Analytical Model and Performance Evaluation for the TRIDNT protocol

Ahmed M. Abd El-Haleem[1], Ihab A. Ali[2], and Ibrahim I. Ibrahim[3]

**[1] Assistant Lecture, Communication Department, Faculty of Engineering, Helwan University
Helwan, Egypt**

**[2]Associate Professor, Communication Department, Faculty of Engineering, Helwan University
Helwan, Egypt**

**[3]Professor, Communication Department, Faculty of Engineering, Helwan University
Helwan, Egypt**

Abstract

In Mobile ad-hoc networks, nodes must cooperate to achieve the routing functions. Node misbehavior due to selfish or malicious intention could significantly degrade the performance of MANET because most existing routing protocols in MANET are aiming at finding the most efficient path. A Two node-disjoint Routes scheme for Isolating Dropper Node (TRIDNT) protocol has been proposed in [13]. The protocol uses an incentive mechanism for selfish node to declare its selfishness behavior and also use two node-disjoint routes to reduce the malicious searching time.

In this paper, we give an analytical model and performance evaluation for the TRIDNT protocol. First we calculate the time taken to detect the malicious node in the routing path. Second we will drive an expression for the expected attempt time until finding a misbehaving free route, and calculate the probability of failing in finding a misbehaving-free route in case of limited number of attempts. Finally, the connection request blocking probability and per flow throughput are calculated. The performance of the proposed TRIDNT protocol is compared with the TWOACK and Muhammad Zeshan protocols. The results indicate that under a low threshold and low traffic intensity conditions the TWOACK and Muhammad Zeshan protocols have a malicious searching time little greater than that of TRIDNT, but with high probability of false reporting of legitimate nodes as misbehaving nodes due to small threshold value and without differentiating between selfish node and malicious nodes. The expected attempt time of TRIDNT protocol have the smallest value because it allows a controlled degree of node selfishness which gives the selfish node the incentive to declare itself and reduce the path diagnosis time, and save the misbehaving detection time. Also TRIDNT protocol has a smaller limited-attempts-based connection blocking probability than TWOACK and Muhammad Zeshan protocols, but has a higher connection request blocking probability than the other two protocols. Finally because per flow throughput is inversely proportional to the expected attempt time, so TRIDNT has the highest per flow throughput. Then we say that TRIDNT protocol can find and isolate the malicious node in small amount of time without using of promiscuous listening, which results in improving the per flow throughput and improving the overall throughput performance of MANET.

Keyword: ***Analytical model, Trust-Based routing Performance evaluation, Ad Hoc Network, Secure Routing Protocol, network security.***

1. Introduction

A wireless Ad Hoc network is a multi-hop self-organized mobile network where nodes exchange data without the need for an underlying infrastructure. Each node of this network has the function of terminal and router responsible for relaying packets to other nodes. Some packets can be delivered from a source node to a destination node by way of various intermediate nodes, thereby maintaining network connectivity and applicability of MANET depends heavily on cooperation between nodes in such a dynamic environment. Due to openness of MANET, nodes moving in any direction can join or leave the network at any time, and also the wireless channel can be publicly accessed without restriction. In such a context misbehaving (selfish/malicious) nodes are more likely to appear. Selfish nodes are characterized by their reluctance to spending resources to cooperate on its behalf. Malicious nodes always attack the network's availability through common techniques such as flooding, black hole and denial of service (DoS) [1]. Because of the difficulties in MANET such as dynamic network topology and constraint battery resources, security solutions that have been deployed for wired networks are not directly portable to ad hoc networks. Various techniques have been proposed to prevent misbehavior in MANETs. As described in [2], [3], these schemes can be broadly classified into security-based schemes [4], [5], [6], [7], and reputation-based schemes [8], [9]. The security-based schemes use cryptographic tools to protect the core routing protocol (signaling packets) from fabrication and modification, which in turn secures the routes, thereby protecting the data that flows through them. In a reputation-based approach, nodes (either individually or collectively) detect, and then declare another node to be misbehaving. This declaration is then propagated throughout the network, leading to the misbehaving node being avoided in all future routes. Trust based routing protocols consist of two parts: a routing part and a trust model, for a survey see [10]. Routing decisions are made according to the trust model. The trust routing protocols have to be able to identify trustworthy nodes and find a reliable and trustworthy route from sender to destination node. This has to be realized within a few seconds or better tenths of seconds, depending on the mobility of the nodes and the number of hops in the route. Most of the existing trust based routing protocols uses continuous promiscuous monitoring of the neighbors; which violate the TCP protocol rules.

Packet Dropping Attack or denial of service attacks has greatest impact on Ad hoc network [11], [12]. In DoS attack the malicious node tends to threaten network throughput through the use of packet dropping attack. This kind of attack could be even worse when supported by the malicious node sending link–layer acknowledgements to neighbor nodes to delay the detection of the attack and hence further decrease the throughput. We propose a protocol called a Two node-disjoint Routes scheme for Isolating Dropper Node in MANET (TRIDNT) [13],[14], this protocol allows monitoring, detecting, and isolating of malicious node, with allowing a controlled degree of node selfishness behavior to give an incentive to the selfish nodes to declare its selfishness behavior to its neighbors.

In this paper a mathematical model and a performance evaluation for the proposed TRIDNT protocol is presented. We derive an expression for the maximum time required to detect the misbehaving node in the routing path; if their; and calculate the expected attempt time by the routing protocol until finding a misbehaving free route. Also in this paper we will limit the number of attempt to find a misbehaving free route to the expected number of available paths without trying to use a test path again and calculate the connection blocking probability; finally we will calculate the throughput of a flow traversing the MANET in the presence of misbehaving nodes

The rest of the paper is organized as follows. Section 2, describes the related work. The TRIDNT protocol overview is presented in Section 3, followed by the Analytical model and performance evaluation in section 4 and 5 respectively. Finally we conclude our work and discuss our plan for future work in section 6.

2. Related work

In [15] Balakrishnan et al, propose a scheme of TWOACK to prevent selfishness in mobile ad hoc networks. They proposed two network-layer acknowledgment-based schemes, termed the TWOACK and the S-TWOACK schemes, which can be simply added-on to any source routing protocol. When a node forwards a packet, the node’s routing agent verifies that the packet is received successfully by the node that is two hops away on the source route. This is done through the use of a special type of acknowledgment packets, termed TWOACK packets. TWOACK packets have a very similar functionality as the ACK packets on the Medium Access Control (MAC) layer or the TCP layer. A node acknowledges the receipt of a data packet by sending back a two-hop TWOACK packet along the active source route. If the sender/forwarder of a data packet does not receive a TWOACK packet corresponding to a particular data packet that was sent out, the next-hop’s forwarding link is claimed to be misbehaving and the forwarding route broken.

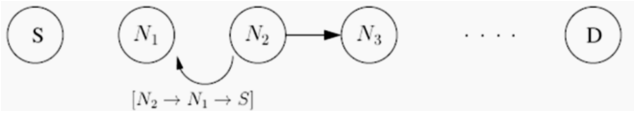


Fig. 1 TWOACK scheme

To detect misbehavior, the sender or router of a data packet maintains a list of data packet IDs that have yet to receive a TWOACK acknowledgment packet from a node two hops away. Each node maintains a unique list for each forwarding link that it is using. When a node, say, N_1, sends or forwards a data packet along a particular route, say, $N_1 \rightarrow N_2 \rightarrow N_3$, it adds the ID of the packet to LIST on its list corresponding to $N_2 \rightarrow N_3$. When it receives a TWOACK packet, it checks for the $N_2 \rightarrow N_3$ combination, and then removes the packet ID from the corresponding LIST. If a data packet ID stays on LIST longer than a certain period of time, termed timeout, misbehavior of link

$N_2 \rightarrow N_3$ is suspected. Every time misbehavior is suspected, a non-negative misbehavior counter C_{MIS} is increased by one. When C_{MIS} exceeds a certain level, termed thresh, a node declares the corresponding link, $N_2 \rightarrow N_3$, misbehaving and sends out an RERR packet informing the source about the same. Every node receiving or overhearing such an RERR packet should identify link $N_2 \rightarrow N_3$ as misbehaving. Every node maintains a list of misbehaving links that it has learned. Such links will not be chosen when it selects routes for data transmission later on.

Based on this claim, the routing protocol avoids the accused link in all future routes, resulting in an improved overall throughput performance for the network. The S-TWOACK (Selective-TWOACK) scheme is a derivative of the basic TWOACK scheme, aimed at reducing the routing overhead caused by excessive number of TWOACK packets. The basic drawback of this scheme is that it can't determine exactly which node is the misbehaving node; it only marks the link interconnecting the two nodes as misbehaving link and tries to avoid using this link in the future.

Muhammad Zeshan et al, [16] proposed a two folded approach, to detect and then to isolate a malicious node causing packet dropping attacks. First approach will detect the misbehavior of nodes and will identify the malicious activity in network. When a Source node forwards any packet to the Destination through a route, all intermediate nodes will send back an ACK packet to its source node. If the Source node doesn't receive the ACK from any intermediate node, it will send again its packet for Destination after a specific time but if again this activity was observed, Source node will broadcast a packet to declare the malicious activity in the network because until now source node upon not receiving ACK packets comes to know that one of its intermediate nodes is misbehaving.

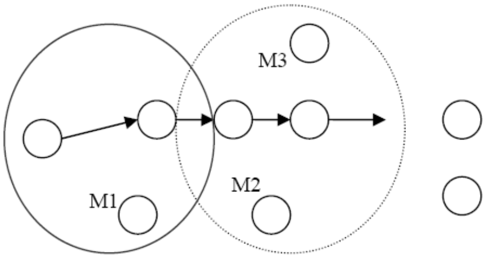


Fig. 2 Muh. Zesh. scheme

Identification of malicious node: Now upon detection of malicious activity in the network by one of intermediate node, other approach will identify that exactly which intermediate node is doing this activity. All nodes which lie in the transmission range of active route like M1, M2 and M3 and also the nodes which are on the active route become in promiscuous listening mode and count number of packet coming into and going out of the nodes of active route.

Each node in this range maintains a list of sent and dropped packets and when number of dropped packets by a particular node exceeds a certain threshold C_{max} (maximum threshold value), the monitoring node in that range declares that node as misbehaving node since a malicious activity have already been observed in the network. All the nodes in the network also maintain a list of malicious nodes. Thus upon receiving broadcast packet all the neighbors will cancel their transmission which that particular node and enter this node into the list of misbehaving nodes. The basic drawback of this scheme is, nodes cooperate together to obtain an objective opinion about another node's trustworthiness, which give the misbehaving node the chance to falsely report the value of trust score (False Misbehavior).

3. The TRIDNT Protocol overview

In this section we briefly describe the TRIDNT protocol proposed in [13], [14]. TRIDNT is a trust based routing protocol used to defend against Packet Dropping Attack in MANETs, it makes the first effort to distinguish between the malicious and selfish node, and allow a controlled degree of node selfishness. The TRIDNT protocol uses AOMDV [17], or multipath DSR [18] with a little modification of the RREQ packet to establish a high trusted two node-disjoints paths between the source and destination nodes.

