TRIDNT: THE TRUST-BASED ROUTING PROTOCOL WITH CONTROLLED DEGREE OF NODE SELFISHNESS FOR MANET

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

[1] Assistant Lecture, Communication Department, Faculty of Engineering, Helwan University, Helwan, ahmed_abdelkhaliq@h-eng.helwan.edu.eg
[2]Associate Professor, Communication Department, Faculty of Engineering, Helwan University, Helwan, ehab_ali02@ h-eng.helwan.edu.eg

ABSTRACT

In Mobile ad-hoc network, nodes must cooperate to achieve the routing purposes. Node misbehaviour due to selfish or malicious intention could significantly degrade the performance of MANET because most existing routing protocols in MANET are aiming at finding most efficiency path.

In this paper, we propose a Two node-disjoint Routes protocol for Isolating Dropper Node in MANET (TRIDNT) to deal with misbehaviour in MANET. TRIDNT allows some degree of selfishness to give an incentive to the selfish nodes to declare itself to its neighbours, which reduce the misbehaving nodes searching time. In TRIDNT two node-disjoint routes between the source and destination are selected based on their trust values. We use both DLL-ACK and end-to-end TCP-ACK to monitor the behaviour of routing path nodes: if a malicious behaviour is detected then the path searching tool starts to identify the malicious nodes and isolate them. Finally by using a mathematical analysis we find that our proposed protocol reduces the searching time of malicious nodes comparing to the route expected life time, and avoids the isolated misbehaving node from sharing in all future routes, which improve the overall network throughput.

KEYWORDS

Ad Hoc Network – Trust-Based routing – Secure Routing Protocol – network security.

1. INTRODUCTION

Mobile Ad Hoc Network (MANET) is an infrastructure-less network, consisting of a set of mobile nodes without any support of base stations or access points. The mobile nodes are free to change their position with any speed and at any time, and they play the role of terminals and routers allowing hop by hop communication among nodes outside wireless transmission range. For lack of network infrastructure, the nodes have to communicate cooperatively. Cooperation at the network layer means routing and forwarding packets. Some nodes may deviate from the protocol for selfish or malicious reasons, these nodes are called misbehaving nodes. Selfish nodes wish to use system services while taking an advantage of saving their resources by deviating from regular routing and forwarding. Malicious nodes wish to mount an attack to either a specific node or the network as whole. Both selfish and malicious nodes disrupt the routing protocol operation and reduce the network throughput. This brings up the need for secure routing protocols, where the Routing protocols must cope with such selfish and malicious behaviour.

DOI : 10.5121/ijnsa.2011.3313

Several routing protocols have been proposed in the literature (see [1], [2], [3]). These focus mainly on efficiency issues such as scalability with respect to network size, traffic load, mobility, and on the adaptability to network conditions such as link quality and power requirements. Some of the proposed routing algorithms also address security issues by using cryptographic tools to secure the routing protocol messages (e.g., [4], [5], [6], [7] for a survey, see [8], [9]). Recently, a new class of routing protocol has been proposed, namely trust based routing as in [10]. Trust based routing protocols consist of two parts: a routing part and a trust model, for a survey see [11]. 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 neighbours; which violate the TCP protocol rules.

This paper focuses on Packet Dropping Attack, and presents a novel routing algorithm resistant to various packet dropping scenarios. Here, 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 neighbour nodes to delay the detection of the attack and hence further decrease the throughput. In this paper, four packet dropping scenarios are considered. In Inclusive Packet Dropping, the malicious node simply drops all received data link layer (DLL) PDU's while positively acknowledging them. This attack is also called Black Hole attack [12], [13], [5]. Periodic Packet Dropping is used by malicious nodes to drop a small fraction of incoming DLL PDU's once per retransmission time out, a variant of Jellyfish (JF) attack reported in [12], [13], [14]. In Frequent Packet Dropping, the malicious node may possibly drop a fraction of incoming DLL PDU's on a random basis. In Selective Packet Dropping, the malicious node drops only these PDU's coming from specific source(s), going to specific destination(s), or following a specific route. The last two attacks are called Gray Hole attack. In all packets dropping attacks scenarios, the overall network throughput will deteriorate [12].

