network. Any RREQ that comes from a normal node is dropped. The proposed technique is divided into three phases: Join Network, Cluster Head Election, and Path Cutoff. When a node joins a network in the Join Network phase, it identifies itself and joins the nearest cluster, and then it gets a Unique Identifier (UID). In the second phase, nodes are elected to be a cluster head to control communication between nodes. And in the third phase, when a node receives an RREQ not from a cluster head, the request is then dropped. The results showed a high Packet Delivery Ratio (PDR) that is almost the same as the native AODV but it also showed a higher overhead than the native AODV.

Vrince Vimal et al. [12] developed a technique used to detect and prevent RREQ Flooding attack in MANET. The developed technique has a Detection and Prevention mechanisms. In Detection mechanism, the number of neighbor nodes is used to determine the value of the threshold, which is used to detect the malicious node. Any node that sends a number of RREQs more than the threshold is considered as a malicious node and is added to the blacklist to avoid communicating with it. In Prevention mechanism, neighbor nodes are notified about the malicious node by an alarm packet. To continue the communication normally, routes are modified by replacing any malicious node that forwards packets to destination nodes, with the nearest normal node. The results showed an increase in Packet Delivery Ratio (PDR) up to 95% compared to native AODV under attack and a high Detection Rate of the malicious nodes up to 90%.

Surendra Kumar et al. [13] developed an algorithm to prevent RREQ attack in MANET. Each node has three lists: whitelist, graylist, and blacklist. Whenever a node receives a request, it searches the sender in these three lists. If the sender is from the blacklist, the request is dropped, and if the packet is from a graylist, then it is checked if there is a black alarm broadcasted about the sender node. If such an alarm exists, the request is dropped; otherwise, the request is served. Finally, if the sender is from the whitelist, then the request is served. The judgment on nodes depends on the request number received from the node. If it is higher than the major threshold, then it is in the blacklist and a black alarm is broadcasted. If it is higher than the minor threshold, then it is in the graylist and a gray alarm is broadcasted. Otherwise, it is in the whitelist. Four different scenarios were used to test the performance of the algorithm. The results of all scenarios show an almost equal threshold but varying energy consumption.

Shruti Bhalodiya et al. [14] proposed a schema to detect the RREQ Flooding attack in MANET. The proposed schema uses a filtering technique to check the RREQ_RATELIMIT for every node. Therefore, whenever a node sends RREQs more than the RREQ_RATELIMIT, then it immediately gets blocked and is considered as a flooder node. The value of RREQ_RATELIMIT is static and equals 10 according to RFC 3561. The results showed an increase in Packet Delivery Ratio (PDR), decrease in End to End Delay, and increase in Throughput compared to the native AODV under attack.

M. Rmayti et al. [15] developed a detection system for RREQ Flooding attack in MANET. The developed system has two components: anomaly notification procedure and

malicious flooding detection mechanism. In anomaly notification procedure, each node in the network exchanges information about generated and received requests. This information can be exchanged by a Hello message, which has an extra field that is designed to carry this information. The exchange process is important to periodically keep track of the network's state as each node keeps track of average requests of other nodes in its table, and whenever it receives information about an average request that exceeds the threshold, it triggers the second component. The threshold value is determined by computing Exponentially Weighted Moving Average (EWMA). In the malicious flooding detection mechanism, each node searches its neighbor node's list to find the source of the Flooding attack by comparing the number of received RREQs with RREQ RATELIMIT. After the detection of the attacker node, an RRER message is broadcasted to cut any communication with the attacking node. They simulated the system and found that the system is capable of detecting a Flooding attack node when α =0.25 in EWMA.

Neetu Singh Chouhan et al. [16] proposed a model to prevent RREQ Flooding attack. The proposed model categorizes nodes into three main types: stranger, acquaintance, and friend type. Each node has a table that categorizes each node in it as acquaintance or friend based on the trust level. Any node that does not exist in the table is considered as a stranger node. Each type also has a threshold value that varies from other types, as the friend type has the highest threshold value and the stranger type has the lowest value. Whenever a node receives an RREQ, it first checks the type of the sender node and counts the number of RREQs received. If the number exceeds the threshold value, the sender node is then considered as a malicious node and the receiver node drops any RREQ coming from that node. The results showed higher Throughput values compared to the native AODV under attack.

Shashi Gurung et al. [17] proposed a novel approach to mitigate RREQ Flooding attack in MANET. The proposed approach is called F-IDS. It is divided into three phases: dynamic threshold calculation, confirmation, and resetting phase. In F-IDS, nodes are in the promiscuous mode to observe the nodes' behavior in the network. In the first phase, after a period of time, each node calculates the threshold value based on the standard deviation of the received requests number. In the second phase, if nodes detect a misbehaving node that broadcasts fake requests greater than the threshold, an alarm is broadcasted to all normal nodes to block this node and add it to the blacklist. In the third phase, nodes reset blocked nodes in the blacklist after a period of time, and only if a node has been blocked for three times, then this node will be blocked forever. The results showed a high average Throughput that is near to the native AODV but a higher Normalized Routing Load than the native AODV.