TRIDNT Protocol operation:

a) **Controlled selfishness behavior**

When the TRIDNT monitoring tool detects a malicious activity, then the path searching tool starts to identify the malicious or compromised nodes in the network and isolates them, and routes around the misbehaving node.

The misbehaving node may be a selfish or malicious node, TRIDNT allows some degree of selfishness for nodes to save their resources (e.g. battery power; where nodes behave differently based on their energy levels. When the energy lies between full energy E and a threshold Es, the node behaves properly. For an energy level lower than the threshold Es, it uses its energy for transmissions of its own packets). The selfish node neighbors will:

1- Remove it from the active routes, which it is an intermediate node on it, and send Route Error (RERR) packet to the sources to establish new routes.
2- Allow it to deny being a member in any new route, and dropping any Route Request (RREQ) packet came from it.
3- Forward to/from it the packets which contain it as destination/source address.

The selfish node neighbors will restrict its selfishness behavior by a time threshold, and a repetition threshold.

So the selfish nodes are excluded from the responsibility of data forwarding. At the same time, this helps the identification of malicious nodes easier. Here we can differentiate between selfish and malicious nodes and save the misbehaving searching time (the time to find the misbehaving “selfish and malicious” node, and route around them) to only a searching time to find the malicious node only. We known that the misbehaving searching time need to be very small “i.e. find the misbehaving node very fast”, because due to the node mobility the route life time is small.

b) Route monitoring tool

In TRIDNT we use the DLL-ACK and the end to end TCP-ACK as a monitoring tool to monitor the behavior of the routing path.

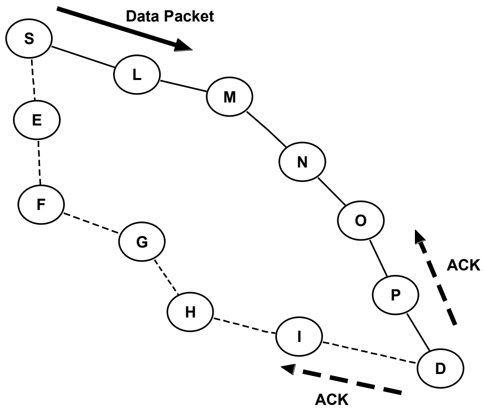


Fig. 3 TRIDNT monitoring tool

During the data transmission the source node send its data packet over the primary path only and each node in the path store the received data packet information in its Data Packet Information (DPI) cache, then forward it to its downstream neighbor, and wait for a data link layer acknowledgment (DLL-ACK) from the neighbor node. On the other hand the source node waits to receive the end to end TCP-ACK from the destination node via the primary and secondary paths.

c) Route searching mechanism

If the source node doesn’t receive one of the two TCP-ACK over the primary or the secondary routes it concludes that there is a malicious node in the primary or secondary routs, then it will run the route searching mechanism by sending a Malicious Search Packet (MSP); which contains information about the lost data packet; via the primary route towered the destination node. Every node receive this packet compare its information with the data packet information's stored in its DPI cache, if it found a match (the node received this data packet and forward it to the next node) it will forward the MSP packet to the next node with overhearing to assure that the neighbor node will forward it. The node which found a mismatch will stop forwarding of MSP packet and generate a Malicious Detection Packet "MDP (detecting node ID, detected node ID)"; it is a high priority packet forwarded with overhearing. Also the node which found that its downstream node doesn't forward the MSP packet generates the MDP packet. The node generating the MDP packet forwards it in the opposite direction to the detected malicious node, toward the source or destination node.
We make MSP and MDP high priority packets to speed up the detection process, and forwarded with overhearing to avoid the malicious node to drop that packets and break the searching and detection process.

d) Malicious node isolation

When a neighbor of a malicious node detect its malicious activity it will send the MDP packet and Because the honest (detecting node) node will suffer from the misbehaviors of malicious node, so it will insert the malicious node ID in its black list regardless of its trust score to prevent any future cooperation with it and isolate it from the network. Also when the trust value of a given node reduced below a given threshold δ it will be marked as misbehaving node and its ID inserted in the black list.

After small number of transaction all malicious nodes’ neighbors will put its ID on their black lists, so the malicious node will be fully isolated from MANET. The misbehaving node can rejoin the network only if it moves from its location and have new neighbors (whose ask the old neighbors about the node reputation), and if its reported trust value is above the trust threshold δ.

4. Analytical Model

In this section, we develop a simple model to predict the time required to detect the malicious node in the routing path if there, the expected attempt time until finding a misbehaving free route, the limited attempts based connection blocking probability and attempt time until blocking, the throughput of a flow traversing a network in the presence of misbehaving nodes, and finally we develop a simple model for the connection request blocking probability based on the probability of not finding a trusted route to carry out the RREQ packet from the source to the destination nodes.

Considering an Ad Hoc network consist of N nodes and $a < N$ misbehaving nodes, with a_m malicious nodes (black hole nodes) and a_s selfish nodes, where $a = a_m + a_s$. With the probability of randomly selected node is misbehaving node $P = \frac{a}{N}$, assuming that the two events (malicious and selfish) are mutually exclusive, then:

$$P = P_s + P_m = \frac{a_s}{N} + \frac{a_m}{N} \tag{1}$$

Where P_s: is the probability of randomly selected node is a selfish node.

P_m: is the probability of randomly selected node is a malicious node.

Let $k = \frac{a_s}{a}$, is the ratio between selfish nodes and overall malicious nodes, $0 \le k \le 1$, then

$$P_s = \frac{ka}{N} \quad , \qquad P_m = \frac{(1-k)a}{N} \tag{2}$$

Let the time taken to establish a route during the route setup phase is τ_{RR} (the time taken starting from S flooding the RREQ until it receive a multiple RREP from D to construct the routing path), and the routing path traversing h relay nodes. We will calculate:

(a) The time required to detect the malicious node if it found in the routing path τ_{md}.
(b) The expected attempt time until finding a misbehaving free route τ_{att}.
(c) The limited attempts based connection blocking probability $P_{Block}(attempts)$, and the attempt time until blocking τ_{block}.
(d) Per flow throughput G.
(e) The connection request blocking probability $P_{Block}(Request)$.

Through this mathematical evaluation we will compare our TRIDNT trust protocol with a two comparable routing protocols TWOACK [15], and Muhammad Zeshan et al. [16], which uses the same criteria in routing around and isolating of misbehaving node.

4.1 Malicious searching time

In this subsection we will calculate the time taken by the routing protocol until finding the malicious node if it is found in the routing path. The selected route on the path setup phase can contain no misbehaving node with probability$(1-P)^h$, no malicious (dropper) node with probability$(1-P_m)^h$, and no selfish node with probability$(1-P_s)^h$. Once the source node knows that there is a malicious node in the routing path it starts the malicious search phase to find that node.
In case if we found a malicious node on the routing path, then the malicious search phase will started, let for the worst case at the end of the malicious search phase a node declare its selfishness behavior so the malicious search phase may aborted, so the search time is approximately equal to the malicious detection time. For simplicity we assume that the routing path contains at least one malicious node or one selfish node, and no path contains both selfish and malicious nodes.

In this section we will calculate the expected value of malicious detection time for:

a) TRIDNT
b) Muhammad Zeshan algorithm
c) TWOACK

4.1.1 TRIDNT

In this part we calculate the maximum time taken until finding the malicious node in TRIDNT routing path. Let the secondary route source node neighbor is the malicious nodes (worst case), and for simplicity let both paths traversing the same number of h relay nodes. When the source node S detect a malicious activity it will start the malicious search phase to find the malicious node, it sends a MSP packet to the destination node D via the primary path, then the MSP packet will travel h links until it reach the destination node. The destination D will forward the MSP to the source node S over the secondary path to search it to find the malicious node, let the malicious node (node number h) drop the packet (i.e. MSP packet will travel h-1 links on the secondary path). Then node number h-1 will inform the destination node D that node h in the secondary path is the malicious node by sending MDP packet which travel *h-1* links until it reach *D* and *h* link until it reach *S*.

Because the multihop wireless ad hoc networks can be modeled as a queuing Network with end-to-end delay equals the sum of queuing and transmission delays at source and intermediate nodes, and both MSP and MDP are high priority packets, then it only suffer from the transmission delays at the intermediate nodes, by neglecting the propagation delay the transmission delay can be calculated as the sum of mean node service time τ_s along the routing path. So the overall malicious detection time of TRIDNT protocol is:

$$\tau_{md} = 2\{h + h - 1\}\,\tau_s$$
$$= 2\,\tau_s(2h - 1) \qquad (3)$$

In [19] calculates the node service time in IEEE 802.11 MAC based wireless ad hoc network as the sum of duration of random back off timer t_i which is exponentially distributed with mean $\frac{1}{\varepsilon}$, the duration for which the timer frozen $Z_i\frac{L}{\omega}$, and packet transmission time $\frac{L}{\omega} + t_o$,where t_o is the time required for the of exchange of RTS, CTS and ACK packets and neglected compared to $\frac{L}{\omega}$. So the value of node service time as in [19] is:

$$t_{ser} = t_i + Z_i\frac{L}{\omega} + \frac{L}{\omega} \qquad (4)$$

Where

Z_i: The number of times the timer of a node *i* is frozen before its expiration, and $E(Z_i) = 4\,N\,A_n\lambda_i\,\overline{t_{ser}}$

L: is the packet size.

ω: node transmission rate.

$A_n = \pi r_n^{\,2}$: node communication area.

r_n : node transmission range.

λ_i : effective arrival rate at a station *i*,

Then the expected value of node service time can be modeled as in [19]:

$$\tau_s = \overline{t_{ser}} = \frac{\frac{1}{\varepsilon} + \frac{L}{\omega}}{1 - 4\,N\,A_n\,\lambda_i\,\frac{L}{\omega}} \qquad (5)$$

The packet generation process at each node is assumed to be an i.i.d Poisson process with rate λ as in [19]. When a node receives a packet from any of its neighbors, it either forwards the packet to its neighbors with probability (1 - p(n)) or absorbs the packet with probability p(n). The probability p(n) is the probability that a node is the destination of a packet given that the node has received the packet from its neighbors (absorption probability). So the effective packet arrival rate at a node *i* can be calculated as in [19]

$$\lambda_i = \lambda\frac{1}{p(n)} \qquad (6)$$

Where $\frac{1}{\mathrm{p(n)}}$ = the expected number of hops traversed by a packet between its source and destination, and can be chosen to equal $\sqrt{\frac{\mathrm{N}}{\log\,\mathrm{N}}}$ as in [19].

Finally we have the overall malicious detection time for TRIDNT protocol is

$$\tau_{md}|_{TRIDNT} = \frac{2\,(2h - 1)\,\left(\frac{1}{\varepsilon} + \frac{L}{\omega}\right)}{1 - 4\,N\,A_n\,\frac{\lambda}{\sqrt{\frac{\log\,N}{N}}}\,\frac{L}{\omega}} \qquad (7)$$

4.1.2 Muhammad Zeshan algorithm

In Muhammad Zeshan algorithm [16] the time of detecting misbehaving node will depend on a threshold called the maximum number of allowable dropped packets C_{max}, and distance between the source node and malicious node. In this algorithm the malicious node identification phase start when the source node broadcast a packet to declare the malicious activity on the active route; for the worst case let the last node in the path (node number *h*) is the misbehaving node; on the average the broadcasted packet will travel *h* nodes until it reach the malicious node neighbors, which take a time equal ($h\tau_n$) where the node total delay τ_n equal to the node service time τ_s plus the node queuing delay τ_q.

