Mitigation of the Effects of Selfish and Malicious Nodes in Ad-hoc Networks

Houssein Hallani and Seyed A. Shahrestani

School of Computing and Mathematics
University of Western Sydney, Australia

Abstract— A wireless Ad-hoc network is a group of wireless devices that communicate with each other without utilising any central management infrastructure. The operation of Ad-hoc networks depends on the cooperation among nodes to provide connectivity and communication routes. However, such an ideal situation may not always be achievable in practice. Some nodes may behave maliciously, resulting in degradation of the performance of the network or even disruption of its operation altogether. To mitigate the effect of such nodes and to achieve higher levels of security and reliability, this paper expands on relevant fuzzy logic concepts to propose an approach to establish quantifiable trust levels between the nodes of Ad-hoc networks. These trust levels are then used in the routing decision making process. Using OPNET and MATLAB simulators, the proposed approach is validated and further studied. The findings show that when the proposed approach is utilised, the overall performance of the Ad-hoc network is significantly improved.

Keywords: Ad-hoc networks, Behavior analysis, Malicious attacks, Simulation, Throughput.

1 Introduction

Wireless networking has experienced fast development in the last few years. A large number of handhelds, portables, and mobile phones have become implanted with wireless communication capabilities [20]. As a result of this, very small computer devices with wireless communication capabilities will soon be embedded in almost every product. The mobility and the freedom offered by these wireless devices allow users to remain connected to their enterprise networks, while on the move [12].

Modern Wireless Local Area Networks (WLANs) with relatively high data rates have become an attractive technology for providing Internet connectivity for mobile users. Professional deployment of WLANs requires the capability to broaden the coverage without the need to deploy a costly infrastructure. Ad-hoc based wireless networks are an attractive solution for this problem. A wireless Ad-hoc network can be considered as a group of wireless devices with radio frequency connectivity that assist each other in transmission of data packets within the network. Data traffic flows over one or more paths between succeeding nodes to reach its destination, making wireless Ad-hoc networks similar to the structure of the Internet [6]. In a wireless Ad-hoc environment, a network can be seen as a collection of end systems that are free to move randomly while maintaining a reliable connection. This kind of network requires no centralised administration or fixed network infrastructure, and can be easily and inexpensively deployed as needed. Ad-hoc wireless networks have recently received a lot of attention. This is mainly due to their potential to support a variety of applications without the need for a fixed infrastructure [5]. Some of the applications where such networks can be usefully deployed are military applications, emergency, search and rescue applications, university campuses, conferences, and hospitals. A key advantage of Ad-hoc networks over conventional WLAN configurations is that Ad-hoc networks have no single point of failure [9].

Most modern networks are based on pre-established relationships between clients and service providers. In most cases, the movement of users from their established environment may cause various difficulties and problems. To overcome some of these difficulties, wireless Ad-hoc networks provide a number of solutions. The first of these relates to ease and simplicity. A node, which is capable of reaching one or more available neighbouring nodes, can be added easily to the network. Secondly, wireless Ad-hoc networks allow the users to

overcome geographical and location limitations. This is due to the fact that all nodes in the network can provide connectivity as opposed to a single access point. Scalability is also an advantage as Ad-hoc networks are robust and can be easily scaled up. Finally, wireless Ad-hoc networks offer a significant cost saving, as the existing environment does not have to be modified drastically to accommodate the addition of nodes to the existing and evolving network. [2].

In our previous works, the effects of the presence of malicious nodes in an Ad-hoc network have been reported [8]. This included the introduction of the BAODV approach which utilises the behaviour history of the network nodes [7]. In this paper a new approach that is based on fuzzy logic concepts to optimise the evaluation of trust between nodes is introduced. Fuzzy logic provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, or imprecise input information. Different factors and parameters should be identified and combined in order to determine if a node is acting maliciously. Incorporating trust in Ad-hoc routing protocols and thereby mimicking human behaviour can facilitate the detection of nodes that misuse the trust placed in them.

To achieve this, the remainder of this paper is organized as follows. The motivations for using fuzzy logic concepts to evaluate trust levels between nodes in an Ad-hoc network are presented in Section 2. In Section 3, a detailed description of the fuzzy trust evaluation application used to evaluate trust levels between nodes is illustrated. An overview of the fuzzy trust algorithm is given in Section 4 The integration of MATLAB and OPNET is described in Section 5. An outline of the simulation setup together with various scenarios used in this study are presented in Section 6. Collected results and their analysis are discussed in Section 7 which is followed by concluding remarks in Section 8.

2 Motivations

In the last few years, different routing protocols for Ad-hoc networks have been proposed. But most of them tend to ignore the fact that all the nodes in the network will not necessarily fully cooperate in routing the packets from source to destination [21]. In general, many Ad-hoc devices operate on battery power. Consequently, power consumption for each transmission has a certain cost and significance. So, in reality, the assumption that all nodes perform the task of forwarding data, from which they do not directly benefit, while consuming their own battery power, is not always achievable [22]. There is little reason to assume that some nodes will not try to achieve the benefits of participating in the network and avoid the disadvantages it involves. This could mean that some nodes may refuse to forward packets as expected and thereby decrease the efficiency of the network. Due to the dynamic nature of Ad-hoc networks, identifying nodes that express such malicious behaviour is a difficult task. The node originating the transmission might be out of range for detecting the malicious act.

The open structure, lack of existing infrastructure and inaccessibility to trusted servers make traditional security methods and systems insufficient for Ad-hoc networks. This problem, faced with the presence of malicious nodes in Ad-hoc networks, requires the existence of a trust level based algorithm to alleviate the effect of such nodes [4]. In general, in Ad-hoc networks central trustworthy authorities do not exist and no trust relationships are present between the comprising nodes. To address this problem an approach arising utilising fuzzy logic concepts to establish trust relationships between nodes is proposed. To facilitate the quantification of trust levels for a node, information about the behaviour history of this node is collected. Incorporating the concept of trust in Ad-hoc routing protocols and thereby mimicking human behaviour, can further improve the performance and the reliability of Ad-hoc networks. It is expected that the establishment and quantification of trust levels can be used to detect nodes that misuse the trust placed in them. The detection of misbehaving nodes can be used to apply trust based route selection strategies to Ad-hoc routing protocols and thereby increase the effectiveness of the network. Four types of misbehaving nodes are considered in this paper. These include nodes that:

- Drop packets randomly.
- Forward packets to the wrong destination.

- Fabricate and transmit falsified routing messages.
- Launch replay attacks.

The trust level that can be assigned to a node is obviously not a crisp value, due to the multiple factors that can affect the trustworthiness of the nodes. Therefore, combining information related to these attacks by monitoring the neighbouring nodes can facilitate the quantification of trust levels. Thus, a model utilising fuzzy logic concepts is developed. To assign trust levels to nodes of Ad-hoc networks, a fuzzy trust evaluation application is developed using MATLAB. This application receives information about the behaviour history of Ad-hoc network nodes. The trust levels are then used by the routing protocol in an attempt to choose the most reliable route between the source and the destination nodes. This approach is implemented and tested to show its benefits and drawbacks.

3 Overview of the Fuzzy Trust Evaluation Model

In human relationships, trust is often expressed linguistically rather than numerically [15]. Trust plays an important role in the cooperation and interaction between real world entities. It is well established that fuzzy logic is suitable to quantify trust among entities that comprise a network or a group. One of the advantages of using fuzzy logic to quantify trust between nodes in Ad-hoc networks is its ability to quantify imprecise data or uncertainty in measuring the security index of Ad-hoc nodes.