In our proposed scheme we establish two node-disjoint routes between the source and destination nodes, these routes have the highest path trust values; to route around misbehaving nodes; one is marked as primary and the other as secondary. Unlike all previous research efforts made to tolerate Packet Dropping Attacks, our work allow some degree of node selfishness; to save their resources partially; and detect the malicious activity faster. We use both DLL-ACK and end-to-end TCP-ACK as monitoring tools; without continuous promiscuous monitoring of the neighbours; and when detecting a malicious activity a new path searching technique is used to identify the malicious or compromised nodes in the routing path and isolate them. Based on this claim, the proposed scheme detect the misbehaving node and avoids it from sharing in all future routes in a few seconds lower than the route expected life time, resulting in an improved overall throughput performance for the network.

The rest of the paper is organized as follows. Section 2, describes the related work. The network assumptions and the TRIDNT operation are presented in Section 3. A time taken to detect the malicious node in the routing path is calculated in section 4, and the performance result is presented in section 5. Finally we conclude our work and discuss our plan for future work in section 6.

2. RELATED WORK

In [15] Marti et al. proposed a mechanism called as watchdog and pathrater on DSR to detect the misbehaving nodes in MANETs. The approach introduces two extensions to DSR: A watchdog detects misbehaving nodes, by maintaining a buffer of transmitted packets and overhearing of other node forwarding. It compares each overheard packet with the packets in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node. A pathrater avoids routing packets through the detected malicious nodes. Each node estimates a link metric with respect to the reliability of links and knowledge about misbehaving nodes. A node assigns this metric to every other known node and periodically updates the metric. The downside of their method is that they cannot distinguish the misbehaving nodes from node failures. An honest node can easily be rated malicious if the transmission breaks up.

CONFIDANT [16] is a protocol which also attempts to detect the malicious nodes in ad hoc networks. Monitor, Reputation System, Path Manager and Trust Manager are the main components of CONFIDANT protocol. For each packet a node forwards, the monitor on that node attempts to ensure that the next-hop node also forwarded the packet correctly (overhearing). When the monitor detects an anomaly, it triggers action by the reputation system, which maintains a local ratings list. These lists are potentially exchanged with other nodes; the trust manager handles input from other nodes. Finally, the path manager chooses paths from the node's route cache based on a blacklist and the local ratings list. CONFIDANT has scalability problems with the number of nodes. The tables maintained by the reputation system of each node may become huge. Also, in scenarios with very high mobility, the overhead can increase considerably.

In [17] 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. 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. 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 future.

Muhammad Zeshan et al, [18] proposed a two folded approach, to detect and then to isolate a malicious node causing packet dropping attacks. First approach will detect the misbehaviour 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. Then upon identification of misbehaving nodes in network other approach will isolate the malicious node from network. All nodes which lie in the transmission range of active route 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, the monitoring node in that range declares that node as misbehaving node. 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 Misbehaviour).

3. THE PROPOSED TRIDNT PROTOCOL

In this section we describe our solution to address the Packet Dropping Attack in MANETs. The proposed protocol makes the first effort to distinguish between the malicious and selfish node, and allow a controlled degree of node selfishness. The proposed monitoring tool detects the malicious activity and then the path searching tool identifies the malicious or compromised nodes in the network and isolates them, and the proposed routing protocol routes around the misbehaving node.

3.1. Network Model and Assumptions

In this work, we assume that the MANET consist of *N* nodes are situated in a bounded 2-dimensional space, within which they are free to move, and a bi-directional communication symmetry on every link between the nodes. For simplicity we also assume that the destination-node is non-malicious, and any routing path contains on the average of h node has at most one malicious node.

Mobile nodes in MANETs often communicate with one another through an error-prone, bandwidth-limited, and insecure wireless channel. We are not concerned with the security problem introduced by the instability of physical layer or link layer. We only assume that: **(1)** Each node in the network has the ability to discover all of its neighbours; **(2)** Each node in the network can broadcast some essential messages to its neighbours with high reliability; **(3)** Each node in the network uses its MAC address as a unique identifier (node ID); **(4)** Each node in the network have a black list containing the misbehaving nodes, a trust table containing the learned network nodes' trust value; which are broadcasted to the node's neighbours periodically; and a Data Packet Information (DPI) cache to store information about the received and processed data or TCP-ACK packets.

In the network layer, a new node model is designed as the basis of our trust model. Some new fields are added into a node's routing table to store its trust value about other nodes and to record the positive and negative ratings when it performs routing with others.