Avita Katal et al. [18] proposed a novel technique to detect and prevent the datagram chunk dropping attack in the network. In datagram chunk attack, the attacker node randomly drops a chunk of datagrams, which has been sent by nodes in the network, and that in turn affects the Throughput of the communication between any two nodes in the network. The proposed technique, which is called Cluster Based Datagram Chunk Dropping Detection and Prevention Technique (CBDCDDPT), is based on clustering the network. In each cluster, a head node is elected by the nodes based on the highest energy, and each cluster head node is responsible for finding the optimal path between any nodes that want to communicate in the network. Each intermediate node including the cluster head has a buffer that consists of two fields: chunk_no and chunk_data. After finding the optimal path between nodes, the source node sends the buffer filled with its corresponding values to the cluster head node, which checks the values of each buffer. If the values are different, then this means that the intermediate node has dropped some chunk of the datagrams, which in turn means that this intermediate node is an attacker node. After the detection and removal of the attacker node, the discovery process between the source and the destination node starts again. The result of the technique shows an enhancement in terms of Throughput.

Mohammad Wazid et al. [19] proposed two techniques that detect the Jellyfish Reorder attack in the network. In Jellyfish Reorder attack, the attacker node reorders the packets sent between the source and the destination node, which in turn affects the goodput of the communication between nodes. Both of the following proposed techniques are based on clustering the network. Generally, all nodes can have the chance to become a cluster head, and the cluster head node is elected based on its effectiveness, for example, if it has high energy. The first proposed technique is called Cluster Based Intrusion Detection and Prevention Technique (CBIDPT). In this technique, each node has a FIFO buffer that stores each sent packet with its corresponding sequence number. An optimal path between the source and the destination node is found by the cluster node. The source node shares the buffer of each packet with the cluster head, and the cluster head compares the sequence number of each packet with all the intermediate nodes in the path. If any of these nodes has a different sequence number (reordered), then this means that there is an attacker node in the path. Following, the cluster head removes the attacker node from the path and searches for a new path. But this technique fails if the attacker node is a cluster head. The second technique is called Super Cluster Based Intrusion Detection and Prevention Technique (SCBIDPT), in which a super cluster is the group of all clusters in the network and a super cluster node is a node that supervises all the cluster head nodes in the network. When the source node sends packets to the destination node, it then shares its buffer with the super cluster node. The aim of the super cluster node is to check the sequence number of each packet in the cluster head nodes and whether there is a different value (reordered), which means that the cluster head node is an attacker node. The super cluster node then removes the attacker node. The results of these two techniques showed a slight increase in terms of End to End Delay but it showed an increase in goodput.

The limitation in [8, 14] is that they depend on a static value as a threshold to detect the attacker node in the network, which should be a dynamic value. The limitation in [8, 12, 13, 17] is that an alarm message is broadcasted to normal nodes after the detection of an attacker node in the