In reality, people tend to interact easily only with those whom they believe to have good behaviour and are trustworthy. Trust can be interpreted as the expectation that a person will act in a reliable and predictable way. If a person has a reputation for not getting jobs done, then people will not trust this person in the future. As a result, people will not assign critical jobs to this person since there is a good chance that the job will not get done [1]. Similarly in Ad-hoc networks, the trust level is affected by the past behaviour of the nodes. A node that in the past demonstrated dependability and responsiveness will gain increasing trust. On the other hand, the unwillingness of a node to cooperate with other nodes will affect its trust level.

As with most other areas of applying fuzzy logic to develop trust models, the process of designing the fuzzy trust evaluation model involves five steps [3]:

- Formulating the problem and selecting the linguistic variables. These variables are the vocabulary of the system in which the rules work;
- Designing the structure of the system which represents the information flow within this system, i.e., what input variables are combined with which other variables via rule blocks;
- Designing fuzzy membership functions for each variable, this is often described in linguistic terms;
- Formulating the strategy through the fuzzy logic rules. As a result, an output value is obtained in linguistic terms; and
- Performing defuzzification to derive an actual crisp value.

3.1 Membership Functions

As in most of the fuzzy logic models, the role of the membership functions in the proposed fuzzy trust evaluation model is to map a crisp input to the corresponding membership degree in linguistic terms.

The reason Gaussian membership functions are utilised in the proposed trust evaluation model instead of simpler triangular functions and sigmoid functions is that adaptability can be easily introduced by simply changing the mean and the variance of the membership functions. Another reason is that a single sigmoid function does not represent a closed class interval. Also, the triangular function will not ensure that all inputs are fuzzified in some class [19].

3.2 Defining the Variables

One of the necessary decision when defining input and output variables is to choose the number of linguistic terms that describe the state of each variable. Most fuzzy logic based models use between three and seven terms for each linguistic variable. One rarely uses fewer than three terms, since most concepts in human language consider at least two extremes and the middle ground. On the other hand, one rarely uses more than seven terms because humans interpret technical figures using their short-term memory. In general, human short-term memory can only compute up to seven symbols at a time [3]. As described in Section 2, the four dominant attacks in Ad-hoc networks are: packet dropping, forwarding messages along wrong paths, fabricating messages and replay attack. In the proposed fuzzy trust evaluation model, the trust level of a node is determined by the percentage of packet dropped, the percentage of packets forwarded to the wrong destination, the number of replay attacks generated by this node, and the number of false routing messages produced by this node. These percentages are treated as fuzzy input variables, characterised by the membership functions shown in Figure 1. The output variable is shown in Figure 2. These variables are:

- Packet_Dropped: This input variable represents the percentage of packets dropped

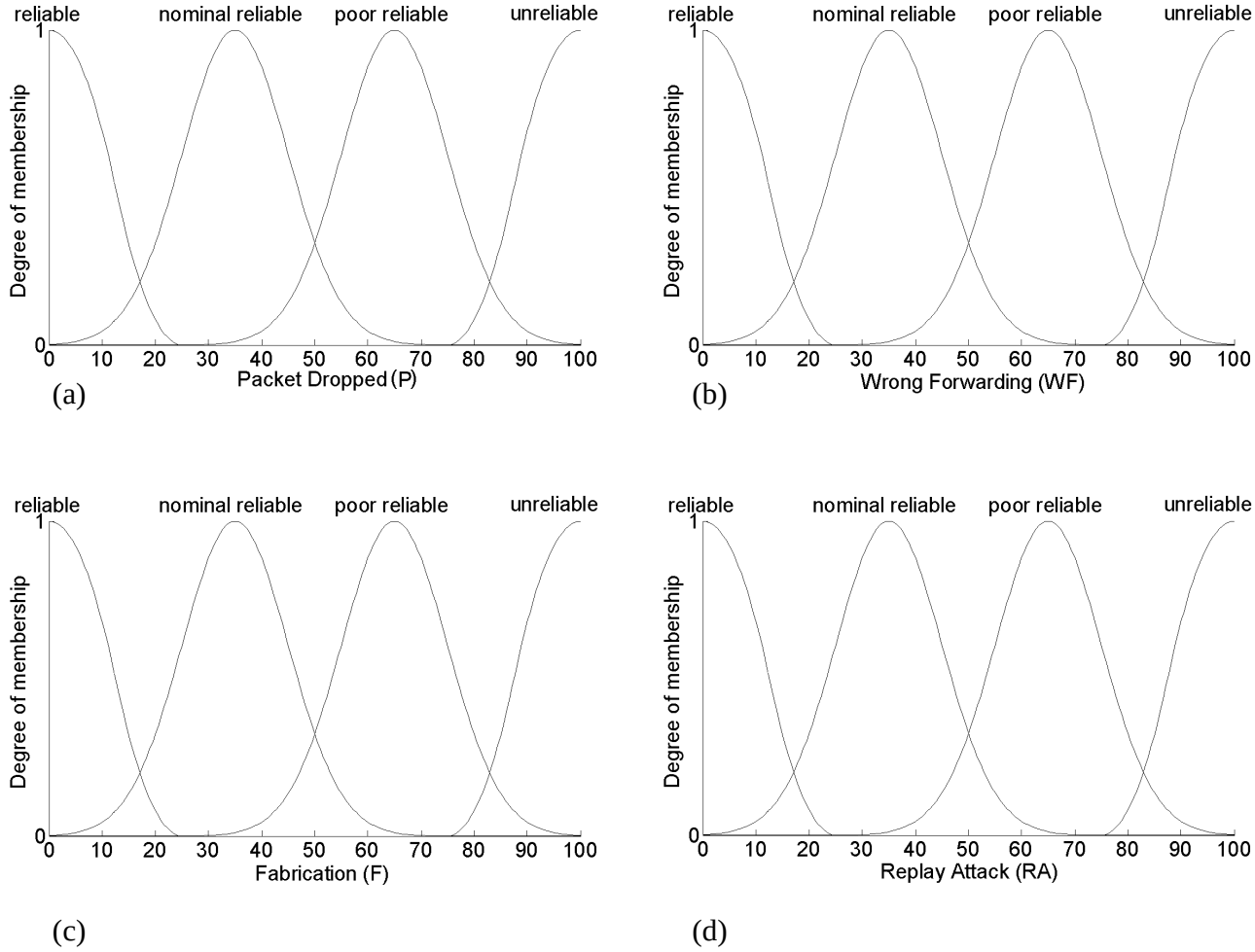

Figure 1Membership functions for the four inputs

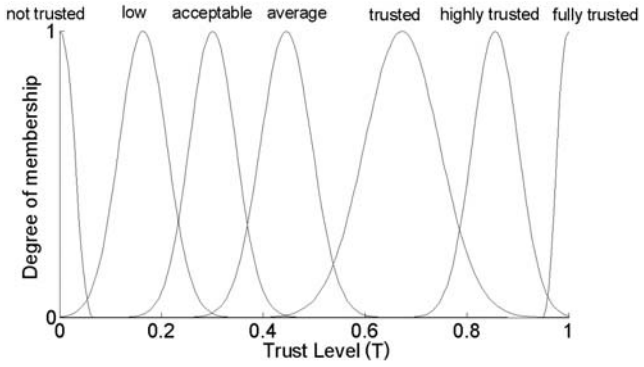

Figure 2 Membership functions for the output

by a node. It has four fuzzy sets defined by the linguistic terms reliable, nominal_reliable, poor_reliable and unreliable. The corresponding membership function P representing these terms is illustrated in Figure 1 (a).

- Wrong_Forwarding: This input variable denotes the percentage of packets forwarded to the wrong destination. Four fuzzy sets represented by the linguistic terms: reliable, nominal_reliable, poor_reliable and unreliable are identified for this variable. The membership function WF that represents these terms is shown in Figure 1 (b).