3.2. Operation of TRIDNT

In TRIDNT we use AOMDV [19], or multipath DSR [20] to establish a two node-disjoints paths between the source and destination nodes. And with a little modification of the RREQ packet to contain a list of unwanted nodes, which the source node doesn't want them to be

members on the discovered route temporarily, also the destination node may have this list (as seen below) and it discard all routes which contain this unwanted nodes.

Also during the RREQ flooding process the intermediate nodes will insert the previous node trust ratings in the RREQ packet if the previous node trust value *T* is less than the trust value contained in the RREQ packet.

When the destination node receives RREQ packet from multiple nodes, it selects two node disjoint paths with the highest path trust value, and certainty factor and unicasts two RREPs (contain the path trust rating) back to the source along the selected two routing paths. We will take the path trust value as the minimum trust value among all links in the routing path, and can be calculated as:

$$T_p = \min\left(T_{S,n1}, T_{n1,n2}, T_{n2,n3}, \ldots, T_{ni,D}\right) \qquad (1)$$

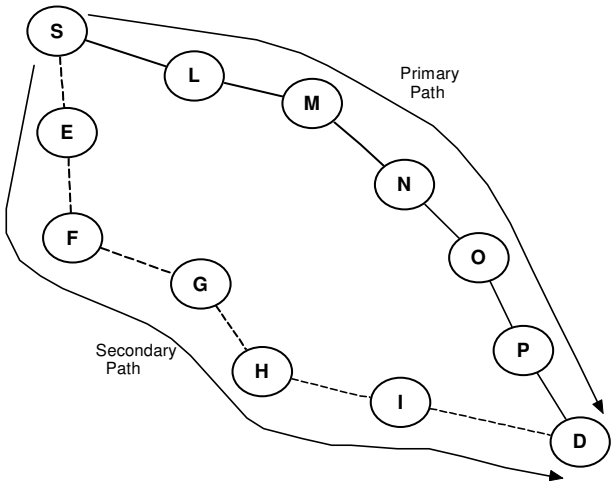


Figure 1. Two node disjoint paths between S and D

The source node marks the highest trusted route as primary used for data forwarding and the other as secondary used as a backup path, as shown in figure 1. The two node-disjoint routes are adopted to ensure reliable communication and search for malicious nodes.

3.2.1 Controlled selfishness behaviour

The misbehaving node may be a selfish or malicious node, selfish node will hurt the network connectivity and is reported as malicious node in all reported trust based routing protocols. We use the observation of that, there is a difference in needs of selfish and malicious nodes, where selfish node needs: (1) to use network resources, (2) save its resources "drop any forwarded packet form other nodes and don't want to be a member in any new routes". But the malicious "dropper" node needs: (1) to be a member in all new routes, (2) mount a denial of service attack by dropping the data packets it receives.

Depending on the different needs of selfish and malicious nodes, we will allow 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).

A new field is inserted in the Hello packet containing the selfishness status. Each node use this field in Hello packet to inform its direct neighbour nodes about its selfishness status, if it is in selfish mode all neighbour nodes 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 coming from it.
3) Forward to/from it the packets which contain it as destination/source address.

The selfish node neighbours will restrict its selfishness behaviour by a time threshold, and a repetition threshold.

By allowing some degree of node selfishness the selfish node declare itself to its neighbour, and malicious node will not declare itself as selfish node because of inconsistency with its needs. So the selfish nodes are excluded from the responsibility of data forwarding. At the same time, this helps in easier identification of malicious nodes. 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 needs to be very small because; due to the node mobility; the route life time is small.

3.2.2 Route monitoring toll

In our approach we use the DLL-ACK and the end to end TCP-ACK as monitoring tool to monitor the behaviour of the routing path, then use a path searching tool to search the misbehaving path to find the malicious node, and then put the malicious node ID in the black list to isolate it.