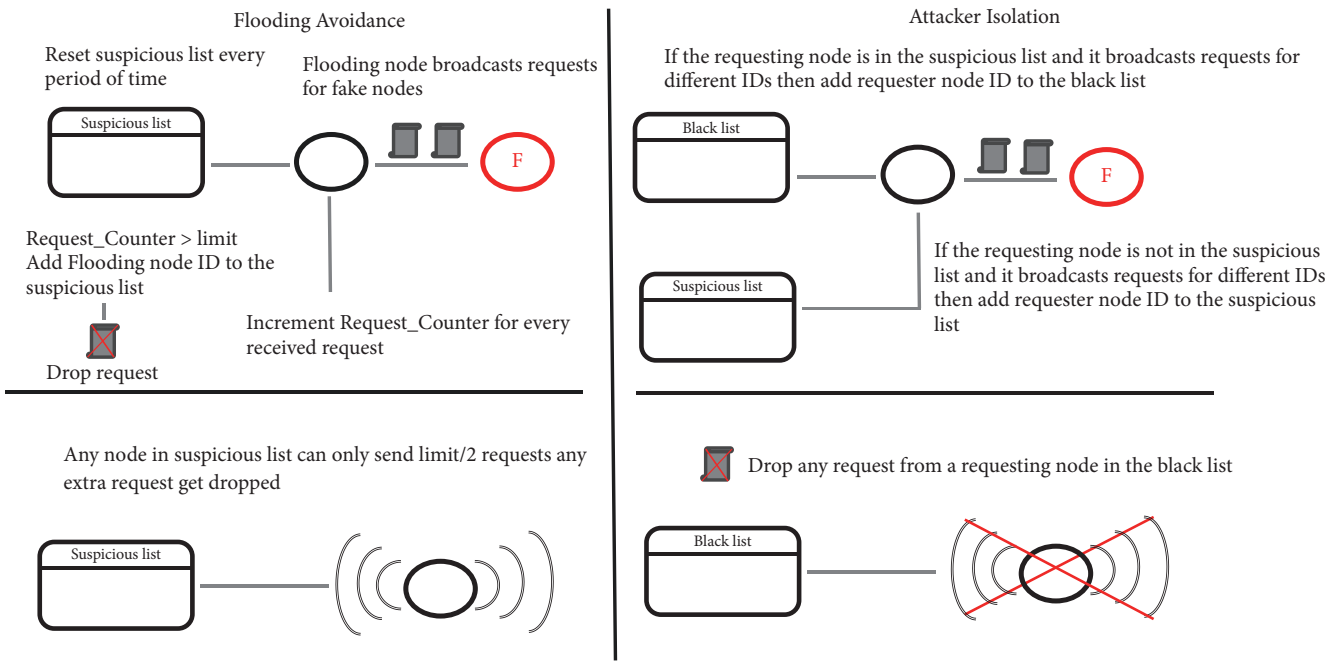

Figure 4: The proposed system model.

network, which makes the network vulnerable to a blackmail attack because a blackmail attacker node can broadcast false alarm messages containing normal nodes IDs to isolate them from other normal nodes in the network. The limitation in [15] is that the detection of an attacker node depends on the exchange of information about other nodes, which makes the network vulnerable to false information exchange by cooperative attacker nodes. The limitation in [11] is that the proposed model depends on clustering the network to detect the attacker node and it is known that clustering has a high overhead in MANET. That is why some network environments avoid clustering. To avoid false information and blackmailing, the detection of the attacker node should be a self-decision, which we were able to achieve in our proposed model.

4. Proposed Model

The proposed model AIF_AODV is developed to avoid the effects of the Flooding attack, identify the attacker, and isolate it (see Figure 4). AIF_AODV consists of two algorithms: Flooding Avoidance and Attacker Isolation algorithm. In Flooding Avoidance algorithm, each node in the network has a table called Request_Counter that records the source of the request and the number of requests received from the same source. Whenever a node receives a request, it first checks if the source of the request is in the Request_Counter table, and then it increases the request counter of that node, or else it adds a new entry for that node in the table. After checking the source of the requesting node, it checks the number of the received requests and if it is higher than the limit, it adds the node to the suspicious list or else processes the request normally. According to AODV RFC [20], any normal node should send up to 10 requests per second. The default value of the limit is set to 10. The limit value varies from half of the limit value to one and a half of the limit value depending on the number of neighbor nodes (closed interval [limit/2, limit∗1.5]). If the number of neighbor nodes is less than half of the limit, then set the limit to limit/2; if it is higher than one and half of the limit, then set the limit to limit∗1.5; otherwise set the limit to an equal number of neighbor nodes. See (1). When the AODV protocol receives Hello message, it stores the ID of the sending neighbor node along with its Destination Sequence Number (DSN) in a table called Neighbors_Table. AODV keeps updating the table by inserting new entries when it receives new Hello messages and by removing old entries when the entry lifetime expires. The number of neighbor nodes (NoN), which is also called connectivity, equals the number of entities in the Neighbors_Table.

$$Lv = \begin{cases} NoN, & \frac{Lv}{2} < NoN < Lv * 1.5 \\ \frac{Lv}{2}, & NoN \leq \frac{Lv}{2} \\ Lv * 1.5, & NoN \geq Lv * 1.5 \end{cases} \quad (1)$$

Lv is the limit value and NoN is the number of neighbor nodes. The default value of Lv is 10 requests per second. To avoid the effects of the Flooding attack, any node in the suspicious list can only send requests up to half of the limit, and nodes only process that number of requests. Any extra request is simply dropped. The avoidance of the attack's effects is achieved by enforcing the nodes to only process a specified number of requests, and hence we prevent flooding the network by attacker requests. The suspicious list gets

```
Begin
 Foreach (received request) Do
   If (source_ID of the request in Request_Counter table) Then
      Increment request_counter of that node;
 End if
 Else Add a new entry for the source of the request to
         Request_Counter table;
 End else
 If  (source_ID of the request in the suspicious list) Then
      limit = limit/2;
 End if
 If  (request counter > limit) Then
      Add source ID to the suspicious list;
      Drop request;
 End if
 Else Process request;
 End else
 End for
End
```

Algorithm 1: Flooding avoidance.

reset every period of time to avoid false judgment on normal nodes. Algorithm 1 describes the Flooding Avoidance.

In Attacker Isolation algorithm, each node has a table called Request_Destination_ID that records the source of the request along with the destination of the request (desired node's ID). We assume that there is no such node in the network that wants to communicate with a large number of nodes at the same time. Whenever a node receives a request, it first checks if the source of the request along with its destination is not in the Request_Destination_ID table, then it adds a new entry for that request. If the number of destinations of a single node is higher than ID_limit, then check if the node is in the suspicious list, and if so, add the node to the blacklist; otherwise add it to the suspicious list. We assumed that ID_limit value is equal to half of the request limit. This algorithm blocks and isolates any node that wants to flood the network with fake requests for different random IDs that do not exist in the network. Algorithm 2 describes the Attacker Isolation. Both mentioned algorithms work together to detect and isolate the Flooding attack in the network.

```
Begin
 Foreach (received request) Do
 If (source_ID of the request and destination not in
        Request_Destination_ID table) Then
   Add a new entry for the source of the request and
   destination to Request_Destination_ID table;
  End if
 If  (source_ID of the request in the Black list) Then
      Drop request;
 End if
 If (ID_request_count > ID_limit) Then
      If (source_ID in the suspicious list) Then
           Add source ID to the Black list;
           Drop request;
      End if
      Else Add source ID to the suspicious list;
           Drop request;
      End else
  End if
 End for
End
```

Algorithm 2: Attacker isolation.