- Fabrication: This input variable corresponds to the number of faulty routing messages fabricated by a node. The linguistic terms reliable, nominal_reliable, poor_reliable and unreliable represent the four fuzzy sets of this input variable. Figure 1 (c) shows the membership function F, which represents the fabrication input variable.

- Replay_Attack: This input variable stands for the number of replay attacks launched by a node. A set of four linguistic terms: reliable, nominal_reliable, poor_reliable and unreliable is defined over this variable. These terms are illustrated using the membership function RA as shown in Figure 1 (d).

- Trust_level: The output variable that represents the trust level of a node is defined as trust_level. This variable has seven fuzzy sets represented by the linguistic terms: not_trusted, low_trust, acceptable_trust, average_trust, trustable, highly_trusted, and fully_trusted. These fuzzy sets are defined by the membership function T and are shown in Figure 2.

3.3 Creation of Rules

The proposed fuzzy trust evaluation model is a Mamdani type with four input and one output variables. The Mamdani type fuzzy inference is used for its simplicity. This type expects the output membership functions to be fuzzy sets. The elements of a fuzzy set are mapped by membership functions to a value, which defines the degree to which a fuzzy variable is a member of a set. The membership functions μ(P), μ(WF), μ(F), μ(RA), μ(T), map the input variables, packet_dropped, wrong_forwarding, fabrication and replay_ attack, and the output variable, trust_level, into the interval (0,1) respectively.

After the fuzzification step produces a set of fuzzy inputs, these inputs can then be processed using rule evaluation. Fuzzy rules in the fuzzy trust evaluation model are 'if-then' statements that describe the action to be taken in response to various fuzzy inputs. The following is one of the fuzzy rules in the proposed model:

if *P is nominal_reliable and WF is reliable and F is reliable and RA is reliable* ***then*** *T is highly_trusted*

The fuzzy operator AND is used to obtain a single number that represents the result of the antecedent evaluation. This number is then applied to the trust_level membership function.

3.4 Defuzzification Method

When each rule is evaluated, the minimum numeric value of its antecedents is assumed to be the rule strength. However, when determining the trust level output fuzzy label, the maximum numeric value is taken to be the label's value. The input of the aggregation process is the list of truncated output functions returned by the implication process of each rule. The output of the aggregation process is one

fuzzy set for each output variable. The numeric values corresponding to the output fuzzy labels are calculated as the maximum truths of various rule strengths fed in from the rule evaluation step.

The result of the fuzzy logic inference is the value of a linguistic variable. In the proposed model a possible inference result could be "trustable". The conversion of such a linguistic result to a real value which represents the trust level of a node is called defuzzification. Therefore, the input for the defuzzification process is a fuzzy set and the output is a single number. The aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single trust level for the node. The defuzzification method used in this thesis is the Centre-of-Maximum (CoM) method, which is essentially a centroid calculation.

4 Overview of the Fuzzy Trust Algorithm

In our work, the main focus surrounds on-demand routing protocols, where the route is discovered only when a node wants to send data to another node. The routing protocol used in this study is the AODV protocol. When a node wants to send data to another node, it broadcasts a Route Request (RREQ) packet to all its neighbours. The RREQ propagates through the network until it reaches the destination or a node with a fresh enough route to the destination. Forwarding of RREQs is done when the node receiving a RREQ does not have a route to the destination. It then rebroadcasts the RREQ. This process is repeated until the RREQ reaches the destination which sends a Route Reply (RREP) back to the sender. When a node detects that a route to a neighbour is no longer valid, which may be caused by a link break, it removes the routing entry and sends a Route Error (RERR) message to the neighbours that are actively using the route, informing them that this route is no longer valid. This procedure is repeated until the message reaches the source where it either stops sending data or requests a new route by generating a new RREQ. A detailed description of this protocol can be found in [18].

In the proposed Fuzzy Trust Algorithm (FTA), each route has a trust level. The route trust level is determined on the basis of the node which has the lowest trust level in that route. The main goal of FTA is to choose the most reliable route between the source and the destination. This is achieved by choosing the route with the highest trust level between the source and the destination nodes. In other words, the route with the highest trust level is comparably the most secure route.

When a source node S desires to transmit a data packet to a destination node D, S must acquire the next hop node along the path to D. If this information is not readily available then route discovery is performed on demand. In a typical Ad-hoc situation, there are R1, . . . , Rn, totally n possible routes from the source S to the destination D. In each route there exist an x number of relay nodes $n_1, \ldots, n_j, \ldots, n_x$ to help in forwarding the packets from S to D.

After applying the fuzzy trust evaluation model each node will have a trust level. Each node is assumed to be able to evaluate the trust level of each of its neighbouring nodes based on the information regarding the behaviour history of these nodes. These trust levels are then used to determine the most appropriate route between S and D. Suppose the current trust level of the j^{th} node in the i^{th} route which is evaluated using the fuzzy trust evaluation model is T_{ij}, then the trust level of the i^{th} route is defined as the minimum trust level of all the nodes that are included in the i^{th} route:

$$(\text{trust level})_i = \min T_{ij},\ j \in (1, \ldots, x).$$

FTA utilises the trust levels to choose the most reliable route between the source node S and the destination node D. According to the AODV routing protocol, the source node S can receive more than one reply in a period of time after sending a RREQ. Those routes from S to D will all include a trust level value. The route with the maximum value of the trust level is then selected. As a result, the desired route "k" can be obtained as the route with the maximum trust level:

$$(\text{trust level})_k = \max (\text{trust level})_i ,\ i \in \{R1, R2, \ldots, Rn.\}$$

4.1 The Route Discovery using Fuzzy Trust Levels

The FTA is based on a source-initiated on-demand routing protocol, so nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges. This type of routing creates routes only when requested by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible routes trust levels have been examined. Once a route has been established, it is maintained by a route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer desired [13]. The FTA uses the following fields with each routing table entry:

- Destination IP Address.
- Destination sequence number: the sequence number is used to ensure that the routes are loop free and that if the intermediate nodes reply to RREQ, they reply with the latest information only.
- Valid destination sequence number flag: this flag is used to indicate whether the sequence number is known or not. This is important when updating route table entries and creating RREQ. For example, if the sequence number for a particular destination is not known and a RREQ is needed, the node must set the unknown sequence number bit in the RREQ. Otherwise the value of the sequence number field could affect the sequence number at other nodes that receive the RREQ, because of the way sequence numbers are compared.
- Trust level: this value corresponds to the minimum trust level of all nodes in the route.
- Hop count: this is the number of hops needed to reach the destination.
- Next hop.
- Lifetime (expiration or deletion time of the route).

When S wants to send a message to D, and does not already have a valid route to that destination, it initiates a path discovery process to locate other nodes. The source node S propagates a RREQ to its neighbours. The RREQ packet includes:

- The IP address of D.
- The sequence number of D.
- Trust level (the minimum trust level of all nodes in the current found route).
- Hop count.
- Lifetime.

The destination sequence number field in the RREQ message is the last known destination sequence number for this destination and is copied from the destination sequence number field in the routing table. If no sequence number is known, the unknown sequence number flag must be set. The trust level field is equal to the source node's trust level. The hop count field is set to zero. When a neighbour node receives the RREQ packet, it will be forwarded if it matches some conditions.