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 neighbour, and wait for a data link layer acknowledgment (DLL-ACK) from the neighbour node, if it did not receive data link layer acknowledgement; it concludes that this neighbour node should be down. In such case, the neighbour is excluded from the node's routing table until it becomes up. However the neighbour's trust rating doesn't change. 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:

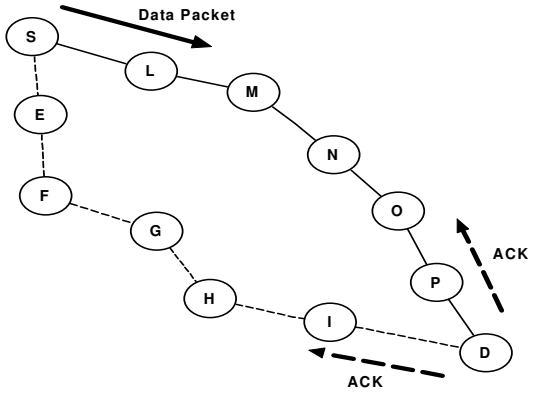


Figure 2. The source node sends the data on the primary path only, and the destination replay with ACK on both primary and secondary paths

Case I: if there is no malicious node in the primary and secondary paths, then the source node receive the TCP-ACK over the two routes (primary and secondary) as shown in figure 2.

Then the source node sends a biggy back Positive Trust Update Message (PTUM) upon transmitting the next packet. If the each of the nodes responsible for forwarding this message received an acknowledgment from a neighbour node in the data link layer and through this neighbour in the transport layer, and received PTUM message from the source node, then each node in the primary path will update the trust value of its upstream and downstream neighbours, and remove the information about the confirmed data packet from its DPI cache. Also destination node will send a biggy back PTUM message when transmitting the next TCP-ACK packet to the source node, to update the trust value of nodes in the secondary path.

Case II: if the source node received an acknowledgment from a neighbour in the data link layer and receive an acknowledgment in the transport layer over the primary path only, even after retransmitting this message (TCP rules); it concludes that the neighbour node or one of its following nodes in the primary routing path may be malicious node trying to make blocking attack and send a faked TCP-ACK or there is a malicious node in the secondary routing path drop the TCP-ACK packet.

Case III: if the source node received an acknowledgment from a neighbour in the data link layer and did not receive an acknowledgment in the transport layer over the primary or secondary route paths, even after retransmitting this message (TCP rules). Then the source node knows that the data packet doesn't reach its destination, i.e. there is a malicious node in the primary path trying to make blocking attack.

In last two cases II and III the source node run the malicious search mechanism, to find the malicious node.

3.2.3 Route searching mechanism

If the source node concludes that there is a malicious node in the primary or secondary routes 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 toward the destination node.

The MSP packet is a high priority packet, and 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 forwarded it to the next node) it will forward the MSP packet to the next node with overhearing to assure that the neighbour 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 these packets and break the searching and detection process.

Case I: if the primary path contains a malicious node, let node N be the malicious node. The source node sends the MSP packet to node L and overhears to be sure that node L will forward that packet. After comparison, node L forwards the MSP packet to M and overhears, then node M compares and forwards it to node N and overhears. The malicious node N has two choices:

(1) It either, stops forwarding the MSP packet and report the destination node using MDP packet (N, M), that node M is the malicious node; node N deny the receiving of this data packet from node M. At the same time node M sure that it forwarded this data packet to node N and receive a DLL-ACK from node N, where it didn't overhear node N forward MSP packet, then node M report the source node that node N is the malicious node; using MDP packet (M, N) as shown in figure 3-a.

(2) Or, it will forward the MSP packet to node O, then node O didn't find a match in its DPI cache, so it will send the MDP packet (O, N) to the destination node. At the same time the malicious node N can inform the source node that node O don't forward the MSP packet; send a MDP packet (N, O) as shown in figure 3-b.