5. Methodology

The proposed model AIF_AODV was developed to prevent congesting the network with request packets and to detect the attacking flooding nodes in it. In order to achieve these goals, AIF_AODV was divided into two algorithms: Flooding Avoidance and Attacker Isolation. We assumed that no normal node in the network will send a number of requests per second higher than the limit value, and also that these requests will not target a number of different nodes that exceeds ID_limit. The idea behind Flooding Avoidance is that when nodes receive a huge number of requests from the same node, they are only processing a specified amount of requests, which equals the limit value in that time and they then drop the rest of the requests. As a result, this node is added to a list called suspicious list, which lowers the allowed number of requests that a node can send to half of the limit at that time. But the nodes in the suspicious list are not considered as attacker nodes, they are only suspicious nodes because it is still unclear whether the sender node is a normal node sending requests to a dead node in the network or the sender node is an attacker node sending requests to a nonexisting node in the network. As a reaction to this uncertainty, a limitation to the allowed number of possible requests sent by

Table 1: Environment parameters.

Simulation Environment Parameters	
Speed	Maximum 15 mps
Pause Time	5s
Simulation Time	Simulation Time 200s
Coordination	850∗850 m
Connection	CBR (Constant Bit Rate)Item size 512(byte)
Radio type	802.11b Radio
Data rate	0.5 Mbps
MAC Protocol	802.11
Routing Protocol	AODV & AIF_AODV
Transport Protocol	UDP
Node Number	20,40,60, & 80
Node Placement	Random
Flooding attack interval	0.06 (16 requests per second)
Transmission range	150 m

a node is an option to prevent the congestion of the network as the rebroadcasting of the requests is controlled. To avoid false judgment on nodes, the suspicious list gets reset every period of time. Because of our assumption that there is no such node in the network that wants to communicate with a large number of nodes at the same time, the idea of Attacker Isolation is to ensure that the requests received by a node are lower than ID_limit; otherwise the sender node is an attacker node and it should be isolated. By combining these two algorithms, we achieved our goal to create a model that enhances AODV to be able to resist Flooding attack.

We used NS-2.35 to simulate and test the proposed AIF_AODV by attacking the network by a flooding node. The creation of scenarios was done using CMU tool, which is a NS-2.35 tool that creates a file containing a random placement and movement of nodes during a fixed period of time. We set the interval of the Flooding attack to 0.06 (16 requests per second). Table 1 shows the environment parameters used in the simulation.

We compared the performance of both native AODV and AIF_AODV under Flooding attack in five different performance metrics: Packet Delivery Ratio (PDR), Throughput, End to End Delay, Average Residual Energy (ARE), and Normalized Routing Load (NRL). AWK scripts were used to obtain values of these performance metrics after analyzing the trace file that is generated by NS-2.35.

Packet Delivery Ratio (PDR) indicates the ratio of packets successfully received by the destination node to the total sent from the source node. PDR can be computed using formula (2).

$$PDR = \frac{R_{packets}}{S_{packets}} \tag{2}$$

$R_{packets}$ is the number of received packets, and $S_{packets}$ is the number of sent packets.

Throughput indicates the rate at which packets are received from the source node over a period of time. Throughput can be computed using formula (3).

$$\text{Throughput} = \frac{R_{packets}}{C_{time}} * \frac{8}{1024} \tag{3}$$

$R_{packets}$ is the number of received packets, and C_{time} is the connection time between nodes.

End to End Delay indicates the average time needed for a packet to be transmitted across the network from the source node to the destination node. End to End Delay can be computed using formula (4).

$$Avg_{EtE} = \frac{\sum_{i-1}^{N} Rt_i - St_i}{N} \tag{4}$$

N is the number of nodes in the network; R_{ti}, S_{ti} are the received and sent time of i^{th} packet consequently.

Average Residual Energy (ARE) measures the average of remaining energy in every node in the network. ARE can be computed using formula (5).

$$ARE = \frac{R_E}{N} \tag{5}$$

R_E is the residual energy and N is the number of nodes in the network.

Normalized Routing Load (NRL) indicates the number of routing packets received over packets received at the destination node. NRL can be computed using formula (6).

$$NRL = \frac{Rt_{packets}}{R_{packets}} \tag{6}$$

$Rt_{packets}$ is the number of routing packets and $R_{packets}$ is the number of received packets at the destination node.