Then all nodes in the malicious node range are in promiscuous mode and count the number of packets coming into and going out of the malicious node, and when the number of dropped packets of the a particular node exceeds a certain threshold C_{max} the detected node is marked as misbehaving node. For the best case let the malicious node is a member in at least number of routes equals C_{max} and the source nodes send a data packets on that routes, and for simplicity let all routes have the same average path length *h* and the malicious node is the last node in all routing paths, then the time taken by these packets to reach the malicious nodes is ($h\tau_n$). But for worst case if the malicious node is a member on only one route, then the source node will try to send a data packet C_{max} times and between each trial the source node will wait a time equal to the retransmission timeout (τ_{ret}) except in the last trial after the data packets travel *h* nodes and reach the malicious node the monitoring nodes counter will reach C_{max}, so the monitoring nodes will detect malicious node after a time equal ($h\tau_n + (C_{max} - 1)\ \tau_{ret}$).

Finally the monitoring nodes will declare the misbehaving node ID to the source node, the declaration message will travel on the average *h* nodes a take a time equal to ($h\tau_n$) until it reach the source node.

So the malicious detection time is:

$$\tau_{md}|_{Muh.,min} = 3h\tau_n \tag{8}$$

$$\tau_{md}|_{Muh.,max} = 2h\tau_n \ + h\tau_n \ + (C_{max} - 1)\,\tau_{ret}$$

$$= 3h\tau_n \ + (C_{max} - 1)\,\tau_{ret} \tag{9}$$

Where: τ_n = node service time τ_s + queuing delay τ_q, is the node total delay.

The node *i* queuing delay can be calculated by using little’s low as in [19]:

$$\tau_q = \frac{\rho}{\lambda_i(1 - \rho)} \tag{10}$$

Where: ρ is the traffic intensity.

By substituting (6) into (10), we have:

$$\tau_q \ = \ \frac{\rho\sqrt{\frac{\log\ N}{N}}}{\lambda(1-\rho)} \tag{11}$$

$$\tau_{md}|_{Muh.,min} = 3h\left(\frac{\left(\frac{1}{\varepsilon} + \frac{L}{\omega}\right)}{1 - 4\,N\,A_n\,\frac{\lambda}{\sqrt{\frac{\log\ N}{N}}}\,\frac{L}{\omega}} + \frac{\rho\sqrt{\frac{\log\ N}{N}}}{\lambda(1-\rho)}\right) \tag{12}$$

$$\tau_{md}|_{Muh.,max} = 3h\left(\frac{\left(\frac{1}{\varepsilon} + \frac{L}{\omega}\right)}{1 - 4\,N\,A_n\,\frac{\lambda}{\sqrt{\frac{\log\ N}{N}}}\,\frac{L}{\omega}} + \frac{\rho\sqrt{\frac{\log\ N}{N}}}{\lambda(1-\rho)}\right) + (C_{max} - 1)\,\tau_{ret} \tag{13}$$

4.1.3 TWOACK

In TWOACK [15] protocol if the data packet is dropped C_{max} times in any link in the path, that link marked as suspect link. For the worst case let the last node in the path (node number *h*) is the misbehaving node; where the malicious node one hop upstream neighbor is the monitoring node, which waits the receiving of TWOACK packet from the destination node during a period called timeout τ_{out}, and count the number of dropped packets on the link between the malicious and destination nodes, if it is exceeds a given threshold value C_{max} the link is marked as misbehaving link. For the best case when the monitoring, malicious, and destination nodes are members in at least C_{max} routes the link is marked as misbehaving after ($(h\text{-}1)\,\tau_n + \tau_{out}$) seconds, in the worst case if all those three nodes are members in only one route the link is marked as misbehaving after ($(C_{max} - 1)\ \tau_{ret} + (h\text{-}1)\,\tau_n + \tau_{out}$) seconds. The detecting node send a RERR packet to the source node S contain the ID of malicious node, which travel (*h-2)* node until it reach the source node. So the malicious detection time is

$$\tau_{md}|_{TWOACK,min} = (h-1)\,\tau_n + \tau_{out}$$

$$+(h-2)\,\tau_n \tag{14}$$

$$\tau_{md}|_{TWOACK,max} = (h-1)\,\tau_n + \tau_{out}$$

$$+(C_{max} - 1)\,\tau_{ret} + (h-2)\,\tau_n \tag{15}$$

$$\tau_{md}|_{TWOACK,min} = (2h-3)\left(\frac{\left(\frac{1}{\varepsilon} + \frac{L}{\omega}\right)}{1 - 4\,N\,A_n\,\frac{\lambda}{\sqrt{\frac{\log\ N}{N}}}\,\frac{L}{\omega}} + \frac{\rho\sqrt{\frac{\log\ N}{N}}}{\lambda(1-\rho)}\right) + \tau_{out} \tag{16}$$

$$\tau_{md}|_{TWOACK,max} = (2h-3)\left(\frac{\left(\frac{1}{\varepsilon} + \frac{L}{\omega}\right)}{1 - 4\,N\,A_n\,\frac{\lambda}{\sqrt{\frac{\log\ N}{N}}}\,\frac{L}{\omega}} + \frac{\rho\sqrt{\frac{\log\ N}{N}}}{\lambda(1-\rho)}\right) + \tau_{out} + (C_{max} - 1)\,\tau_{ret} \tag{17}$$

4.2 Expected attempt time to find misbehaving free route

In this subsection we will try to calculate the expected attempt time the source node taken until it can find a misbehaving free rout in case of high dense ad hoc network.

After setting up a routing path between the source and destination nodes, the path can breaks duo to one of two reasons:

a) If any node in the routing path is moved out of the communication range from one of its upstream or downstream neighbor nodes.
b) If any node in the path found to be in misbehaving activity; selfish node or malicious node.

The established path have an expected life time E(τ_L) [11] as determined by factors such as the node velocity and node density. The probability of route breaks due to misbehaving activity of at least one node on the path during the route expected life time is equal {1-(1-P)^h}.

Where: Probability of at least one node on the path is selfish equal {1-(1-P$_s$)^h }.

Probability of at least one node on the path is malicious equal {1-(1-P$_m$)^h }.

The route repairing time depend on the number of delays, first the time taken to diagnose that the route is broken (τ_{diag}), secondly the route reestablishment time τ_{RR} . In case of route breaks due to malicious node, there is another time delay called a malicious detection time τ_{md} . Let the source node try $\tau_{s,\ att}$ attempt time until finding a selfish free route, and $\tau_{m,\ att}$ attempt time until finding a malicious free rout, so the total attempt time

$$\tau_{att} = \tau_{\mathrm{s,att}} + \tau_{\mathrm{m,att}}$$

$$\tau_{att} = \sum_{j=1}^{t_s}\left[E\left(\tau_{diag}^{j}|_{selfish}\right) + E\left(\tau_{RR}^{j}\right)\right] + \sum_{j=1}^{t_m}\left[E\left(\tau_{diag}^{j}|_{malicious}\right) + E\left(\tau_{RR}^{j}\right) + E\left(\tau_{md}^{j}\right)\right] \quad (18)$$

Where τ_{diag}^{j} is the diagnostic time of j^{th} attempt.

t_s the number of attempts until find a selfish free route.

t_m the number of attempts until find a malicious free route.

4.2.1 TRIDNT

In this part we will calculate the expected attempts time TRIDNT protocol taken until finding a misbehaving free route. Due to the frequent topology changes of ad hoc network the routes contains a malicious node can be malicious free route in a few seconds, and the selfish nodes may be switched to the cooperation mode so the routes contains a selfish node may converted to a selfish free route at any time. For this reasons the number of attempt until finding a misbehaving free route may not limited to the maximum number of available node-disjoint routes, so the expected attempts time for the TRIDNT protocol is

$$E(\tau_{att})|_{TRIDNT} = \sum_{t_s=1}^{\infty}\left\{\sum_{j=1}^{t_s}\left[E\left(\tau_{diag}^{j}|_{selfish}\right) + E\left(\tau_{RR}^{j}\right)\right]\right\} \times \{(1-P_s)^h(1-(1-P_s)^h)^{t_s}\} + \sum_{t_m=1}^{\infty}\left\{\sum_{j=1}^{t_m}\left[E\left(\tau_{diag}^{j}|_{malicious}\right) + E\left(\tau_{RR}^{j}\right) + E\left(\tau_{md}^{j}\right)\right]\right\} \times \{(1-P_m)^h(1-(1-P_m)^h)^{t_m}\} \quad (19)$$

When the source node send the data packet and wait a retransmission time out and don't receive the TCP ACK via the primary and secondary routs it will try to send the packet again if it also don't receive the TCP ACK again it start the malicious detection phase. So the diagnose time for TRIDNT in case of malicious node is $\tau_{diag}^{j}|_{malicious} = 2\tau_{ret}$. Because the selfish node will declare itself, the diagnose time in case of selfish node is $\tau_{diag}^{j}|_{selfish} < \tau_{ret}$.

For simplicity, we consider a fixed path length equal h, and $E\left(\tau_{xx}^{j}\right) = E(\tau_{xx}) \quad \forall\, j$, so

$$E(\tau_{att})|_{TRIDNT} = \sum_{t_s=1}^{\infty}\left\{t_s\left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right](1-P_s)^h\left(1-(1-P_s)^h\right)^{t_s}\right\} + \sum_{t_m=1}^{\infty}\left\{t_m\left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md})\right](1-P_m)^h\left(1-(1-P_m)^h\right)^{t_m}\right\} \tag{20}$$

$$E(\tau_{att})|_{TRIDNT} = \left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right](1-P_s)^h \sum_{t_s=1}^{\infty}\left\{t_s\left(1-(1-P_s)^h\right)^{t_s}\right\} + \left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md})\right](1-P_m)^h \sum_{t_m=1}^{\infty}\left\{t_m\left(1-(1-P_m)^h\right)^{t_m}\right\} \tag{21}$$

Because $\sum_{i=0}^{\infty} i\, x^i = \frac{x}{(1-x)^2}$, $x < 1$, so

$$E(\tau_{att})|_{TRIDNT} = \left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right](1-P_s)^h\left(\frac{1-(1-P_s)^h}{(1-P_s)^{2h}}\right) + \left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md})\right](1-P_m)^h\left(\frac{1-(1-P_m)^h}{(1-P_m)^{2h}}\right) \tag{22}$$

$$E(\tau_{att})|_{TRIDNT} = \left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right]\left(\frac{1-(1-P_s)^h}{(1-P_s)^{h}}\right) + \left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md})\right]\left(\frac{1-(1-P_m)^h}{(1-P_m)^{h}}\right) \tag{23}$$

$$E(\tau_{att})|_{TRIDNT} = \left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right]\left(\frac{1}{(1-P_s)^h} - 1\right) + \left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md}|_{TRIDNT})\right]\left(\frac{1}{(1-P_m)^h} - 1\right) \tag{24}$$

Substituting from equation (2) into equation (24), then we have

$$E(\tau_{att})|_{TRIDNT} = \left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right]\left(\frac{1}{\left(1-\frac{ka}{N}\right)^h} - 1\right) + \left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md}|_{TRIDNT})\right]\left(\frac{1}{\left(1-\frac{(1-k)a}{N}\right)^h} - 1\right) \tag{25}$$

$$E(\tau_{att})|_{TRIDNT} = E(\tau_{att})|_{selfish} + E(\tau_{att})|_{malicious} \tag{26}$$

4.2.2 Muhammad Zeshan algorithm

Muhammad Zeshan algorithms can't distinguish between misbehaving activity due to selfishness behavior or due to malicious activity (dropper), it deals with the selfish node as attacking node. So the expected attempt time is

$$E(\tau_{att})|_{Muh.} = \left[E\left(\tau_{diag}|_{Muh.}\right) + E(\tau_{RR})\right.$$

$$\left.+E(\tau_{md}|_{Muh.})\right]\left[\frac{1}{\left(1-\frac{a}{N}\right)^{h}} - 1\right] \qquad (27)$$

In Muhammad Zeshan algorithms after two failed trials of sending data packets it will start the malicious node identification phase, so the diagnoses time $\tau_{diag}|_{Muh.} = 2\,\tau_{ret}$.