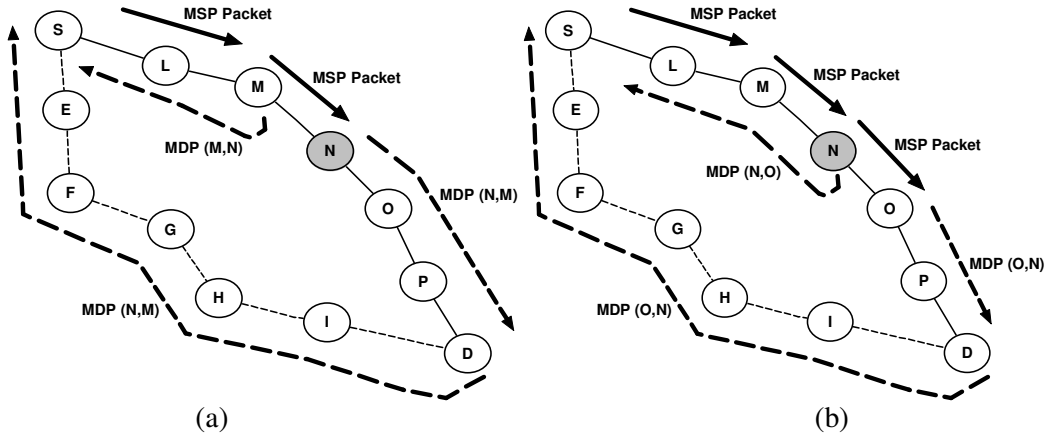
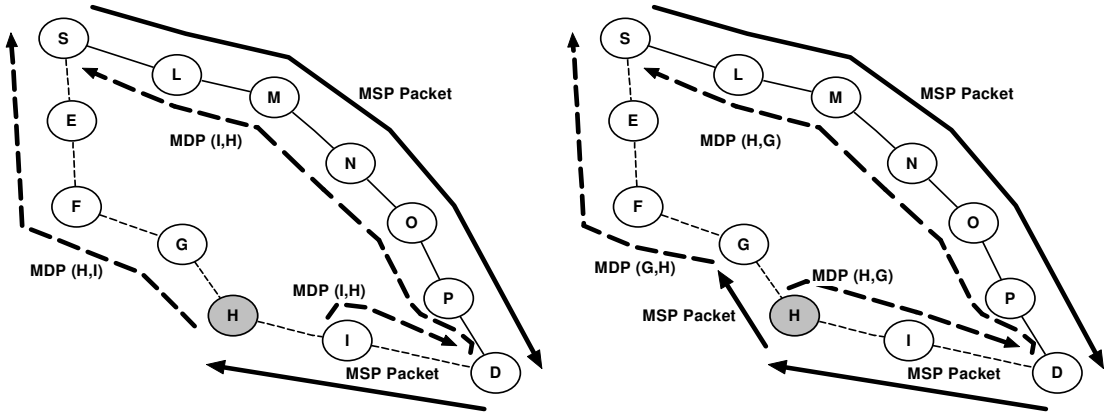


Figure 3. Malicious node N (a) don't forward the MSP packet, (b) forward the MSP packet.

In both cases when the source and destination nodes; and all nodes in the routing path; receive the MDP packet, they will update the trust value of both the detecting and detected node negatively and the trust value of other nodes in the routing path positively. Because the honest node (node M or node O) will suffer from the misbehaviours of malicious node N, so it will insert the malicious node N ID in its black list regardless of its trust score to prevent any future cooperation with it and isolate it from the network.

When the destination node receives the MDP packet it forwards it to the source via the secondary route, on the other hand if the source node doesn't receive an MDP from the destination node via the secondary route it will send the received MDP from the primary route to it until it can discard the suspect nodes from any future selected routes to that source. Finally the source node mark the secondary route as primary and start a route discovery phase to find a secondary node-disjoint route not containing both the detecting and detected node on it.

Case II: if the secondary path contains a malicious node, let node H be the malicious node. The source node sends the MSP packet to node L, and node L forward it to node M → N → O → P → to the destination. When the destination node receives the MSP packet, it will be sure that there is no malicious node in the primary route, and then the destination node will modify the MSP packet to contain the information of TCP-ACK packet and forward it to node I on the secondary path. Node I forward it after comparison to node H (the malicious node) the malicious node H has the same two choices as in case I as shown in figure 4.



(a) (b)

Figure 4. Malicious node H (a) don't forward the MSP packet, (b) forward the MSP packet.

When any node receives the MDP packet, it will update the trust values of both the detecting and detected node negatively, and the trust value of other nodes in the routing path positively. Also the detecting node (node I or node G), will insert node N ID in its black list; regardless of its trust value; to reject any future cooperation between them.