6. Results

After the creation of the random scenarios using CMU tool and after programming both the Flooding attack and the AIF_AODV, we were ready to use NS-2.35 to simulate these scenarios and to test AIF_AODV against Flooding attack. We obtained the following results.

As shown in Figure 5, the result of PDR in native AODV is the lowest when there is a Flooding attack, especially when the number of nodes increased. It is clear that the effect of the attack increases when the number of nodes increases because of the rebroadcasting of fake requests and the overhead of finding the fake nodes in the network. The result of PDR in native AODV is the highest when there is no Flooding attack in the network. The result of AIF_AODV simulation shows a higher PDR than native AODV under Flooding attack, but a slightly lower PDR than native AODV without Flooding attack.

As shown in Figure 6, the result of Throughput in native AODV when there is a Flooding attack is decreasing, while the number of nodes increases as a result of the rebroadcasting of fake requests. The Flooding attack will lead to

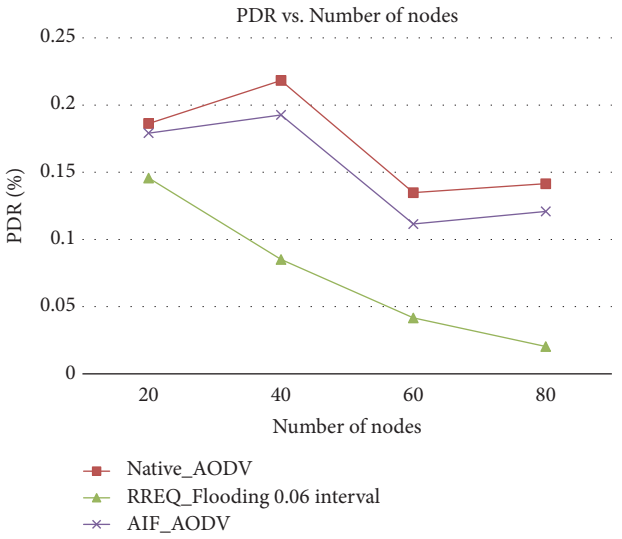

Figure 5: PDR vs. number of nodes.

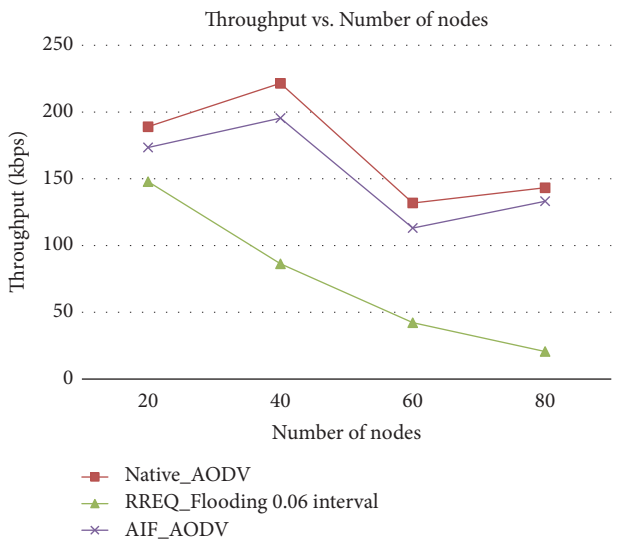

Figure 6: Throughput vs. number of nodes.

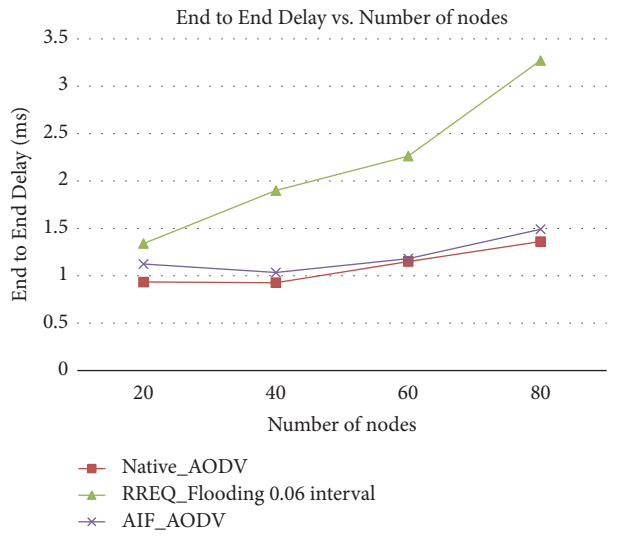

Figure 7: End to End Delay vs. number of nodes.