When an intermediate node receives the RREQ from its neighbour, it first increases the hop count value in the RREQ by one. This is to account for the new hop through the intermediate node if the packet is not going to be discarded. The originator sequence number contained in the RREQ must be compared to the corresponding destination sequence number in the routing table. If the originator sequence number of the RREQ is greater than the existing value, the intermediate node compares the trust level contained in the RREQ to its current trust level to get the minimum. The intermediate node then updates the trust level of RREQ with the minimum. At this stage, the updated trust level of the RREQ is the trust level of the route. If the originator sequence number contained in the RREQ is greater than the existing value in its routing table, the relay node creates a new entry with the sequence number of the RREQ. If the originator sequence number contained in the

RREQ is equal to the existing value in its routing table, the trust level of the RREQ must be compared to the corresponding trust level in the routing table. In the case that the trust level contained in the RREQ is greater than the trust level in the routing table, the relay node updates the entry with the information contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record the addresses of neighbours from which the first copy of the broadcast packet was received in their routing tables. This in turn establishes a reserve path. If additional copies of the same RREQ are received later, these packets will be discarded

Once the RREQ reaches the destination D or an intermediate node with a valid route to D, the destination or intermediate node generates a Route Reply (RREP) packet and unicasts it back to the neighbour from which it received the RREQ. In the case where the generating node is the destination itself, it must update its own sequence number to the maximum of its current sequence number and the destination sequence number in the RREQ packet originating the RREP. The destination node places its sequence number into the destination sequence

number field of the RREP and enters the value zero in the hop count field of the RREP. When generating a RREP message, a node copies the destination IP address, the originator sequence number and the trust level from the RREQ message into the RREP message.

When an intermediate node receives the RREP from its neighbour, it first increases the hop count value in the RREP by one. As the RREP is forwarded back along the reverse path, the hop count field is increased by one at each hop. Thus, when the RREP reaches the source, the hop count represents the distance, in hops, of the destination node D from the source node S. The originator sequence number contained in the RREP must be compared to the corresponding destination sequence number in the routing table entry. If the originator sequence number of the RREP is greater than the existing value, the node compares the trust level contained in RREP to its current trust level to get the minimum, and then updates the trust level of RREP with that minimum. This minimum value represents the trust level of the route. The intermediate node creates a new entry with the destination sequence number of RREP and marks the destination sequence number as valid in two situations:

- If the sequence number in the routing table entry is marked as invalid.
- If the destination sequence number in the RREP is greater than the node's copy of the destination sequence number.

The trust level field in the routing table entry is set to the trust level contained in the RREP. If the originator sequence number contained in the RREP is equal to the existing destination sequence number in the node's routing table, the entry of this sequence number is updated with the information contained in the RREP. In this case, the trust level in the intermediate node's routing table is set to the trust level in the RREP.

The next hop in the route entry is assigned to be the node from which the RREP is received, which is indicated by the source IP address field in the IP header. The current node can use this route to forward data packets to the destination.

4.2 Route Maintenance

Similar to the AODV routing protocol, a node uses a 'hello' message which is a periodic local broadcast by a node to inform each mobile node in its neighbourhood to maintain the local connectivity. A node should only use the 'hello' messages if it is part of an active route. The way it works is as follows. If a node has previously received a 'hello' message from one of its neighbours, and later for some reason has not received any packets (either 'hello' or data packets) from this node for a certain time, the node should assume that the link to this neighbour is now lost. When this happens, the node will send a Route ERRor (RERR) message to all predecessors indicating which link has failed. Then the source initiates another route search process to find a new path to the destination or start the local repair. The flowchart of the proposed FTA approach is shown in Figure 3.

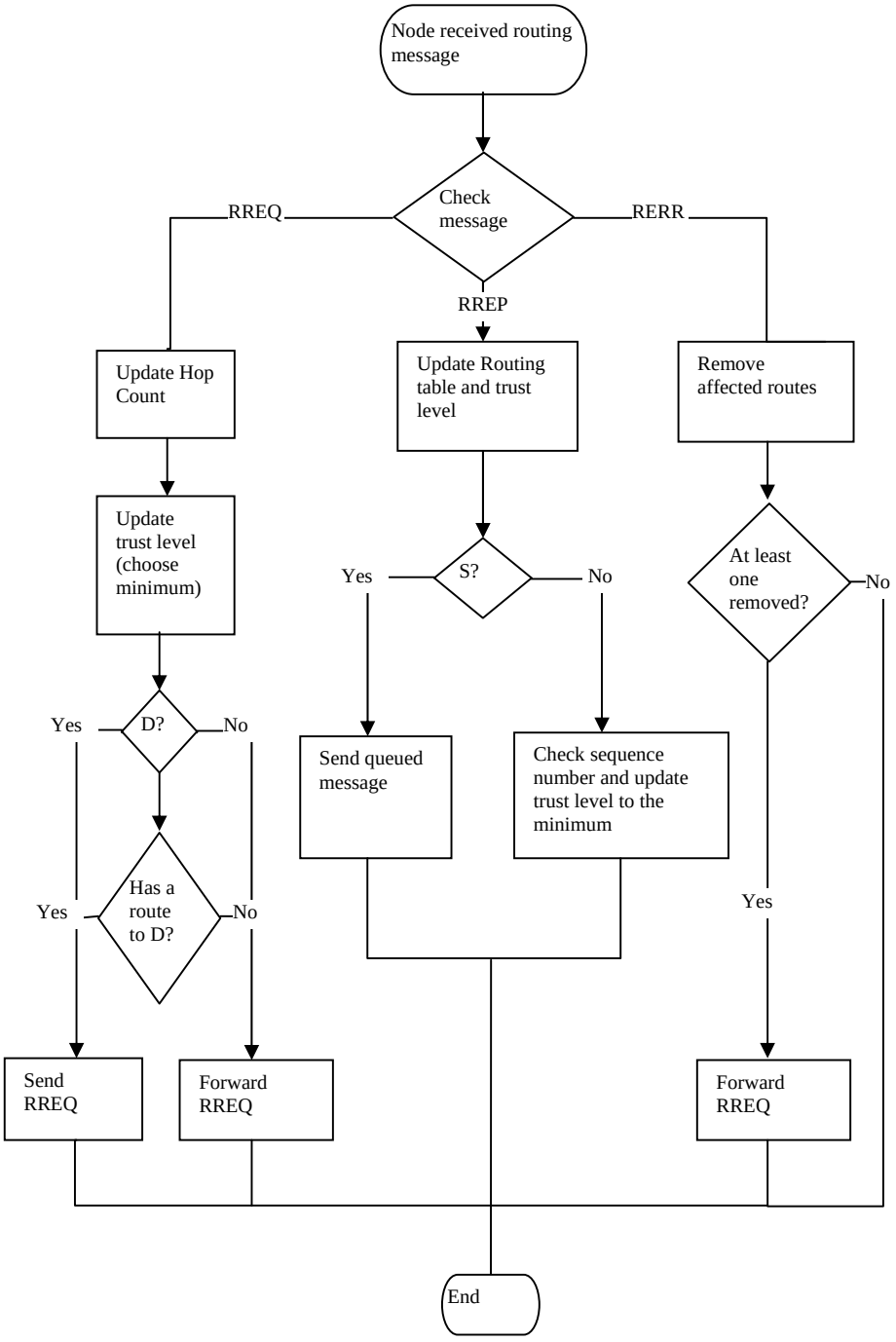

Figure 3 FTA flowchart

5 Integration of MATLAB and OPNET

The integration of MATLAB's fuzzy logic toolbox with OPNET can facilitate the evaluation of trust levels of the nodes of Ad-hoc networks. In this thesis, MATLAB and OPNET are interfaced so as to use the fuzzy trust evaluation model, which was developed in MATLAB, in the simulation conducted in OPNET. Output results from this model are passed to OPNET during the simulation of the Ad-

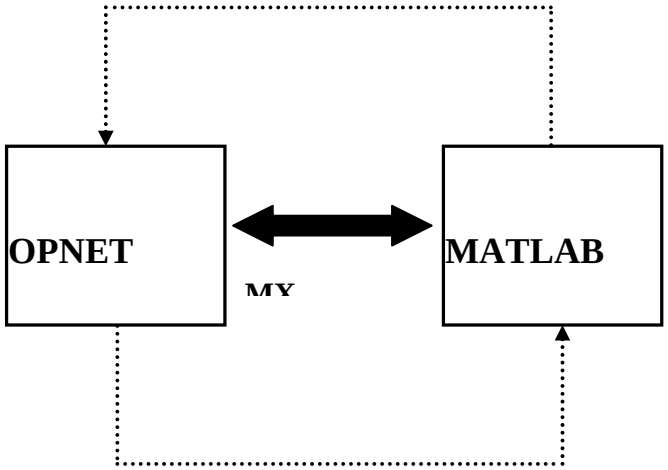

Figure 4 OPNET and MATLAB interface

hoc network. This output is then used in the routing decision making process.