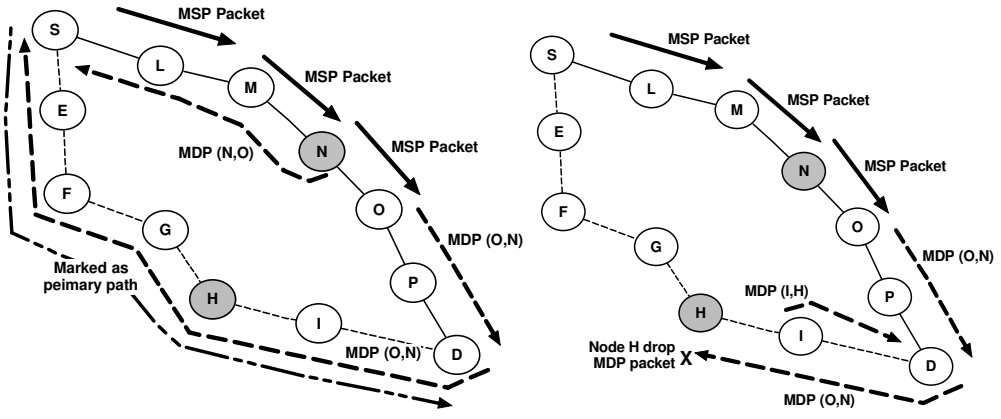
When the destination node receives the MDP packet it forwards it to the source via the primary route, on the other hand if the source node doesn't receive a MDP from the destination node via the primary route it will send the received MDP from the secondary route to it until it can discard the suspect nodes from any future selected routes to that source. Finally the source node

starts a route discovery phase to find a secondary node disjoint route not containing both the detecting and detected node on it.

Case III: if both the primary and secondary routes contain malicious nodes. The source node sends the MSP packet via the primary path and waits the MDP packet, if:

The primary path malicious node send the MDP packet (N, O) to the source node, on the other hand the destination node receive an MDP packet (O, N) from node O, then it will forward it to the source node via the secondary path. If the secondary path malicious node forward the MDP packet (O, N), then the source node receive the MDP packet (O, N) and mark the secondary path as primary and search for new secondary path don't contain both the detecting and detected nodes (M, N), as shown in figure 5-a. Because the new primary route contain also a malicious node, then the source node don't receive TCP-ACK packet from the destination, so it start a new malicious search procedure to find the malicious node. When finding the new malicious node the source node marks the new secondary route as primary and search a new secondary route, and so on.

The primary path detecting node send the MDP packet (O, N) to the destination node, then the destination node send the MDP packet (O, N) to the source node via the secondary path. Figure 5-b shown that there is a malicious node in the secondary path drop the MDP packet (O, N), and don't reach the source node. When node I found that node H drops the MDP packet (O, N), it will send a new MDP packet (I, H) to the destination node. When the source node doesn't receive the MDP packet from both the primary and secondary paths, it will try to run the malicious search again, and if it doesn't receive the MDP again it conclude that there are a malicious nodes in the primary and secondary paths. So the source node flooding a RREQ toward the destination node, and when the destination node receive the RREQs it delete the paths contains nodes (O, N, I, H) and select the two highest trusted paths.



(a) (b)

Figure 5. The source node S (a) receives the MDP packet (b) don't receive the MDP packet.

3.2.4 Malicious node isolation

When a neighbor of a malicious node detect its malicious activity it will send the MDP packet and put the malicious node ID on its black list to isolate it. 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' neighbours 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 neighbours (whose ask the old neighbours about the node reputation), and if its reported trust value is above the trust threshold δ.

4. MALICIOUS SEARCHING TIME ANALYSIS

In this part we will calculate the time required to detect the malicious node in the routing path, which called the malicious detection time τ_{md}.

In the proposed TRIDNT protocol, once the source node known that there is a malicious node in the routing path during the path forwarding and monitoring phase it start using the route searching mechanism to detect the malicious node.

Let both primary and secondary route contains a malicious nodes (worst case), and for simplicity let both paths traversing the same number of *h* relay nodes, which represent a random sample of the *N* network node.

Here the source node *S* will start to find the malicious node by sending 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 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 both MSP and MDP are high priority packets, then it only suffer from propagation delay (mean node service time τ_s). So the overall malicious detection time of TRIDNT protocol is

$$\tau_{md}|_{primary} = 2\{h + h - 1\}\,\tau_s$$

$$= 2\,\tau_s\,(2h - 1) \qquad (2)$$

In [21] the node service time is calculated as the sum of duration of random back off timer $\frac{1}{\varepsilon}$, the duration for which the timer frozen, time of exchange RTS, CTS and ACK packets (IEEE 802.11 MAC protocol delay), and transmission time $\frac{L}{\omega}$. Also author in [21] neglects the RTS, CTS time comparing to transmission time, so the expected value of node service time as in [21] is:

$$\tau_s = \frac{\frac{1}{\varepsilon} + \frac{L}{\omega}}{1 - 4\,N\,A(N)\lambda_i\,\frac{L}{\omega}} \qquad (3)$$

Where L: is the packet size.