4.2.3 TWOACK algorithms

Also TWOACK algorithms can't distinguish between misbehaving activity due to selfishness behavior or due to malicious activity (dropper), it deals with the selfish node as attacking node. So the expected attempt time is

$$E(\tau_{att})|_{TWOACK} = \left[E\left(\tau_{diag}|_{TWOACK}\right) + E(\tau_{RR})\right.$$

$$\left.+E(\tau_{md}|_{TWOACK})\right]\left[\frac{1}{\left(1-\frac{a}{N}\right)^{h}} - 1\right] \qquad (28)$$

In TRIDNT the diagnoses is done during the malicious detection phase, so we will tack the diagnoses time $\tau_{\mathrm{diag}}|_{\mathrm{TWOACK}} = 0$.

4.3 Limited-attempts-based connection blocking probability

If we try to limit the number of attempts for finding a misbehaving free route to the maximum number of available node-disjoint routes, and after we examine all the available routes without finding a misbehaving free route the connection will blocked. In this part we will develop a simple model to predict the limited-attempts-based connection blocking probability in the ad hoc network in the presence of misbehaving nodes.

The connection will be blocked if the numbers of attempts to find a misbehaving free route t_{all} reach the maximum available number node-disjoint routes n_r, and we can't find a misbehaving free route. So the connection attempts blocking probability can be calculated as:

$$P_{Block}(attempts) = P_r(t_{all} > n_r) = 1 - P_r(t_{all} \leq n_r)$$

$$= 1 - \sum_{t_{all}=1}^{n_r} P_r(t_{all})$$

$$= 1 - \sum_{t_{all}=1}^{n_r} (1-P)^h(1-(1-P)^h)^{t_{all}}$$

$$= 1 - (1-P)^h \sum_{t_{all}=1}^{n_r} (1-(1-P)^h)^{t_{all}} \qquad (29)$$

Because, $\sum_{i=0}^{k} x^i = \frac{x^{k+1}-1}{x-1}$, so

$$P_{Block}(attempts) = 1 - (1-P)^h * \left(\frac{(1-(1-P)^h)^{n_r+1} - 1}{(1-(1-P)^h) - 1}\right) \qquad (30)$$

Final we have

$$P_{Block}(attempts) = (1-(1-P)^h)^{n_r+1} \qquad (31)$$

Given that each route has an average number of hops h, the maximum number of disjoint routes, corresponding to a scenario where each node belongs to a particular route (i.e., as a source, a relay node, or a destination), can simply be written as in [20]:

$$n_r \cong \frac{N}{h} \qquad (32)$$

Because $= \frac{a}{N}$, the connection blocking probability can be write as

$$P_{Block}(attempts) = \left(1 - \left(1 - \frac{a}{N}\right)^h\right)^{\frac{N}{h}+1} \qquad (33)$$

The attempt time until blocking τ_{block}, is the time taken by the trust based routing protocol to examine all available routes and decide to block the connection request if it didn't find a misbehaving free route and can be calculated as

$$\tau_{block} = n_r\left[E\left(\tau_{diag}\right) + E(\tau_{RR}) + E(\tau_{md})\right]$$

$$\tau_{block} = \frac{N}{h}\left[E\left(\tau_{diag}\right) + E(\tau_{RR}) + E(\tau_{md})\right] \qquad (34)$$

In TRIDNT protocol we allow a controlled degree of node selfishness, so the attempts time until blocking can calculated as the time until TRIDNT protocol decide to block the connection request if it didn't find a malicious and selfish free routes, and can write as

$$\tau_{block}|_{TRIDNT} = \frac{Nk}{h}\left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right]$$

$$+\frac{N(1-k)}{h}\left[E\left(\tau_{diag}|_{malicious}\right)\right.$$

$$\left.+E(\tau_{RR}) + E(\tau_{md})|_{TRIDNT}\right] \quad (35)$$

Where:

$\frac{Nk}{h}$; *The number of routes contains a selfish node.*

$\frac{N(1-k)}{h}$; *The number of routes contains a malicious node.*

$\left[E\left(\tau_{diag}|_{selfish}\right) + E(\tau_{RR})\right]$; *The selfish route repairing time.*

$\left[E\left(\tau_{diag}|_{malicious}\right) + E(\tau_{RR}) + E(\tau_{md})|_{TRIDNT}\right]$; *The malicious route repairing time.*

Also TWOACK and Muhammad Zeshan protocols attempt time until blocking can be written as

$$\tau_{block}|_{TWOACK,or\ Muh.} = \frac{N}{h}\left[E\left(\tau_{diag}\right)|_{TWOACK,or\ Muh.}\right.$$

$$\left.+E(\tau_{RR}) + E(\tau_{md})|_{TWOACK,or\ Muh.}\right] \quad (36)$$

We can say that if the expected attempt time to find a misbehaving free route is less than the attempt time until blocking, this indicate a low blocking probability, as the expected attempt time increased above the attempt time until blocking the blocking probability increased according the difference between them.

4.4 Per flow throughput

In this subsection we use the model in [11], to predict the normalized throughput (goodput) of a flow traversing a network in the presence of misbehaving nodes.

When a routing path is established between the source node *S* and the destination node *D*, per flow throughput can be calculated as the ratio between the useful time (duration of operational time in transmitting data), to the total time taken in transmission (useful time plus attempt time), where the time of reestablishing a beaked route (attempt time) having a zero transmitted data. So the per flow goodput is

$$G = \frac{E(\tau_l)}{E(\tau_l) + E(\tau_0)} \quad (37)$$

Where (τ_0) : is the total expected time of zero throughput,

$$E(\tau_0) = \tau_{RR} + E(\tau_{att}) \quad (38)$$

So the per flow goodput can be calculated as

$$G = \frac{E(\tau_l)}{E(\tau_l) + \tau_{RR} + E(\tau_{att})} \quad (39)$$

4.5 Connection request blocking probability

A large number of misbehaving nodes can partition the ad hoc network in which multihop communication becomes impossible. However, misbehaving nodes can have the effect of starving multihop flows and giving all the capacity to one-hop flows that have no relay nodes. So, in this section we calculate the connection request blocking probability, i.e. the case if there isn't a trusted route to carry the RREQ packet from the source to the destination.

At the start of path setup phase the source node will flood the RREQ packet if it found at least one trusted neighbor in single route case, and at least two trusted neighbors in two node-disjoint route case, otherwise the connection request will blocked.

4.5.1 Case 1: single route (Muhammad Zeshan and TWOACK algorithms)

In the single route case the source node flood the RREQ packet when it find at least one trusted node on its transmission range, and the trusted route can be built if and only if the source node downstream trusted neighbor and all intermediate nodes have at least one downstream trusted node over their communication range. Finally the route completed if the destination node has a direct connection with at least one trusted node from the indirect source node downstream neighbors.

So for Muhammad Zeshan and TWOACK algorithms a trusted route can be found if:

1) The source "S" and destination "D" nodes have at least one trusted node over their communication range, and
2) Each intermediate node has at least two trusted nodes over its communication range (one upstream node and one downstream node).

Then we can write the probability of finding a trusted route from S to D as:

$$P_r(trusted\ route) = P_r(at\ least\ one\ trusted\ node\ neighbor)^2 * P_r(at\ least\ two\ trusted\ node\ neighbors)^h$$

So the connection request blocking probability can be written as:

$$P_{Block}(Request) = 1 - P_r(trusted\ route)$$

$$\begin{aligned} P_{Block}(Request) \\ &= 1 - ([P_r(at\ least\ one\ neighbor\ over\ the\ node\ coverage\ area) P_r(the\ neighbor\ is\ trusted)]^2 \\ &\quad * [P_r(at\ least\ two\ neighbors\ over\ the\ node\ coverage\ area) P_r(the\ neighbors\ are\ trusted)]^h) \end{aligned}$$

For a large number of uniformly distributed MANET nodes over an area A_t, the probability of K_n nodes are located in the coverage area of a given node can be approximated with Poisson distribution as stated in [24]:

$$\begin{aligned} P_r(K_n\ neighbor\ over\ the\ node\ coverage\ area) \\ &= \frac{(\rho_N A_n)^{K_n}}{K_n!}\ e^{-\rho_N A_n} \end{aligned} \qquad (40)$$

Where ρ_N is the network nodes density $= \frac{N}{A_t}$.
Then the connection blocking probability is:

$$P_{Block}(Request) = 1 - \left\{ \left[(1 - P_r(K_n = 0)) * \left(1 - \frac{a}{N}\right) \right]^2 * \left[(1 - P_r(K_n = 0) - P_r(K_n = 1)) \left(1 - \frac{a}{N}\right)^2 \right]^h \right\} \qquad (41)$$

$$P_{Block}(Request) = 1 - \left\{ \left[(1 - e^{-\rho_N A_n}) * \left(1 - \frac{a}{N}\right) \right]^2 * \left[(1 - e^{-\rho_N A_n} - \rho_N\, A_n\, e^{-\rho_N A_n}) \left(1 - \frac{a}{N}\right)^2 \right]^h \right\} \qquad (42)$$

Because $N = \rho_N\ A_t$, then finally we have

$$\begin{aligned} P_{Block}(Request)|_{TWOACK,and\ Muh.} \\ &= 1 - \left\{ \left[(1 - e^{-\rho_N A_n}) * \left(1 - \frac{a}{\rho_N\ A_t}\right) \right]^2 \right. \\ &\quad \left. * \left[(1 - e^{-\rho_N A_n} - \rho_N\, A_n\, e^{-\rho_N A_n}) \left(1 - \frac{a}{\rho_N\ A_t}\right)^2 \right]^h \right\} \end{aligned} \qquad (43)$$

4.5.2 Case 2: tow node-disjoint routes (TRIDNT scheme)

In the two node-disjoint routes case the source node flood the RREQ packet when it find at least two trusted nodes on its transmission range, and the trusted route can be built if and only if the source node downstream trusted neighbors and all intermediate nodes have at least two downstream trusted nodes over their communication range. Finally the route completed if the destination node has a direct connection with at least one trusted node from the indirect source node downstream neighbors on each route.
So for TRIDNT scheme two node-disjoint routes can be found if:

1) The source "S" and destination "D" nodes have at least two trusted node over their communication range, and
2) Each intermediate node has at least three trusted nodes over its communication range (one upstream node and two downstream nodes).