For interfacing OPNET and MATLAB, the MX interface is used as seen in Figure 4. This interface is provided by MATLAB, which allows programs in OPNET to call functions developed in MATLAB. This is achievable by following the steps which are described below.

- MATLAB engine is started from OPNET. As a result of this it will be possible to work with the MATLAB command window using OPNET. To achieve this, the following files are included in the bind_shobj_libs environment attribute in OPNET: libmat.lib, libeng.lib, libmex.lib, libmx.lib
- The directory where the above files are present is included in bind_shobj_flags. After including the necessary files into the include path, the MATLAB engine can be started by OPNET at the beginning of the simulation using the function engOpen(). This provides the OPNET simulator with a pointer to a memory location that can be used to pass commands to the MATLAB engine. The engine pointer can be shared among different processes by declaring the engine pointer in a header file common to all process models.
- The workspace in MATLAB is set up by passing the inputs from OPNET. Variables can be exchanged between OPNET and MATLAB using functions like engPutArray() and engGetArray(), engOutputBuffer(), and engEvalString().
- After passing the variables to MATLAB, OPNET can execute any desired function in MATLAB by simply calling it.
- After executing functions in MATLAB, outputs are transferred to OPNET using engGetArray().

6 Simulation Study Setup

The simulation is carried out using OPNET Modeler V11.5 OPNET Modeler is used to construct models for two different purposes: to study system behaviour and performance; and to deliver a modeling environment to end users [17] A network model may contain any number of communicating entities called nodes. Nodes are instances of node models; developed using the Node Editor. Network models consist of nodes and links that can be deployed within a geographical context. Node models consist of modules and connections [11].

Each simulation scenario consists of fifty nodes. The channel speed of the wireless LAN is set to 11 Mbps. The routing protocol used in the simulation is the AODV protocol. Figure 5 shows a snapshot of the simulation setup.

To study the effects of the presence of malicious nodes in Ad-hoc networks, three performance metrics will be measured for a number of scenarios and situations. These are the throughput, the round-trip delay, and the packet loss rate. The total measured throughput is considered as the average amount of data payload transmitted and received over a period of time between two nodes. It is measured in Mbps. The packet loss percentage at nodeX for transmission between nodeX and nodeY describes the percentage of packets transmitted from nodeX over the network that did not reach nodeY. The round-trip delay refers to the average time taken by a packet to complete one full trip from source to destination and back and is measured in msec.

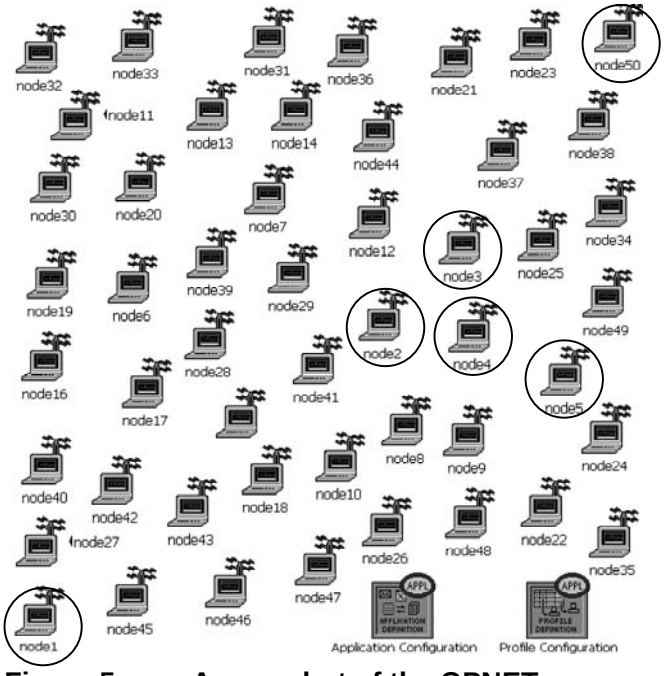

Figure 5. A snapshot of the OPNET simulation setup

In order to facilitate the comparisons between the different approaches, all performance parameters are combined into one indicative index. The Overall Performance Index (OPI) is calculated as the weighted sum of the three performance metrics that have been considered so far. The sum of the weights $w_t + w_{pl} + w_d$ is equal to 100%. The OPI is defined using the following formula:

$$OPI = w_t * Throughput\_Ratio + w_{pl} * Packet\_Loss\_Ratio + w_d * Round\_Trip\_Delay\_Ratio$$

where w_t, w_{pl}, and w_d are the weights corresponding to the throughput, the packet loss rate and the round trip delay metrics respectively. Throughput_Ratio, Packet_Loss_Ratio, and Round_Trip_Delay_Ratio are the ratio of the measured values to the nominal values. Distributing the weights between the three performance metrics can differ from one application to another. For example, packet loss has a higher impact on audio and video based applications than the throughput and the round trip delay. However, it is well known that the packet loss usually has more effect on the performance of Ad-hoc networks. Packet loss results in packet retransmissions which reduces throughput and increases round trip delay between nodes. Therefore, the weight for the packet loss parameter has been chosen to be twice that of the throughput and the round trip delay. As a result of that the weights are distributed as follows: $w_t = 25$, $w_d = 25$, $w_{pl} = 50$.

The simulation studies consist of a number of scenarios replicating practical situations. Each scenario runs in five different situations. In the first situation, none of the fifty nodes of the Ad-hoc network acts maliciously. In the second situation, five nodes chosen randomly out of the fifty nodes are acting maliciously. In the third situation, ten malicious nodes are present. In the fourth situation, fifteen nodes act as malicious nodes. In the fifth situation, twenty out of the fifty nodes are malicious nodes.

To facilitate convenient assignment of any node as a malicious one, a Boolean parameter has been implemented to define a node as a malicious node. It can be set or reset. Using this implementation capability, it is straight forward to set up a different number of malicious nodes. The malicious nodes are implemented in four different ways. Some malicious nodes drop packets based on the simulation time (for example dropping all packets when the simulation time is between 50 and 100 sec). Other malicious nodes forward some of the packets to the wrong destinations. Some other malicious nodes fabricate and broadcast false routing messages. Other malicious nodes launch replay attacks.

Also, to study the effect of nodes mobility on the performance of Ad-hoc networks, all nodes move randomly 60 sec after the start of each simulation with a speed of 10 m/s. The rationale behind waiting for 60 seconds before the nodes start to move is to give them a reasonable time to establish their routing tables. Nodes move for 20 sec, pause at their destination for 60 sec and move back to their original locations.

Four scenarios are applied in the evaluation of the FTA approach. In these scenarios node1 sends traffic to node50 using other nodes as relay nodes. Simulations here can be summarised as follows:

- In the first scenario, node1 sends TCP traffic to node50 through other nodes that are acting as relay nodes. All nodes in this scenario are stationary nodes.