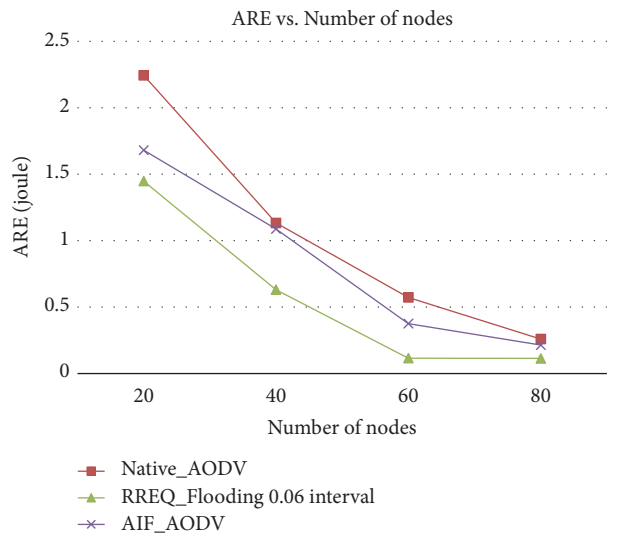

Figure 8: ARE vs. number of nodes.

congestion in the network which also leads to dropping and delaying normal packets, which in turn will affect the Throughput and PDR. The result of PDR in native AODV is the highest when there is no Flooding attack in the network. The result of AIF_AODV shows a higher Throughput than native AODV under Flooding attack and a slightly lower Throughput than native AODV without Flooding attack.

As shown in Figure 7, the result of End to End Delay in native AODV when there is a Flooding attack is increasing, while the number of nodes increases because of the congestion generated by the flooding node. Normal packets will get dropped or delayed, which will increase the End to End Delay. The result of End to End Delay in native AODV when there is no Flooding attack in the network is the lowest. The result of AIF_AODV shows a lower End to End Delay than native AODV under Flooding attack because AIF_AODV detects and isolates the attack node in the network. AIF_AODV shows a slightly higher End to End Delay than native AODV without Flooding attack, because AIF_AODV uses the same mechanism of native AODV in finding the shortest path between nodes that want to communicate.

As shown in Figure 8, the result of ARE in native AODV when there is a Flooding attack is the lowest, especially when the number of nodes increases as the Flooding attack consumes the energy of nodes by keeping them busy in rebroadcasting fake requests in the network. The result of ARE in native AODV when there is no Flooding attack in the network is the highest. The result of AIF_AODV shows a higher ARE than native AODV under Flooding attack because AIF_AODV prevents the Flooding attack in the network. AIF_AODV shows a slightly lower ARE than native AODV without Flooding attack because AIF_AODV has a higher overhead than native AODV because it uses extra tables that store information about nodes.

Table 2: Simulation results of the flooding attack.

Number of Nodes	Native_AODV Without RREQ_Flooding	AIF_AODV	Native_AODV With RREQ_Flooding
	Packet Delivery Ratio (PDR) (%)		
20	0.186	0.179	0.145
40	0.218	0.192	0.085
60	0.134	0.111	0.041
80	0.141	0.120	0.020
	Throughput (kbps)		
20	189.0	173.4	147.8
40	221.5	195.4	86.35
60	131.8	113.1	42.21
80	143.2	133.2	20.64
	Avg of End to End Delay (ms)		
20	0.934	1.124	1.339
40	0.926	1.034	1.899
60	1.149	1.180	2.262
80	1.360	1.490	3.272
	Avg Residual Energy (ARE) (joule)		
20	2.244	1.680	1.447
40	1.133	1.087	0.630
60	0.572	0.375	0.115
80	0.259	0.214	0.113
	Normalized Routing Load (NRL)		
20	0.610	0.880	9.420
40	0.920	1.490	34.14
60	3.130	4.080	104.2
80	3.670	4.610	281.9

As shown in Figure 9, the result of NRL in native AODV when there is a Flooding attack is increasing when the number of nodes increases because the flooding node keeps broadcasting fake requests, and the nodes will continue to rebroadcast these fake requests which will increase the number of routing packets. When the number of nodes increases, the rebroadcasting of fake requests will also increase. The result of NRL in native AODV is the lowest when there is no Flooding attack in the network. The result of AIF_AODV shows a lower NRL than native AODV under Flooding attack and a slightly higher NRL than native AODV without Flooding attack.

Table 2 shows the numeric results of comparison between AIF_AODV and native AODV in terms of PDF, Throughput, End to End Delay, ARE, and NRL.

We implemented AIF_AODV in different scenarios in order to compare it with the two other proposed models [9, 14] from the related work section.

We compared the overall performance of AIF_AODV with the proposed model in [9], which we called DPDS (Dynamic Profile Based Detection Scheme), in terms of Throughput and End to End Delay. The number of attacker nodes increases from 1 to 6, the terrain coordination is 1700x700, and the number of normal nodes varies from 24 to 29. See Figure 10. In DPDS, they obtained a 236.22% increase in Throughput and a 96.23% decrease in End to End Delay. In AIF_AODV, we obtained a 311.32% increase in Throughput and a 40.10% decrease in End to End Delay. By comparing the two results, AIF_AODV proved that it is better than DPDS in terms of Throughput but not in terms of End to End Delay. Table 3 shows the results of comparing both