Then we can write the probability of finding a trusted route from S to D as:

$$\begin{aligned} P_r(trusted\ route) \\ &= P_r(at\ least\ two\ trusted\ node\ neighbor)^2 \\ &\quad * P_r(at\ least\ three\ trusted\ node\ neighbors)^{2h} \end{aligned}$$

So the connection request blocking probability can be written as:

$$\begin{aligned} P_{Block}(Request) \\ &= 1 - ([P_r(at\ least\ two\ neighbor\ over\ the\ node\ coverage\ area) P_r(the\ neighbor\ are\ trusted)]^2 \\ &\quad * [P_r(at\ least\ three\ neighbors\ over\ the\ node\ coverage\ area) P_r(the\ neighbors\ are\ trusted)]^{2h}) \end{aligned}$$

Then the connection blocking probability is:

$$P_{Block}(Request) = 1 - \left[(1 - P_r(K_n = 0) - P_r(K_n = 1)) \left(1 - \frac{a}{N}\right)^2 \right]^2$$

$$* \left[\left(1 - P_r(K_n = 0) - P_r(K_n = 1) - P_r(K_n = 2)\right) \left(1 - \frac{a}{N}\right)^3 \right]^{2h} \qquad (44)$$

$$P_{Block}(Request) = 1 - \left[\left(1 - e^{-\rho_N A_n} - \rho_N\, A_n\, e^{-\rho_N A_n}\right) \left(1 - \frac{a}{N}\right)^2 \right]^2$$
$$* \left[\left(1 - e^{-\rho_N A_n} - \rho_N\, A_n\, e^{-\rho_N A_n} - \frac{(\rho_N\, A_n)^2}{2}\, e^{-\rho_N A_n}\right) \left(1 - \frac{a}{N}\right)^3 \right]^{2h} \qquad (45)$$

Finally we have

$$P_{Block}(Request)|_{TRIDNT} = 1 - \left[\left(1 - e^{-\rho_N A_n} - \rho_N\, A_n\, e^{-\rho_N A_n}\right) \left(1 - \frac{a}{\rho_N\, A_n}\right)^2 \right]^2$$
$$* \left[\left(1 - e^{-\rho_N A_n} - \rho_N\, A_n\, e^{-\rho_N A_n} - \frac{(\rho_N\, A_n)^2}{2}\, e^{-\rho_N A_n}\right) \left(1 - \frac{a}{\rho_N\, A_n}\right)^3 \right]^{2h} \qquad (46)$$

5. Results and performance evaluation

In this section the performance results for the proposed TRIDNT protocol is compared with the TWOACK and Muhammad Zeshan protocols. The comparison is done using four metrics, the time required to detect the malicious node in the routing path, the expected attempt time until finding a misbehaving free route, the limited attempts based connection blocking probability and attempts time until blocking, the per flow throughput, and finally the connection request blocking probability. Through this performance evaluation section we assume that the node transmission range for asymptotically connected network is as stated in [19] $r_n = \sqrt{\frac{log\ N}{N}}$, the packet size L = 1 K bits, the node transmission rate ω = 10^6 bits/sec , the number of network nodes N = 800 nodes, the average path length h =20 nodes, the ratio of selfish nodes to the overall number of misbehaving nodes k = 0.5, the number of malicious nodes a = 40, and Poisson arrival rate at a station λ = 0.7.

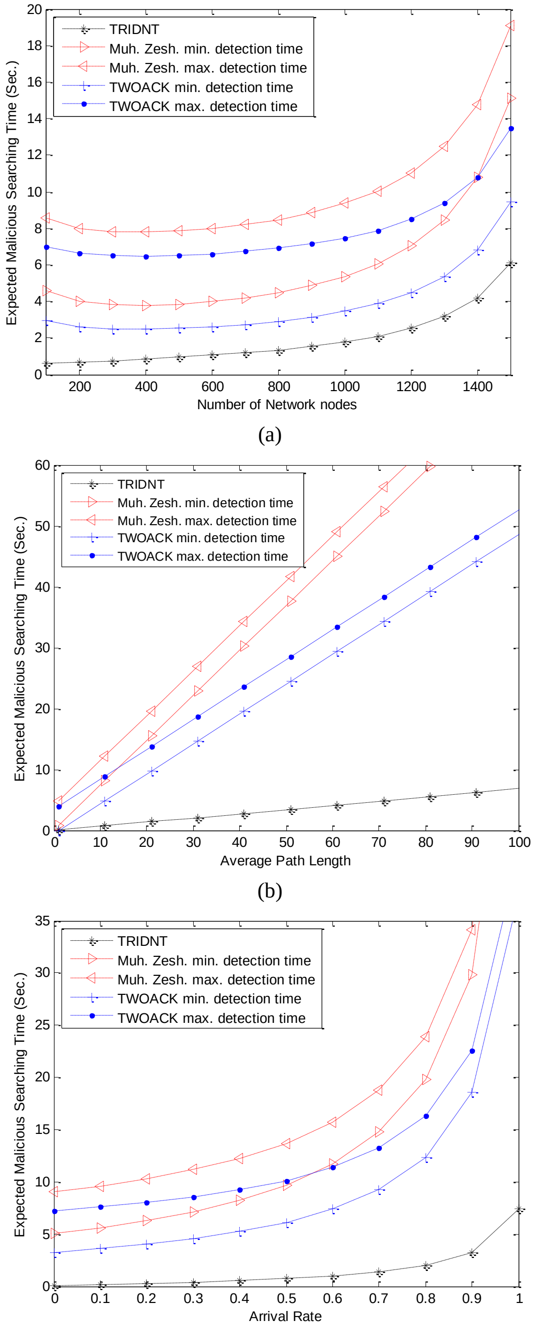
Because the random back off time equal the multiplication of random number by the slot time, as stated in [21], where the slot time = 20 μ Sec, and 0 < random number < CW. And the contention window 31 ≤ CW ≤ 1023 , the CW starts from CW_{min} and increased exponential as the unsuccessful data packet transmission increase (collision increase), then the CW is exponentially related to node packet generation rate. So $\frac{1}{\varepsilon} = C\ (1\text{-}\ e^{-\lambda})$ sec. Where C is a constant value, we will calculate that constant as the average back off time at the maximum CW, so we will tack $\frac{1}{\varepsilon} = 10.23\ (1\text{-}\ e^{-\lambda})$ m sec.

The default retransmission timeout value of TCP (TCP retransmission rule), $\tau_{ret} = 1$ sec. In case of controlled degree of node selfishness, the time taken to diagnose that the route is broken due to selfish node will has a value $\tau_{diag} < \tau_{ret}$ because when there is a selfish node its neighbor will inform the source node directly, so let $\tau_{diag}|_{selfish} = 0.9\ \tau_{ret}$. Also let $\tau_{RR} = 1$ sec as reported in [11]. And we will take $E(\tau_l) = 10$ sec corresponding to V_{max} = 30 m/sec as reported in [22], [23].

From figure 4 we see that the TRIDNT protocol will find the malicious node at a time smaller than the minimum expected malicious searching time of the TWOACK and Muhammad Zeshan protocols in case of low traffic intensity. The probability that TWOACK and Muhammad Zeshan protocols have a minimum expected malicious searching time is low due to the lowest probability of a malicious node to be a member in C_{max} routes in the same time, and without differentiation between selfish node and malicious nodes. Also as the traffic intensity increased the expected malicious searching time of TWOACK and Muhammad Zeshan protocols increasing at rate higher than that of TRIDNT protocol, this because in TRIDNT we use the MSP and MDP packets with high priority which will not suffer from the queuing delay, but in the other two protocols the searching process done by trying to send the data packet C_{max} number of times until it detect the malicious node so during the searching process the packet suffer from a queuing delay.

Figure 4-a shows how the expected malicious searching time, varies with the number of network nodes for TRIDNT, TWOACK, and Muhammad Zeshan protocols. We can see that as the network node increases the malicious detection time increases because as the network nodes increases the node offered traffic load will increased which will increase the node service time as seen in equation (5). Also we can see that the TWOACK protocol has the smallest expected malicious searching, and the TRIDNT protocol expected malicious detection time increases with rate equal to the TWOACK and Muhammad Zeshan protocols.

The expected malicious searching time versus the average path length is shown in figure 4-b.



(a)

(b)

(c)

Fig. 4 Comparison of TRIDNT, TWOACK and Muhammad Zeshan against expected malicious searching time

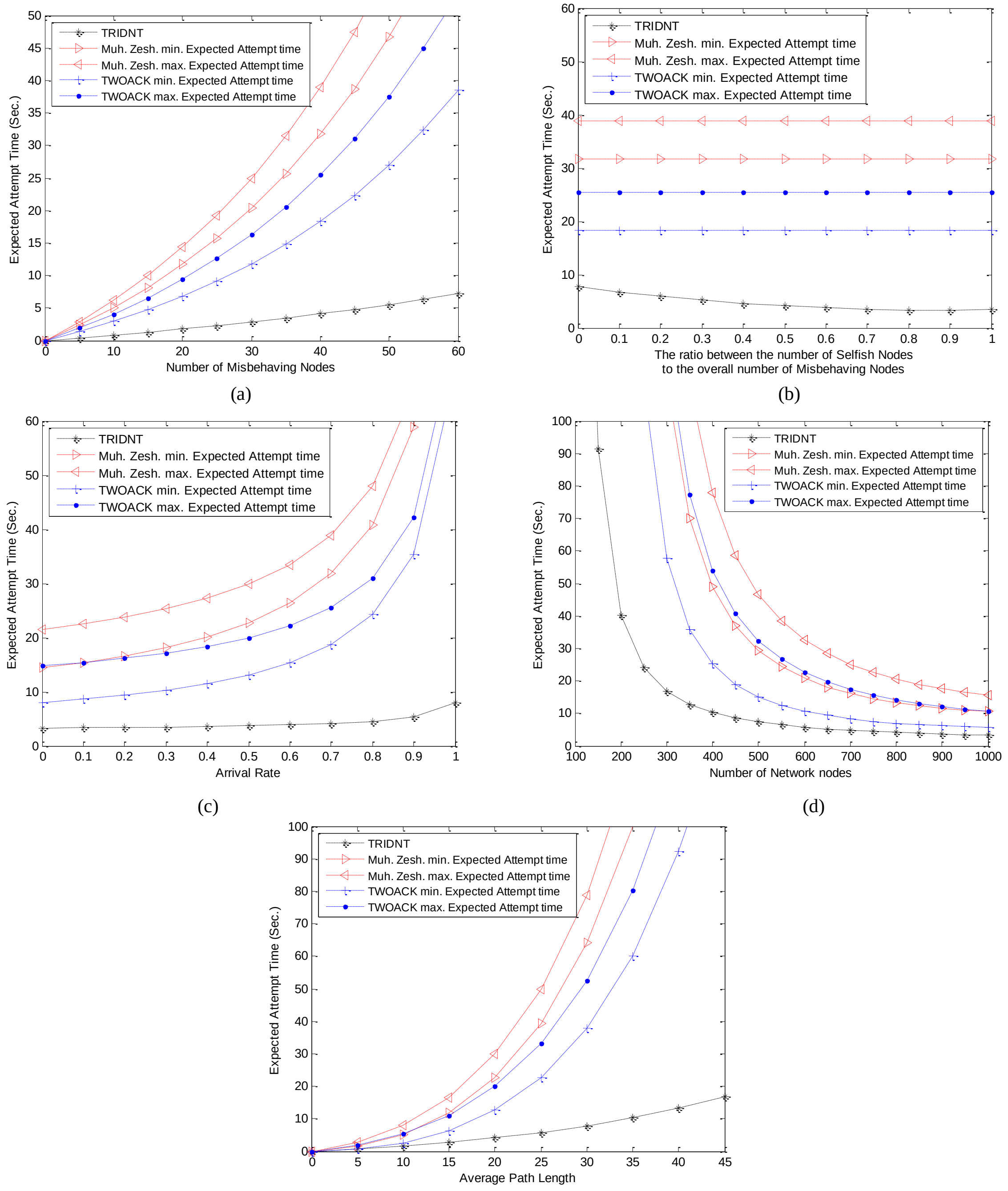
We can see that as the average path length increase the malicious node searching time increase. This phenomenon is expected because, the malicious node searching time is directly proportional to the number of node in the routing path as seen in equations (7),(12), (13), (16) and (17). From this figure we can find that the expected malicious searching time of TRIDNT protocol is smaller than that of Muhammad Zeshan and TWOACK protocols, the TRIDNT increasing rate = 68.5 m sec/node, TWOACK increasing rate = 485.05 m sec/node, Muhammad Zeshan increasing rate = 737.28 m sec/node.