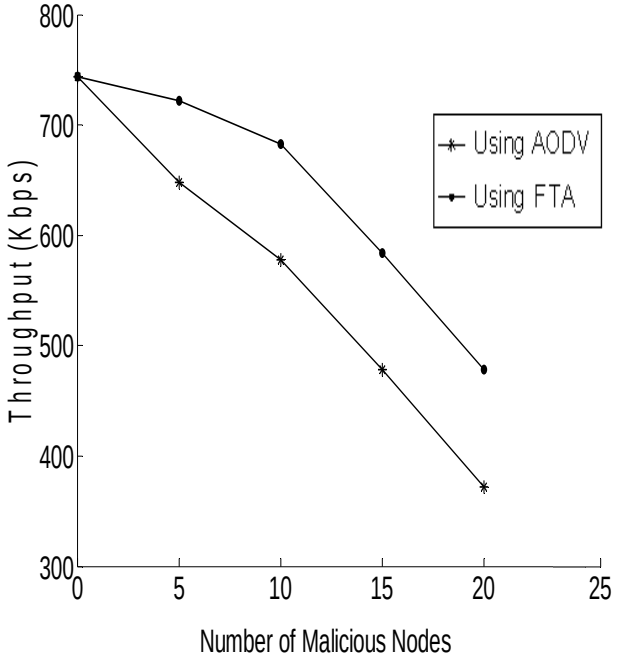

Figure 6 Throughput comparison for the first scenario between AODV and the proposed FTA approach

- In the second scenario node50 receives TCP traffic generated and sent from node50 through other nodes that are acting as relay nodes. All nodes are moving according to the trajectory described in the previous section.

To check the effect of the transport layer protocol used between the communicating benign nodes on the performance of the Ad-hoc network, the same scenarios are repeated when the communicating benign nodes send UDP data traffic. Therefore:

- In the third scenario node50 receives UDP traffic sent from node1, using some nodes which are acting as routers forwarding packets to the destination node. In this scenario all nodes are motionless.

- In the fourth scenario node1 sends traffic to node50. All nodes are moving according to the trajectory defined in the previous section.

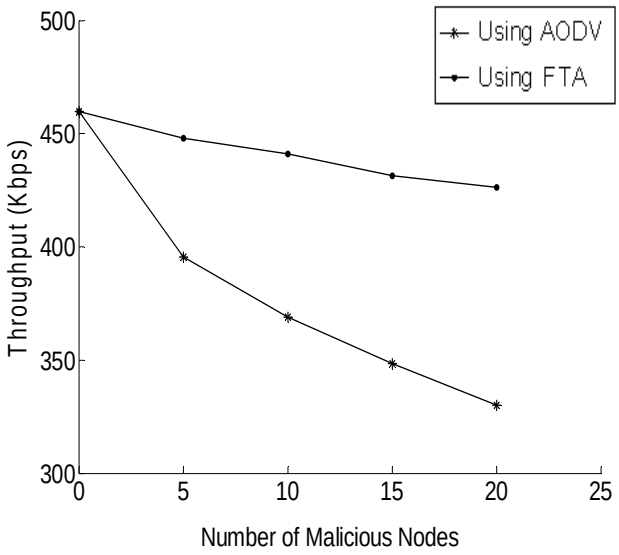

Figure 7 Throughput comparison for the second scenario between AODV and the proposed FTA approach

7 Collected Results and Analysis

A detailed analysis of an Ad-hoc network simulation results after applying the FTA approach are presented in this section. The variations of the throughput, round trip delay and packet loss are analysed individually. In most cases, the performance results of the evaluations metrics are plotted as graphs for easy comparison and quick reference. All simulations run for five minutes and the results are the average of repeating each simulation ten times.

7.1 Throughput Measurements

The results of the throughput measurements after applying the new FTA approach are reported here. In Figure 6 to Figure 10, the number of malicious nodes is plotted against the throughput for the first, second, third, and fourth scenarios respectively. These graphs show both situations before and after applying the proposed FTA approach. These simulations are carried out with the number of malicious nodes varying from nil to 40% of the total number of nodes.

These graphs show that the proposed FTA approach can achieve up to 30% improvement in the throughput over the AODV protocol. This can be described by noting that the number of malicious nodes existing in the route between the

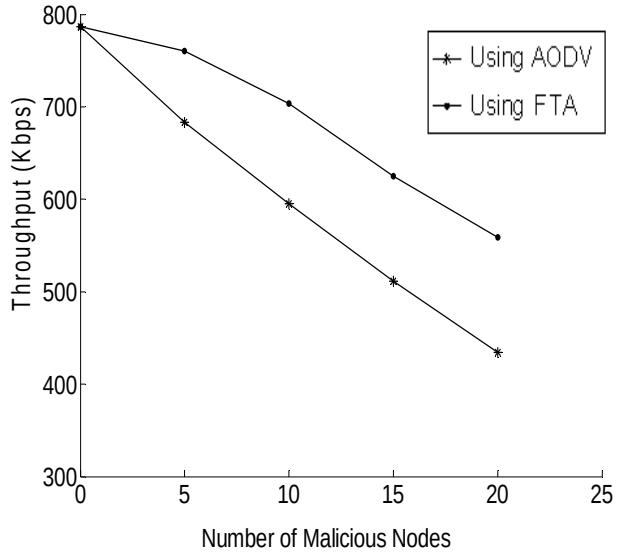

Figure 9 Throughput comparison for the third scenario between AODV and the proposed FTA approach

communicating benign nodes is less in the FTA approach compared to AODV. This can be due to the fact that with more malicious nodes existing in the route, data from source to destination are more vulnerable to attacks, causing the deterioration of network performance.

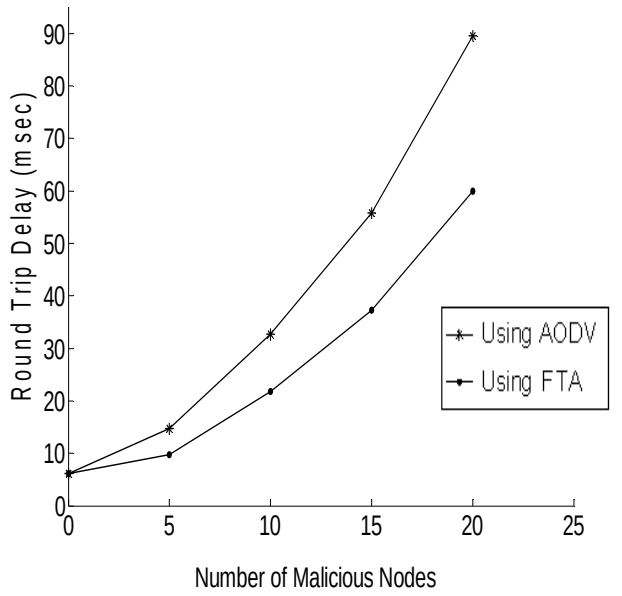

Figure 10 Round trip delay for the first scenario between AODV and the proposed FTA approach

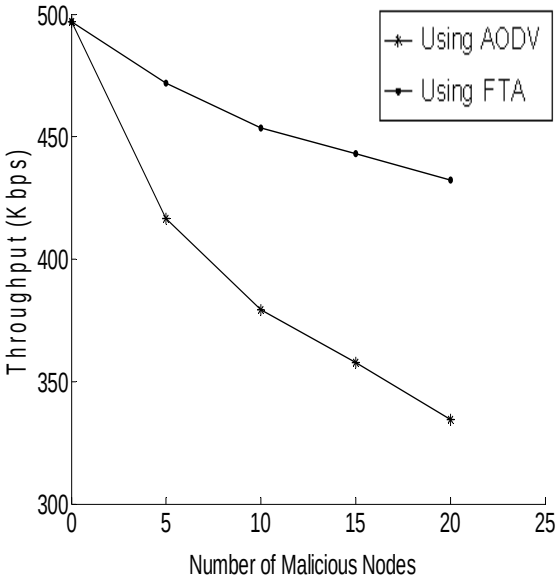

Figure 8 Throughput comparison for the fourth scenario between AODV and the proposed FTA approach

From these graphs, it is also evident that the improvements in the throughput values that can be achieved after applying the FTA approach are more pronounced when the network contains a larger number of malicious nodes. This can be explained by the fact that, as the number of malicious nodes increases, the number of reliable routes decreases. With five or more malicious nodes, however, reliable routes become rare, making it extremely likely to encounter a malicious node on the path. For instance, when a route consists of six nodes and 40% of the nodes are acting maliciously, then the probability that any route does not contain more than one malicious node is: $(0.6)^4 = 0.1296$ which means that only one out of eight routes is reliable.