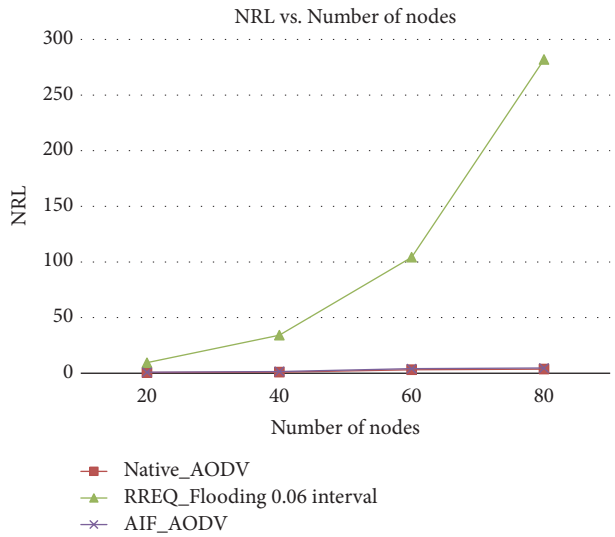

Figure 9: NRL vs. number of nodes.

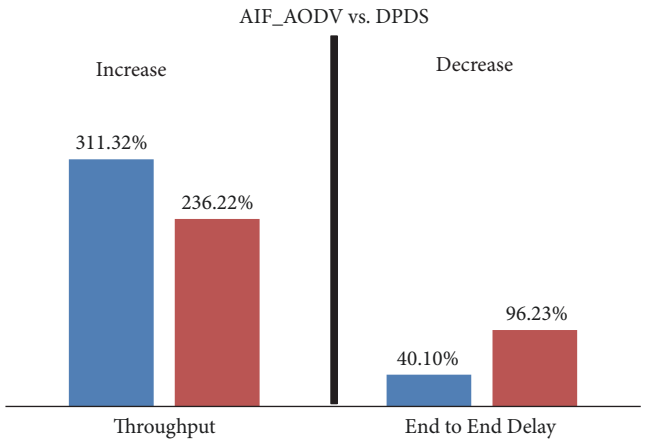

Figure 10: AIF_AODV vs. DPDS in terms of Throughput and End to End Delay.

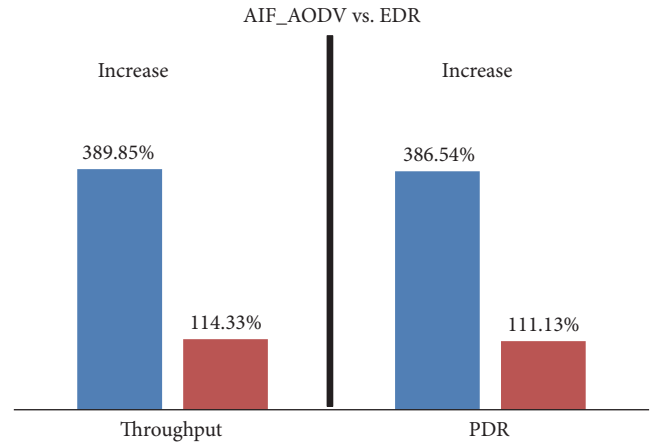

Figure 11: AIF_AODV vs. EDR in terms of Throughput and Packet Delivery Ratio (PDR).

Table 3: Comparison results between AIF_AODV and DPDS.

Metric	AIF_AODV	DPDS
Throughput	311.32% (increase)	236.22% (increase)
End to End Delay	40.10% (decrease)	96.23% (decrease)

Table 4: Comparison results between AIF_AODV and EDR.

Metric	AIF_AODV	EDR
Throughput	389.85% (increase)	114.33% (increase)
PDR	386.54% (increase)	111.13% (increase)

AIF_AODV and DPDS in terms of Throughput and End to End Delay by varying the number of attacking nodes. Out of this comparison, we can see the ability of AIF_AODV to resist multiple flooding attacker nodes in the same network.

We compared the overall performance of AIF_AODV with the proposed model in [14], which we called EDR (Enhanced Detection and Recovery), in terms of Throughput and PDR. The number of nodes increases from 25 to 100 nodes, the terrain coordination is 500x500 m, and the speed of nodes is 3 mps (low mobility scenario). See Figure 11. In EDR, they obtained a 114.33% increase in Throughput and a 111.13% increase in PDR. In AIF_AODV, we obtained a 389.85% increase in Throughput and a 386.54% increase in PDR. By comparing the two results, AIF_AODV proved that it is better than EDR in terms of Throughput and PDR. Table 4 shows the results of the AIF_AODV and EDR comparison in terms of Throughput and PDR while the number of the nodes changes. Note that AIF_AODV works better in low mobility scenarios than high mobility scenarios.

Out of these comparisons, we can conclude that the proposed AIF_AODV has a better Throughput characteristic than other models.