ω: node transmission rate.

A(N) = π r(n)2 : node communication area.

r(n) : node transmission range.

λi : effective arrival rate at a station, and the packet generation process at each node is an i.i.d Poisson process with rate λ, $\lambda_i = \frac{\lambda}{\sqrt{\frac{\log N}{N}}}$ [21].

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

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

5. PERFORMANCE RESULT

Through this performance evaluation we assume that the node transmission range $r(n) = \sqrt{\frac{\log N}{N}}$ as stated in [21], the packet size L = 1 K bits, the node transmission rate $\omega = 10^6$ bits/sec , the number of network nodes N = 600 nodes, the average path length h =20 nodes, and Poisson arrival rate at a station $\lambda = 0.5$.

Because the random back off time equal the multiplication of random number by the slot time, as stated in [22], where the slot time = 20 μ Sec, and

$$0 < \text{random number} < CW$$

$$31 \leq CW \leq 1023$$

The contention window 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 $\frac{1}{\varepsilon} = C\,(1 - e^{-\lambda})$ 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 $\frac{1}{\varepsilon} = 10.23\,(1 - e^{-\lambda})$ m sec.

Also will assume that the routing path have an expected life time $E(\tau_i) = 10$ sec corresponding to V_{max} = 30 m/sec as reported in [23], [24].

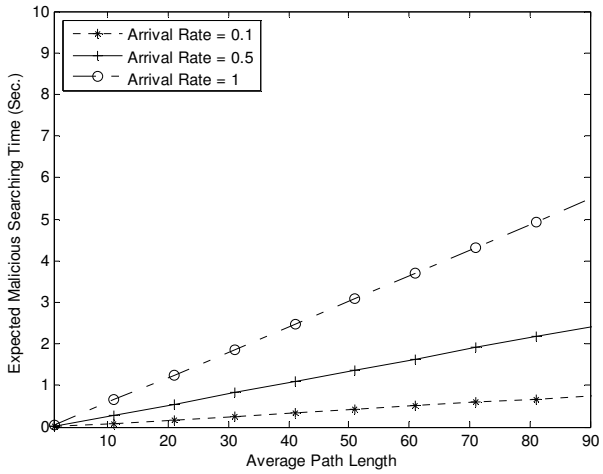


Figure 6. Expected malicious search time vs. the average path length

The expected malicious searching time versus the average path length is shown in figure 6. We can see that as the average path length increases the malicious node searching time increases, 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 (2), and (4). From this figure we can find that TRIDNT protocol can detect the malicious node on a fraction of path life time $\left(\frac{\text{malicious searching time}}{\text{path expected life time}} * 100\%\right)$ changes from 10% at low traffic to 60 % at high traffic; at average path length equal 15% from the total number of network nodes. Also we find that at medium traffic the malicious searching time has an acceptable range, so TRIDNT can find the malicious node in a few seconds which is lower than the expected route life time.

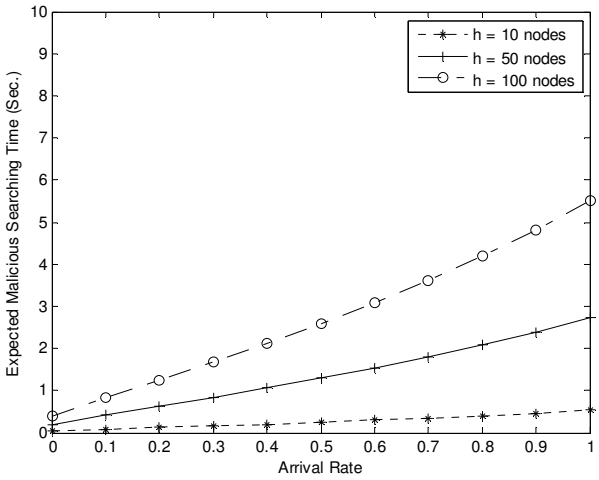


Figure 7. Expected malicious search time vs. the node arrival rate

Figure 7 shows the relation between the expected malicious searching time and the packet arrival rate. From the figure we can see that the malicious searching time increase as the arrival rate increase because the node service time increased as seen in equations (4). Also we can see that TRIDNT protocol can detect the malicious node on a fraction of path life time

$\left(\frac{\text{malicious searching time}}{\text{path expected life time}} * 100\%\right)$ changes from 5% at small number of average path length to 55% at large number of average path length; under a high traffic condition.