It is also noticeable from these graphs that the throughput is lower when the nodes are mobile. The main reason behind this is the reinitiating of the routing process caused by the link break between nodes. In general, when nodes are mobile, the number of link changes increases causing link breakages and communications disruption [16].

7.2 Round Trip Delay Measurements

This section analyses the round trip delay measurement between communicating benign nodes

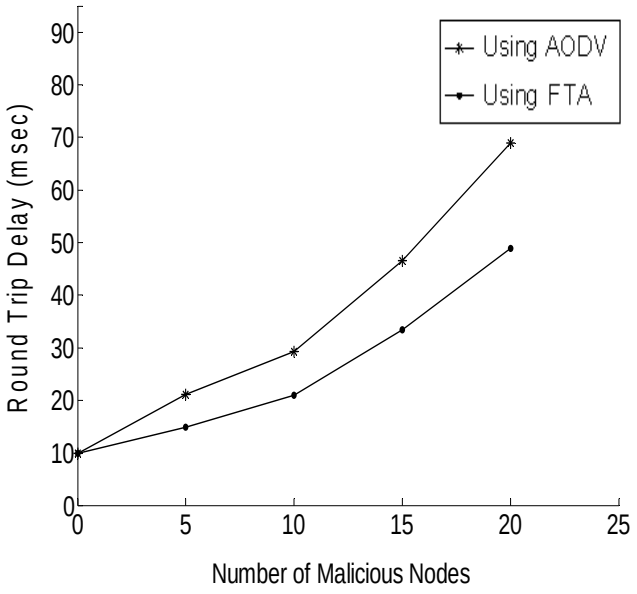

Figure 11 Round trip delay comparison for the fourth scenario between AODV and the proposed FTA approach

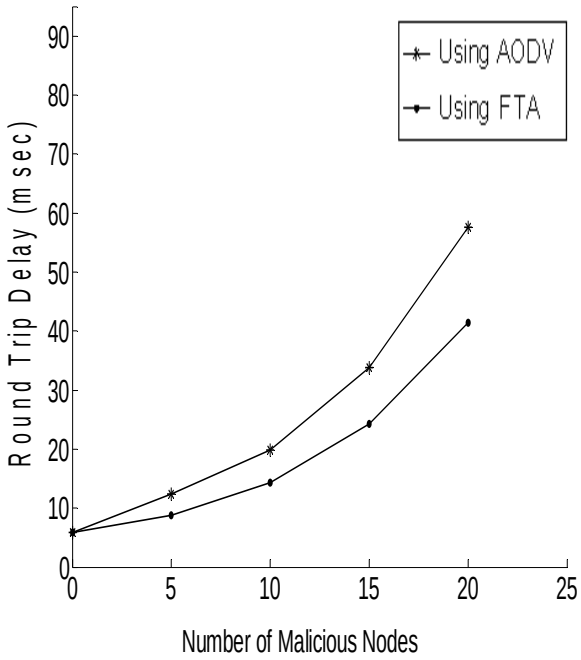

Figure 12 Round trip delay comparison for the third scenario between AODV and the proposed FTA approach

after applying the FTA approach. In this thesis, the round trip delay measurement is considered as the average time taken to complete one full trip from source to destination and back. The graphs in Figure 9 to Figure 11 show the round trip delay variations This simulation is done with the number of malicious nodes changing from 0 to 20 nodes. It is clear from these graphs that when applying the FTA approach the average time for a given packet to complete a full round trip between node1 and node50 is relatively lower. This applies to all scenarios. For instance, in Figure 13, when 40% of the nodes are acting maliciously, the round trip delay decreased to 59 msec after applying the FTA approach. This is compared to 89 msec when nodes use AODV protocol. As mentioned in the previous section, the main reason for this behaviour is that the new route between source and destination has either no, or less malicious, nodes. When using the AODV protocol, as the number of malicious nodes increases, for the first, second, third, and fourth scenarios respectively. These graphs also show the situations before and after applying the FTA approach.

the total expected area covered by their radios increases and the likelihood of even a single reliable route existing decreases. On the other hand, by using the trust levels of the Ad-hoc network nodes, the FTA approach is able to find more reliable routes.

From these graphs it can also be noted that, as the number of malicious nodes gets higher, the improvements in the round trip delay after applying the FTA approach are more achievable. This is mainly due to the fact that the higher the percentage of malicious nodes, the higher the probability that these nodes will participate in the route between the benign nodes. This can lead to more route request messages being dropped, causing a delay at the sending node.

It can also be noted that the round trip delay is lower when the nodes are stationary. This can be explained by noting that when moving, nodes will lose connections with their neighbours, causing delay at the sending node in order to reinitiate the routing process. Also as the nodes move, the channels change so rapidly that it is hard to perform channel estimation. Moving nodes make the distinction between near and far nodes blurred, causing transmissions to interfere with each other [10].

7.3 Packet Loss Rate Measurement

The analysis presented in this section discusses the results of the packet loss rate after applying the FTA approach. The results in TABLE I show the packet loss rate values for the first, second, third, and fourth scenarios when 40% of the nodes are acting maliciously. As in the previous measurements, the results cover situations both before and after applying the proposed fuzzy trust based approach. It is noticeable here that there is a relatively higher packet loss rate experienced with the AODV protocol for all scenarios. For instance, the packet loss rate has decreased to 38% after applying the FTA approach, compared to 52% when nodes use AODV. The argument and explanation provided in the previous sections hold here. The decrease in the packet loss when using the FTA approach can be credited to the fact that the new route between the source and the destination has no, or less, malicious nodes. As a malicious node starts to launch attacks, its trust level becomes lower. Therefore, it is less likely to participate in the route between the communicating nodes and disrupt the operation of the network.

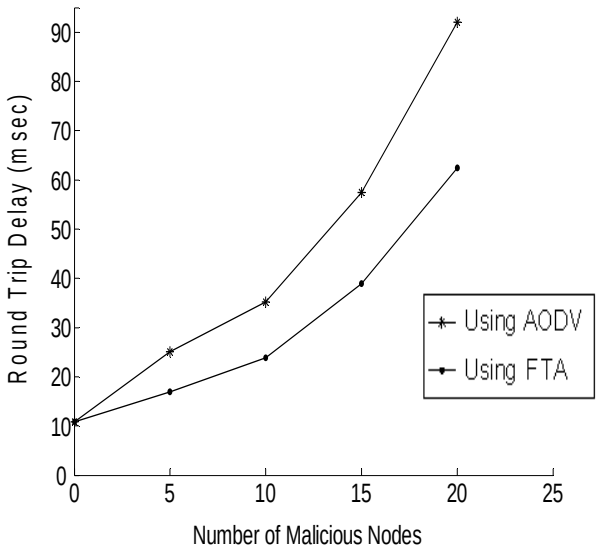

Figure 13 Round trip delay comparison for the second scenario between AODV and the proposed FTA approach

TABLE I Packet loss comparison for the first, second, third, and fourth scenarios after

	Using AODV	Using FTA
First Scenario	51%	39%
Second Scenario	57%	45%
Third Scenario	44%	31%
Fourth Scenario	52%	38%

It can also be noted that the packet loss rate is lower when the nodes are motionless. This can be attributed to the fact that packets are dropped when connections are lost between moving nodes. As the mobility of nodes increases, the topology changes in the network become more frequent. This causes a decrease in the accuracy of the routing information maintained by the routing protocol [14]. Therefore, the packet loss rate shows a slow increase as the mobility of the nodes increases. In summary, it can be concluded that the decrease in performance is mainly due to communication failures which arise more frequently when nodes are moving.