7. Conclusions

Flooding attack is considered one of the Denial of Service (DoS) attacks that consume the network resources. Flooding attack affects the network in different performance metrics. Prevention and detection of a flooding node in the network are important to avoid its effect on the network. AIF_AODV depends on two algorithms to avoid the effects of a Flooding attack in the network and to isolate the attacker node. The simulation results of the proposed AIF_AODV showed that its PDF, Throughput, End to End Delay, ARE, and NRL are very close to the native AODV. AIF_AODV proved its efficiency in avoiding the effects of a Flooding attack, especially in low mobility scenarios. As a future work, we aim to find an algorithm to detect Error Flooding and Sleep Deprivation attack in MANET.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] K. S. Varsha and S. N. Raj, "Applications, challenges and protocols of MANETs: a review," *Asia-Pacific Journal of Convergent Research Interchange*, vol. 4, no. 1, pp. 21–29, 2018.

[2] J. Kaur and G. Singh, "MANET routing protocols: a review," *International Journal of Computer Sciences and Engineering (ICSE)*, vol. 5, no. 3, pp. 60–64, 2017.

[3] M. B. Lubdha, L. Jain, and D. P. Gayatri, "Study of various routing protocols in mobile ad-hoc networks," *The International*

Journal of Scientific Research in Network Security and Communication (IJSRNSC), vol. 6, no. 1, pp. 1–6, 2018.

[4] V. Goyal and G. Arora, "Review paper on security issues in mobile adhoc networks," *International Research Journal of Advanced Engineering and Science (IRJAES)*, vol. 2, no. 1, pp. 203–207, 2017.

[5] M. M. Alani, "MANET Security: A Survey," in *Proceedings of the in International Conference on Control System, Computing and Engineering, IEEE*, Malaysia, 2014.

[6] C. M. Nalayini, J. Katiravan, and A. Prasad, "Flooding attack on MANET – a survey," *International Journal of Trend in Research and Development (IJTRD)*, pp. 25–27, 2017.

[7] M. A. Abdelshafy and P. J. B. King, "Analysis of security attacks on AODV routing," in *Proceedings of the 2013 8th International Conference for Internet Technology and Secured Transactions, ICITST 2013*, pp. 290–295, London, UK, March 2013.

[8] O. Singh, J. Singh, and R. Singh, "SAODV: statistical ad hoc on-demand distance vector routing protocol for preventing mobile adhoc network against flooding attack," *Advances in Computational Sciences and Technology*, vol. 10, no. 8, pp. 2457–2470, 2017.

[9] T. Pandikumar and H. Desta, "RREQ flooding attack mitigation in MANET using dynamic profile based technique," *International Journal of Engineering Science and Computing (IJESC)*, vol. 7, no. 6, pp. 12700–12705, 2017.

[10] S. Jatthap and P. Dashore, "Battery capacity based detection and prevention of flooding attack on MANET," *International Journal of Advance Research in Computer Science and Management Studies (IJARCSMS)*, vol. 4, no. 9, pp. 89–99, 2016.

[11] D. S. Rao and V. Padmanabhuni, "An efficient RREQ flooding attack avoidance technique for adaptive wireless network," *International Journal of Applied Engineering Research (IJAER)*, vol. 11, no. 5, pp. 3696–3702, 2016.

[12] V. Vimal and M. J. Nigam, "Plummeting flood based distributed-DoS attack to upsurge networks performance in ad-hoc networks using neighborhood table technique," in *Proceedings of the 2017 IEEE Region 10 Conference, TENCON 2017*, pp. 139–144, Malaysia, November 2017.

[13] S. Kumar, S. Alaria, and V. Kumar, "Prevention in sleep deprivation attack in MANET," *International Journal of Latest Technology in Engineering (IJLTEMAS)*, vol. 4, no. 2, pp. 139–144, 2015.

[14] S. Bhalodiya and K. Vaghela, "Enhanced detection and recovery from flooding attack in MANETs using AODV routing protocol," *International Journal of Computer Applications*, vol. 125, no. 4, pp. 10–15, 2015.

[15] M. Rmayti, Y. Begriche, R. Khatoun, L. Khoukhi, and D. Gaiti, "Flooding attacks detection in MANETs," in *Proceedings of the International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications, SSIC 2015*, China, August 2015.

[16] N. S. Chouhan and S. Yadav, "Flooding Attacks Prevention in MANET," *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, vol. 1, no. 3, pp. 68–72, 2011.

[17] S. Gurung and S. Chauhan, "A novel approach for mitigating route request flooding attack in MANET," *Wireless Networks*, vol. 23, no. 4, pp. 1–16, 2017.

[18] A. Katal, M. Wazid, R. H. Goudar, and D. P. Singh, "A cluster based detection and prevention mechanism against novel datagram chunk dropping attack in MANET multimedia transmission," in *Proceedings of the 2013 IEEE Conference on Information and Communication Technologies, ICT 2013*, pp. 479–484, India, April 2013.

[19] M. Wazid, A. Katal, and R. H. Goudar, "Cluster and super cluster based intrusion detection and prevention techniques for JellyFish Reorder Attack," in *Proceedings of the 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, PDGC 2012*, pp. 435–440, India, December 2012.

[20] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing (RFC 3561)," 2003.