Figure 4-c shows the relation between the expected malicious searching time and the packet arrival rate. From figure 4-c we can see that the malicious searching time increase as the arrival rate increase because the node service time and queuing delay is directly proportional to the arrival rate as seen in equations (5), and (11). We can see that at a small value of arrival rate TRIDNT protocol has the smallest malicious searching time than TWOACK and Muhammad Zeshan protocols, and the malicious searching time of TWOACK and Muhammad Zeshan protocols is increased with rate higher than that of TRIDNT. This because in TWOACK and Muhammad Zeshan as the arrival rate increased, the number of packets in the queue increase which increases the queuing delay, and the node service time also increases but in TRIDNT the malicious searching phase packet suffer only from the increasing of node service time only.

As shown in figure 5 we draw the TRIDNT expected attempt time and compare it with the expected attempt time in TWOACK and Muhammad Zeshan protocols. As we seen in the figure the TRIDNT protocol has the smallest expected attempt time, and in other cases the TRIDNT protocol expected attempt time is little smaller than the minimum expected attempt time of TWOACK protocol, there are three reasons behind that: **1)** TRIDNT protocol use two high priority packets during the misbehaving searching phase (don’t suffer from queuing delay), but in the TWOACK and Mohamed Zeshan protocols trying to send the data packets many times until them detects the misbehaving node which mean the packets during searching phase will suffer from queuing delay. **2)** The worst case in TRIDNT the searching packet will travel two routes only until it find the malicious node, but in TWOACK and Muhammad Zeshan protocols the data packet will travel one route C_{max} number of times until it find the misbehaving node. **3)** TRIDNT differentiate between malicious and selfish node and give an incentive to selfish node to declare itself which save the selfish searching time, but in TWOACK and Muhammad Zeshan protocols can’t differentiate between malicious and selfish nodes.

In figure 5-a the relation between the expected attempt time and the number of misbehaving node we find that the expected attempt time increased as the number of misbehaving nodes increased because the probability of finding a misbehaving nodes in the routing path increased as the number of misbehaving nodes increased, also we can see that TRIDNT protocol have the smallest expected attempt time and the smallest increasing rate, due to the reasons stated above.

Figure 5-b shown the variation of expected attempt time as a function of the ratio of selfish nodes to the overall malicious nodes as seen the TWOACK, and Muhammad Zeshan protocols have a fixed expected attempt time because it deals with selfish nodes as malicious nodes.



(a)

(b)

(c)

(d)

(e)

Fig. 5 Comparison of TRIDNT, TWOACK and Muhamed Zeshan against expected attempt time.

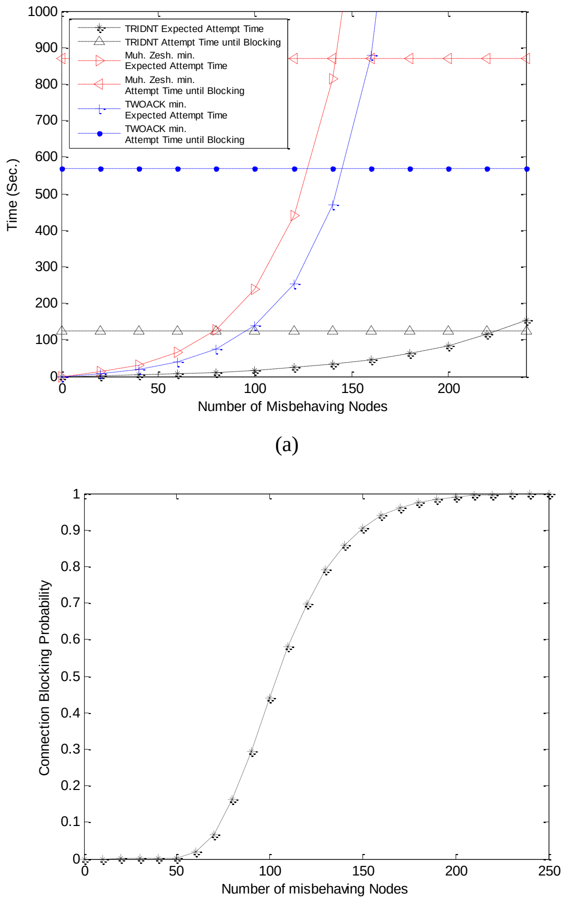
Figure 5-b show that the TRIDNT expected attempt time decreased from 7.775 sec until it reach it minimum value (= 3.4 sec) at k = 1. This behavior is due to the relation between the TRIDNT expected attempt time to find a selfishness free route $(E(\tau_{att})|_{selfish})$ is directly proportional to the value of k, the TRIDNT expected attempt time to find a malicious free route $(E(\tau_{att})|_{malicious})$ is inversely proportional to the value of k, and the overall expected attempt time $E(\tau_{att})|_{TRIDNT} = \left(E(\tau_{att})|_{selfish}\right) + \left(E(\tau_{att})|_{malicious}\right)$. And the decreasing rate of $E(\tau_{att})|_{malicious}$ is higher than the increasing rate of $E(\tau_{att})|_{selfish}$, so the overall expected attempt time of TRIDNT protocol is decreased as the ratio of selfish nodes to the overall malicious nodes increased.

The relation between the expected attempt time and packet arrival rate is shown in figure 5-c. As shown the TRIDNT expected attempt time is the smallest and has the smallest increasing rate, due to the reasons stated above.

The expected attempt time versus the number of network nodes is shown in figure 5-d. in this figure we find that as the number of network nodes increased the expected attempt time decreases because as N increased the probability of finding a malicious node in the routing path decrease $P = \frac{a}{N}$, at a fixed number of misbehaving node which decreases the expected attempt time. From this figure we see that TRIDNT has the smallest expected attempt time at small number of network nodes and little smaller than the TWOACK minimum expected attempt time at a large number of network nodes.

The relation between the expected attempt time and the average path length is shown in figure 5-e. The figure shown that the increasing rate of expected attempt time for TWOACK and Muhammad Zeshan protocols is greater than the increasing rate of TRIDNT expected attempt time, this done because as the average path length increases the queuing delay increased which has greater effect on increasing the expected attempt time of TWOACK and Muhammad Zeshan protocols.

In figure 6-a we compare between the expected attempt time and the attempt time until blocking as function on the number of misbehaving node. In figure 6-a at a small number of misbehaving nodes the three protocols have an expected attempt time smaller than the attempt time until blocking, so them will found a misbehaving free route with very small blocking probability. Also as the number of misbehaving node increased the value of the expected attempt time will be closer to the value of the attempt time until blocking, which means that the blocking probability is increased.



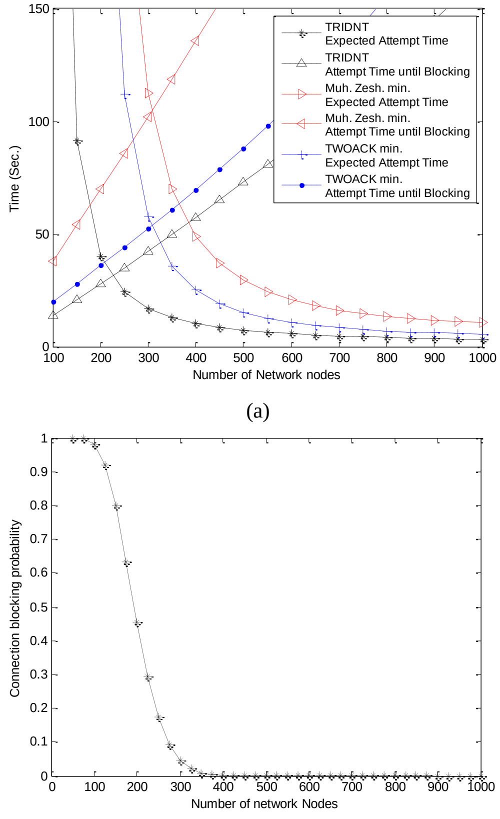
(a)

(b)

Fig. 6 (a) Comparison between the expected attempt time, and attempt time until blocking, for TRIDNT, TWOACK and Muhammad Zeshan protocols. (b) Limited-attempts-based connection blocking probability versus the number of misbehaving nodes.

And at a maximum number of misbehaving nodes the value of expected attempt time exceed the value of attempt time until blocking, so the connection will be blocked. The phenomena showed in figure 6-a matches the limited-attempts-based connection blocking probability curve in figure 6-b. from figure 6-a we can observe that the expected attempt time equal the attempt time until blocking when the number of network nodes equal 140 nodes for Muhammad Zeshan protocol, 145 nodes for TWOACK protocol and 225 for TRIDNT, which mean that TRIDNT protocol has the lowest limited-attempts-based connection blocking probability.

Figure 7-a show a comparison between the expected attempt time and the attempt time until blocking as function on the number of network nodes.