These results also clearly show that as the number of malicious nodes in the network increases, the improvements in the packet loss rate that can be achieved after applying the FTA approach are more significant. This is mainly due to the fact that the higher the percentage of malicious nodes, the higher the probability that these nodes will drop the routing data messages, leading to a higher loss rate. With a high number of malicious nodes and without using the fuzzy trust evaluation approach, the percentage of successfully established routes decreases.

7.4 Overall Performance Index Comparison

The main goal of using the OPI is to facilitate the comparison between AODV and the proposed FTA

approach. As stated in Section 4, the Overall Performance Index is defined as a weighted sum of the throughput, round trip delay and packet loss parameters. TABLE II shows a comparison of the performance index before and after applying the FTA approach.

These values clearly show the improvement in the Overall Performance Index that is achieved after applying the FTA approach. For instance, for the eleventh scenario and when 20 malicious nodes are present in the network, the OPI indicates a nearly 19% improvement.

8 Conclusions and Future Work

This paper has highlighted the importance of using trust levels to improve the reliability and performance of Ad-hoc networks. Evaluating trust levels between nodes of Ad-hoc networks poses a big challenge due to the lack of infrastructure in Ad-hoc networks. To overcome this limitation, a new approach based on fuzzy logic concepts is proposed to facilitate the evaluation of trust levels between nodes of Ad-hoc networks. Simulation and experimental results collected after applying the FTA approach show significant improvements in the performance and the reliability of Ad-hoc networks in the presence of malicious nodes. For instance, the OPI for the fourth scenario improved by 18.91% after applying the fuzzy trust based approach.

However, a number of further investigations could be conducted to extend this approach. As stated in Section 2, human beings make many trust-based decisions on a subconscious level. Incorporating concepts similar to the way humans think into the FTA approach has the potential to further facilitate the evaluation of trust levels. Artificial Neural Networks for instance are used to perform tasks similar to those performed by human brains. The learning capability of Artificial Neural Networks made them a prime target for combination with fuzzy based systems in order to automate or support the developing process of such systems. Therefore, a future research direction would be to take advantage of the learning capability of Artificial Neural Networks by combining ideas and concepts evolving from such networks with the fuzzy trust based approach.

TABLE II Overall Performance Index comparison for the first, second, third, and

	Using AODV	Using FTA
First Scenario	45.54	59.66
Second Scenario	52.5	67.59
Third Scenario	46.62	64.32
Fourth Scenario	54.08	72.99

Acknowledgment

We would like to thank OPNET for their kindness in providing us with Modeler software license, which has greatly assisted in finalizing this paper. We also would like to thank Cisco for the generous scholarship which has permitted us to move forward with our studies.

References:

[1] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei, "Reputation and Trust-Based Systems for Ad-hoc and Sensor Networks," *Algorithms and Protocols for Wireless Ad-hoc and Sensor Networks, A. Boukerche (ed.), Wiley & Sons*, 2007.

[2] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, 2005, vol. 47, pp. 445-487.

[3] C. Altrock, *Fuzzy Logic and Neuro-Fuzzy Applications Explained,* Prentice-Hall 1995.

[4] V. Balakrishnan, V. Varadharajan, U. K. Tupakula, and P. Lucs, "Trust and Recommendations in Mobile Ad-hoc Networks," In *Proc. of the 3rd Int. Conf. on Networking and Services (ICNS 07), Athens, Greece,* 2007, pp. 64-69.

[5] E. Barka, "On The Impact of Security on the Performance of WLANs," *Journal of Communications*, 2007, vol. 2, pp. 1-11.

[6] T. Fowler, "Mesh Networks for Broadband Access," *IEE Review*, 2001, vol. 47, pp. 17-22.

[7] H. Hallani and S. A. Shahrestani, "Utilizing Behaviour History to fight malicious nodes in Wireless Ad-hoc Networks," In *Proc. of the 8th*

International Business Information Management Association (IBIMA) Conference, Dublin, Ireland, June, 2007, pp. 84-89.

[8] H. Hallani and S. A. Shahrestani, " Wireless Ad-hoc Networks: Employing Behaviour History to Combat Malicious Nodes," In *Proc. of the 1st International Conference on Signal Processing and Telecommunication Systems (ICSPCS'07), Gold Coast, Australia*, December, 2007, pp. 1-6.

[9] Y. Hu, A. Perrig, and D. Johnson., "Ariadne: A secure on-demand routing protocol for Ad-hoc networks," *Wireless Networks,* 2005, vol. 11, pp. 21-38.

[10] S. A. Jafar, "Too Much Mobility Limits the Capacity of Wireless Ad-hoc Networks," *IEEE Transactions on Information Theory* 2005, vol. 51, pp. 3954-3964.

[11] X. Liu, "Application of OPNET in the Network Project and Design," *CONTROL AND AUTOMATION*, 2006, pp. 104-106.

[12] C. Mallett, W. Millar, and H. Beane, "Perspectives on Next Generation Mobile," *BT Technology Journal*, 2006, vol. 24, pp. 151-160.

[13] N. Meghanathan, "A Simulation Study on the Stability-oriented Routing Protocols for Mobile Ad-hoc Networks," In *Proc. of the Int. Conf. on Wireless and Optical Communications Networks*, 2006, pp. 1-5.

[14] N. Moghim, F. Hendessi, N. Movehhedinia, and T. A. Gulliver, "Ad-hoc Wireless Network Routing Protocols and Improved AODV," *Arabian Journal for Science and Engineering*, 2003, vol. 28, pp. 99-114.

[15] S. Nefti, F. Meziane, and K. Kasiran, "A Fuzzy Trust Model for E-Commerce," In *Proc. of the 7th IEEE Int. Conf. on E-Commerce Technology (CEC 05)* 2005, pp. 401-404.

[16] P. C. Ng and S. C. Liew, "Throughput Analysis of IEEE802.11 Multi-Hop Ad-hoc Networks," *IEEE/ACM Transactions on Networking*, 2007, vol. 15, pp. 309-322.

[17] OPNET Modeler, "http://www.opnet.com/."

[18] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," In *Proc. of the Mobile Computing Systems and Applications*, 1999, pp. 90-100.

[19] F. Qiu and J. R. Jensen, "Opening the black box of neural networks for remote sensing image classification," *International Journal of Remote Sensing*, 2004, vol. 25, pp. 1749-1768.

[20] J. P. Shim, U. Varshney, S. Dekleva, and G. Knoerzer, "Mobile and Wireless Networks: Services, Evolution & Issues," *IEEE International Journal of Mobile Communications Magazine*, 2006, vol. 4, pp. 405-417.

[21] O. K. Tonguz and G. Ferrari, *Ad-Hoc Wireless Networks: A Communication - Theoretic Perspective*, John Wiley & Sons, 2006.

[22] X. Yang and N. Vaidya, "Priority Scheduling in Wireless Ad-hoc Networks," *Wireless Networks*, 2006, vol. 12, pp. 273-286.