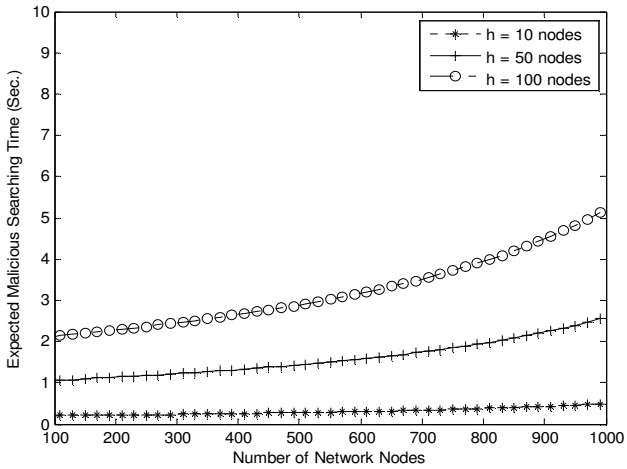


Figure 8. Expected malicious search time vs. the number of network nodes

Figure 8 shows how the expected malicious searching time, varies with the number of network nodes for TRIDNT protocol. 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 (4). Also we can see that at a small number of the average path length (= 10 nodes) the malicious searching time increasing rate equal 0.32 ms/node, but at h =100 nodes the malicious searching time increasing rate equal 3.31 ms/node. This means that at small number of average path length the number of network node has a little impact on the malicious searching time, and this effect increased as the number of the average path length increased.

6. CONCLUSION AND FUTURE WORK

In this paper we proposed a general solution to packet dropping misbehaviour in MANET. The solution allows monitoring, detecting, and isolating the malicious node. In TRIDNT the malicious node neighbours will isolate it after a few numbers of transactions. Also TRIDNT allows a controlled amount of node selfishness behaviour to give an incentive to the selfish nodes to declare its selfishness behaviour to their neighbours, to reduce the searching time of misbehaving nodes to search for malicious nodes only. The mathematical result show that TRIDNT protocol can find the malicious node in a small amount of time comparable to the route expected life time, especially for high dense networks with medium traffic intensity. So we can find an isolate the malicious node; denied access to the network; in small amount of time without using promiscuous listening, which results in an improved overall throughput performance for MANET.

In the future we will design a trust model to calculate the node and path trust values, and define a trustworthy accurate threshold. Also we will simulate TRIDNT to show the results and effectiveness of our solution, and compare it with existing trust based routing algorithms like TWOACK, and Muhammad Zeshan proposed schemes. 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 study the situation when there are more than one malicious node in the route from the source and destination.

ACKNOWLEDGEMENTS

The authors are grateful to Professor Ibrahim I. Ibrahim, and Professor Abdel Rahman H. El-Sawy for helpful comments on this paper.

REFERENCES

[1] D. Johnson and D. Maltz, "Dynamic Source Routing protocol (DSR) for Mobile Ad Hoc Wireless Networks for IPv4," RFC 4728, February 2007. [Online] Available: http://www.ietf.org/rfc/rfc4728.txt.

[2] Perkins CE, Belding-Royer E, Das SR. "Ad hoc on-demand distance vector (AODV) routing, " RFC 3561, July 2003. [Online] Available: http://www.ietf.org/rfc/rfc3561. txt.

[3] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Proc. ACM SIGCOMM, pp. 234-244, 1994.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Yih-Chun Hu, Adrian Perrig, "A Survey of Secure Wireless Ad Hoc Routing," IEEE Security and Privacy, vol. 2, no. 3, pp. 28-39, Mar. 2004.

[10] 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.

[11] 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.

[12] 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.

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

[14] A. Kuzmanovic and E. Knightly, "Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephants," in Proceedings of ACM SIGCOMM, 2003.

[15] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking, PP. 255–265, 2000.

[16] 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.

[17] 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

[18] 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, PP. 568 – 572, 2008,.

[19] 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).

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

[21] 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.

[22] 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.

[23] 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.

[24] 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

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.