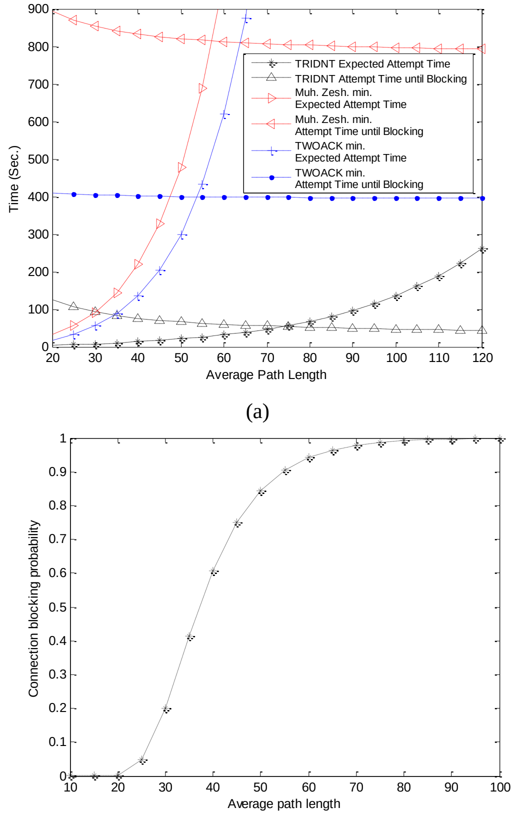
(a)

(b)

Fig. 7 (a) Comparison between the expected attempt time , and attempt time until blocking, for TRIDNT, TWOACK and Muhammad Zeshan protocols versus the number of network nodes. (b) Limited-attempts-based connection blocking probability versus the number of network nodes.

Figure 7 shown that at a small number of network nodes the expected attempt time is higher than the attempt time until blocking, so the connection will blocked and this done because at a small number of network nodes the probability of finding a misbehaving node in the path increased. And as the number of network nodes increased the expected attempt time until finding a misbehaving free route is become smaller than the value of attempt time until blocking, which means the blocking probability will decreases and this match the result in figure 7-b. Also we can see that TRIDNT will success to find a route with low limited-attempts-based connection blocking probability at a smaller number of network nodes (225 nodes) than TWOACK and Muhammad Zeshan protocols do (number of network nodes = 310 nodes) , which mean that TRIDNT protocol has the lowest limited-attempts-based connection blocking probability.

A comparison between the expected attempt time and the attempt time until blocking versus the average path length is shown in figure 8.

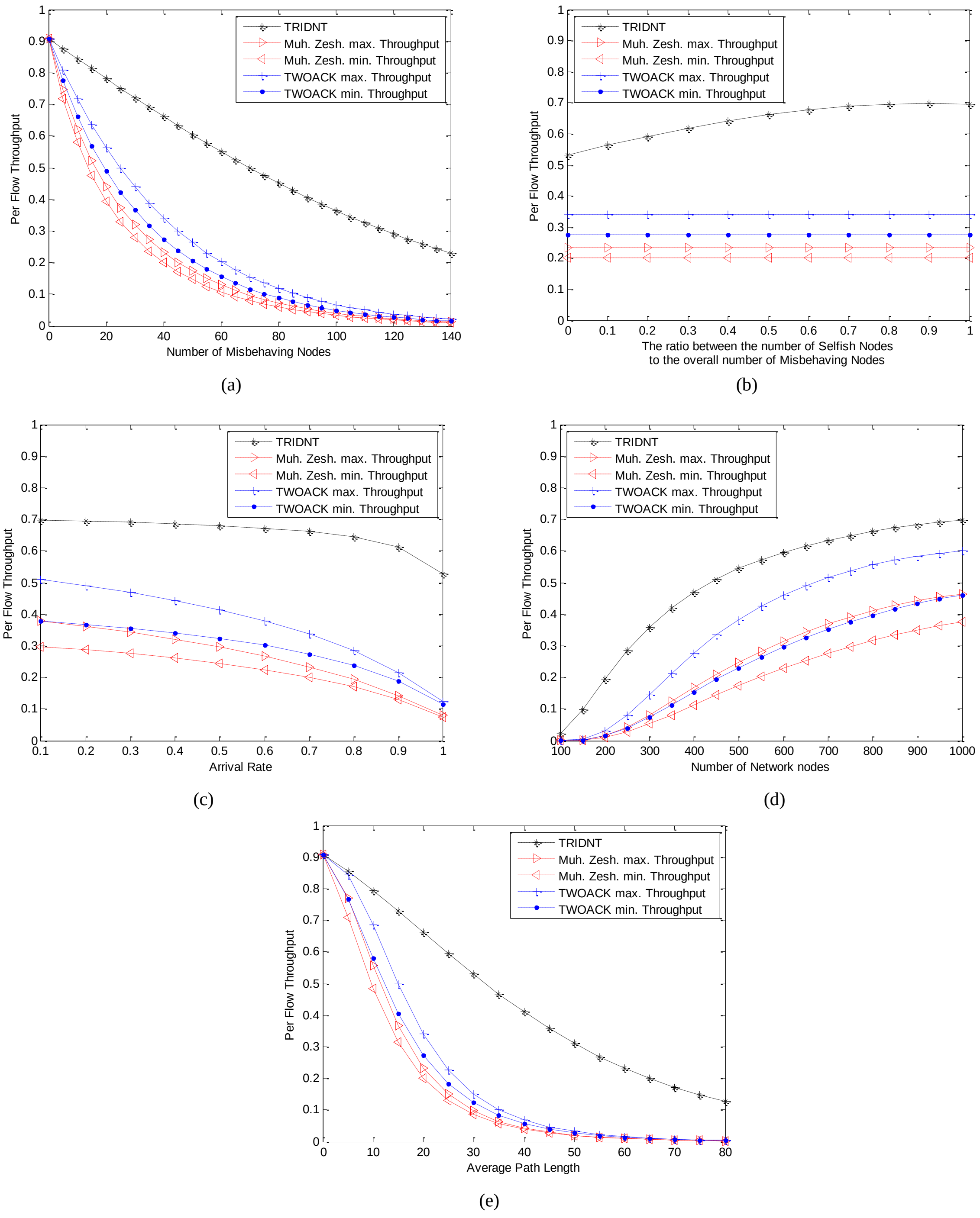


(a)

(b)

Fig. 8 (a) Comparison between the expected attempt time , and attempt time until blocking, for TRIDNT, TWOACK and Muhammad Zeshan protocols versus the number of network nodes. (b) Limited-attempts-based connection blocking probability versus the number of network nodes.

Figure 8-a shown that at a small value of average path length the value of attempt time until blocking is higher than the value of expected attempt time until finding a malicious free route because at small value of average path length the number of available paths is high, so the blocking probability is small. Also we can see from figure 8-a as the average path length increased the expected attempt time increased due increasing the probability of finding a malicious node in the route, and the attempt time until blocking is decreased because the number of available paths decreases, so the blocking probability will increase. This will done until the value expected attempt time exceeds the value of attempt time until blocking at average path length equal 55 nodes for TWOACK and Muhammad Zeshan protocol and 75 nodes for TRIDNT, which mean that TRIDNT has a smaller limited-attempts-based connection blocking probability than TWOACK and Muhammad Zeshan protocols.



(a)

(b)

(c)

(d)

(e)

Fig. 9 Comparison of TRIDNT, TWOACK and Muhammad Zeshan against per flow throughput

Figure 9 show the comparison between the per flow throughput for TRIDNT protocol, and TWOACK and Muhammad Zeshan protocols. In this figure we can see that TRIDNT protocol has the higher per flow throughput than TWOACK and Muhammad Zeshan protocols, because it has the smallest expected attempts time and the per flow throughput is inversely proportional to the value of expected attempt time.

In figure 9-a, the relation between per flow throughput and the number of misbehaving nodes is shown, and we can see that per flow throughput is inversely proportional to the number of misbehaving nodes. This done because as the number of misbehaving nodes increases the number of attempts until finding a misbehaving free route increases, which increases the time of zero data flow which decrease the per flow throughput.

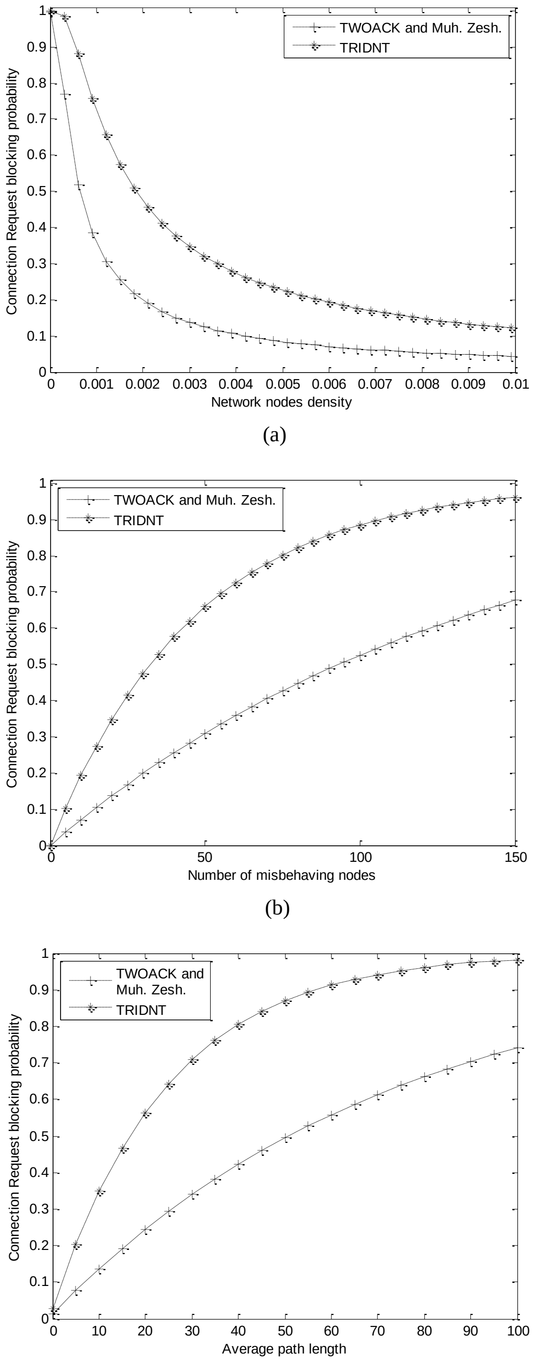
Figure 9-b show the variation of per flow throughput as a function of the ratio between the number of selfish nodes to the overall number of misbehaving nodes as seen the TWOACK and Muhammad Zeshan protocols have a fixed value of throughput because its expected attempt time of them not depend on the ratio between the number of selfish nodes to the overall number of misbehaving nodes. Also we can see that per flow throughput value of TRIDNT protocol changes from 53.26% at k=0 to 69.44% at k = 1, because TRIDNT, which is highest value of per flow throughput.

The relation between per flow throughput and packet arrival rate is shown in figure 9-c. We see that the TRIDNT has the maximum per flow throughput with the smallest decreasing rate.

Per flow throughput versus the number of network nodes is shown in figure 9-d. In this figure we see that as the number of network nodes increased the per flow throughput increases due to the decrease of the probability of finding a misbehaving node in the routing path, and TRIDNT have the higher per flow throughput value at a small number of network nodes, and TWOACK maximum per flow throughput is little smaller than the TRIDNT per flow throughput at a large number of network nodes.

figure 9-e shown the relation between per flow throughput and the average path length. The figure shown that the decreasing rate of per flow throughput for TWOACK and Muhammad Zeshan protocols is greater than the decreasing rate of TRIDNT per flow throughput, this done because TWOACK and Muhammad Zeshan protocols expected attempt time increasing with rate higher that of TRIDNT due to the increasing value of queuing delay.

Our comparison in figure 10 was carried out with mobile nodes moving in a 1000×1000 m^2 flat area with density equal 0.003 nodes per m^2, and 20 expected misbehaving nodes. Each node's transmission range is 250 m with average path 10 nodes length. This comparison show that both TWOACK and Muhammad Zeshan algorithms have a smaller connection request blocking probability than TRIDNT scheme, the reason behind that is TRIDNT uses two trusted node-disjoint routes between the source and destination node which have a smaller probability of occurrence than finding one trusted route as in TWOACK and Muhammad Zeshan algorithms.



(a)

(b)

(c)

Fig. 10 Comparison of TRIDNT, TWOACK and Muhammad Zeshan against Connection request blocking probability

The relation between network nodes density and connection request blocking probability is shown in figure

10-a. We see that for low dense network the misbehaving nodes success in partition the ad hoc network into small portion witch increase the probability of connection request blocking for path length greater than one hob, but as network density increased the connection request blocking probability decreases because there will be a lot of alternate route witch increase the chance of finding a trusted routes.

Figure 10-b shows the relationship between the connection request blocking probability and the number of misbehaving nodes at medium dense network. In this figure we see at small value of misbehaving nodes the ad hoc network connection request blocking probability is very small, here the misbehaving node can’t disrupt the correct function of the network, but as the number of misbehaving nodes increases the TRIDNT scheme connection blocking probability increase with rate higher than that of TWOACK and Muhammad Zeshan algorithms because as the number of misbehaving node increases the process of finding one trusted route become a hard problem and the process of finding a two trusted node-disjoint routes become impossible.

In figure 10-c the relation between the connection request blocking probability and the average path length is shown at medium dense network, and medium number of misbehaving nodes. We see that the misbehaving nodes success on partitioning the ad hoc network which give the process of finding a path of low number of intermediate nodes the probability of success than that path of high number of intermediate nodes.

6. Conclusion

This paper presents an analytical model and performance evaluation of the TRIDNT ad hoc trust base routing protocol. The mathematical analysis shows that the TRIDNT protocol reduces the time required to detect the misbehaving node in the routing path, especially in the network having medium and high traffic intensity. TWOACK and Muhammad Zeshan protocols have a small minimum misbehaving searching time under low traffic intensity, or small number of network nodes.

TRIDNT protocol give the incentive to the selfish node to declare its selfishness behavior which reduce the diagnosis time taken to know if there is a selfish node in the routing path, and also saving the time taken until detect the selfish node. So the expected attempt time until finding a misbehaving free route of TRIDNT protocol is smaller than the minimum expected attempt time until finding a misbehaving free route of Muhammad Zeshan and TWOACK protocols.

By comparing the expected attempt time until finding a misbehaving free route and the expected attempt time until the connection is blocked of the three protocols with the limited-attempts-based connection blocking probability curve, we found that TRIDNT limited-attempts-based connection blocking probability is smaller than the minimum limited-attempts-based connection blocking probability of the TWOACK and Muhammad Zeshan protocols.

Because per flow throughput is inversely proportional to the expected attempt time until finding a misbehaving free route, and TRIDNT protocol has the smallest value of the expected attempt time, so TRIDNT protocol has a higher per flow throughput value than the TWOACK and Muhammad Zeshan protocols.

TRIDNT protocol has a higher connection request blocking probability than TWOACK and Muhammad Zeshan protocols because it searching for a two node-disjoin routes between the source and destination nodes to work but TWOACK and Muhammad Zeshan protocols needs only one route between the source and destination. And the connection request blocking probability has a greater effect especially in case of low dense network, high average path length, and high number of misbehaving nodes.

Note that as the traffic intensity increase the malicious detection time of TWOACK and Muhammad Zeshan protocols will increased exponentially while the malicious detection time of TRIDNT remain unchanged, which will increases the expected attempt time and decreases the per flow throughput of TWOACK and Muhammad Zeshan protocols with high values, so TRIDNT protocol is the best choice under the high traffic intensity conditions.

Therefore, TRIDNT protocol success to find a misbehaving free route faster than the other two protocols, differentiating between selfish and malicious nodes which reducing the expected attempt time until finding a malicious free route, reducing the limited-attempts-based connection blocking probability, and increase the per flow throughput especially in high dense networks having a medium number of misbehaving nodes, and under medium and high traffic intensity condition.

In the future we will simulate the proposed TRIDNT protocol and compare the simulation result with the mathematical results. A detailed simulation evaluation will be conducted in terms of Routing Packet Overhead, Security Analysis, Mean Time to detect dropper node, Overall Network Throughput, and Average Latency. Also we will find the optimum value of node selfishness threshold, and we will extend our protocol to deal with the case of more than one misbehaving node in the routing path.

References

[1] Y. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing," IEEE Security & Privacy, 2004, pp. 28-39.

[2] Abusalah, L. Khokhar, A. Guizani, M. " A survey of secure mobile Ad Hoc routing protocols," Communications Surveys & Tutorials, IEEE, Volume: 10 Issue: 4, Jan 2009, On page(s): 78 – 93.

[3] Dongbin Wang, Mingzeng Hu, Hui Zhi, "A Survey of Secure Routing in Ad Hoc Networks," The Ninth International Conference on Web-Age Information Management (waim), pp. 482-486, 2008.

[4] Rajendra Prasad Mahapatra, SM IACSIT, and Mohit Katyal " Taxonomy of Routing Security for Ad-Hoc Network," International Journal of Computer Theory and Engineering, Vol. 2, No. 2, PP. 303-307, April 2010.

[5] B. Vaidya, S. S. Yeo , D.-Y. Choi , S. Jo Han, " Robust and secure routing scheme for wireless multihop network, " Personal and Ubiquitous Computing magazine , 4 April 2009. © Springer-Verlag London Limited 2009.

[6] Y-C. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks, " In Wireless Networks Journal 11, pp.21-38, 2005.

[7] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Neil Levine, C. Shields, and E. M. Belding-Royer, "Authenticated Routing for Ad Hoc Networks, " Proceedings of IEEE journal on selected areas in communications, Volume 23, No. 3, March 2005.

[8] S. Buchegger and J.-Y. LeBoudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks), " In Proceedings of the 3rd ACM International Symposium on Mobile and Ad Hoc Networking & Computing (MobiHoc 2002), PP. 226–236, Lausanne, Switzerland, June 2002.

[9] Wei Gong, Zhiyang You, Danning Chen, Xibin Zhao, Ming Gu, and Kwok-Yan Lam" Trust Based Routing for Misbehavior Detection in Ad Hoc Networks," JOURNAL OF NETWORKS, VOL. 5, NO. 5, PP. 551-558, MAY 2010.

[10] Marianne A. Azer, Sherif M. El-Kassas, Abdel Wahab F. Hassan, Magdy S. El-Soudani, "A Survey on Trust and Reputation Schemes in Ad Hoc Networks," Third International Conference on Availability, Reliability and Security, PP. 881-886, 2008.

[11] I Aad., , J.-P. Hubaux, E.W. Knightly, "Impact of Denial of Service Attacks on Ad Hoc Networks, ", IEEE/ACM Transactions on Networking, Volume 16, Issue 4, pp. 791 – 802, Aug. 2008.

[12] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," in Proceedings of Mobicom, 2004.

[13] Ahmed M. Abd El-Haleem, Ihab A. Ali, Ibrahim I. Ibrahim, and Abdel Rahman H. El-Sawy "TRIDNT: Isolating Dropper nodes with some degree of Selfishness in MANET" The First International Conference on Computer Science and Information Technology January 2 ~ 4, 2011. Bangalore, India. CCSIT 2011, Part I, CCIS Volume 131, pp. 236–247, 2011. © Springer-Verlag Berlin Heidelberg 2011. [Online] Available: http://www.springerlink.com/content/v46830531212j154/

[14] Ahmed M. Abd El-Haleem, Ihab A. Ali, Ibrahim I. Ibrahim, and Abdel Rahman H. El-Sawy "Trust Model for TRIDNT Trust Based Routing Protocol" 2nd International Conference on Computer Technology and Development (ICCTD) 2010, 2-4 Nov. 2010 Cairo, Egypt, PP. 538 – 544, @ IEEE 2010. [Online] Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5645954.

[15] K. Balakrishnan, J. Deng, and P.K. Varshney. "Twoack: preventing selfishness in mobile ad hoc networks,". In The IEEE Wireless Communication and Networking Conference(WCNC'05), pp. 2137-2142, New Orleans, LA,USA, March 2005.

[16] Muhammad Zeshan, Shoab A. Khan, Ahmad Raza Cheema, Attique Ahmed, "Adding Security against Packet Dropping Attack in Mobile Ad Hoc Networks," International Seminar on Future Information Technology and Management Engineering 2008, pp. 568 – 572.

[17] Mahesh K. Marina, and Samir R. Das "Ad hoc on-demand multipath distance vector routing," Wirel. Commun. Mob. Comput. 2006; pp. 969–988, Published online in Wiley InterScience (www.interscience.wiley.com).

[18] Nasipuri, A. Das, S.R. " On-demand multipath routing for mobile ad hoc networks," Computer Eight International Conference on Communications and Networks 1999, pp. 64–70.

[19] N. Bisnik, A. Abouzeid "Queuing network models for delay analysis of multihop wireless ad hoc networks," International Conference On Communications And Mobile Computing Pages: 773 – 778.

[20] Panichpapiboon, S.; Ferrari, G.; Wisitpongphan, N.; Tonguz, O.K " Route reservation in ad hoc networks: is it a good idea?," Wireless Communications and Networking Conference, IEEE 13-17 March 2005, Vol. 4, PP. 2045 – 2050.

[21] IEEE Standard for Information technology Part 11: "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 12 June 2007 http://standards.ieee.org/getieee802/802.11.html.

[22] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy, "PATHS: analysis of path duration Statistics and their impact on reactive MANET routing protocols," in Proceedings of Mobihoc, 2003.

[23] Teresa Albero-Albero, Víctor-M. Sempere-Payá and Jorge Mataix-Oltra, "Study of the Path Average Lifetime in Ad Hoc Networks Using Stochastic Activity Networks ," the 16th International Conference on Analytical and Stochastic Modeling Techniques and Applications 2009, LNCS 5513, pp. 71–88, 2009 © Springer-Verlag Berlin Heidelberg 2009.

[24] S. Waharte, R. Boutaba," On the probability of finding non-interfering paths in wireless multihop networks" 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet 2008, PP. 914-921.

Biography

Eng. Ahmed M. Abd El-Haleem is a Teacher Assistant in Communication department at Helwan University, was born in Egypt in 1979. He obtained his B.Sc., and M.Sc. degrees from Helwan University, Egypt in 2001 and 2006 respectively. He has long experience in teaching and research. His research interests include Computer Networks, and Secure Routing Protocols.

Dr. Ihab Ali, was born in Egypt in 1962. He obtained his B.Sc., M.Sc. and Ph.D. from Helwan University, Egypt in 1985,1991 and 1997 respectively. He has long experience in teaching and research at different institutions. He is a senior member of IEEE. His research interests include Computer Networks, Network Security and Secure Routing Protocols.

Prof. Ibrahim Ismail is currently a professor at Communications Engineering Dept., Helwan University. He obtained his B.Sc. from Helwan University, Egypt at 1976, M.Sc. from Cairo University, Egypt at 1983 and Ph.D. from Queen University of Belfast, United Kingdom at 1987 respectively.