International Journal of Computer Networks and Communications Security

VOL. 2, NO. 10, OCTOBER 2014, 339–347
Available online at: www.ijcncs.org
ISSN 2308-9830

SecTORA: Empowerment of TORA Protocol to Deal with the Elimination of Data Packets by Intruder Nodes

Farzad Moradi

Saghez Branch, Islamic Azad University, Saghez, Iran

E-mail: frzmoradi@gmail.com

ABSTRACT

One of the attacks that the intruder nodes carry out by penetrating the network is eliminating the data packets, which results in impairing the network efficiency. The method introduced in this paper aims at detecting this attacks and decreasing their effects. The new method is peculiar to those networks which enjoy TCP protocol in their transmission layer. In the present paper one of the famous routing protocols in ad hoc networks, TORA protocol, was studied and the proposed method was implemented on it. A new protocol, called SecTORA, was introduced by changing basic TORA algorithm, which offered two new characteristics: (1) It discovered the routes with intruder nodes by benefiting from the retransmitting characteristic of TCP protocol and the sequence number field in the TCP packet header, (2) It lessened the harmful effects of intruder nodes on omitting the data packets by sending data packets through main and backup paths interchangeably. The simulations carried out showed that the SecTORA protocol was 17.4 percent more efficient than TORA in the networks intruded by attackers; but, this efficiency was 1.47 percent less for the networks which had not been intruded.

Keywords: *Ad hoc Networks, Routing Protocols, TORA.*

1 INTRODUCTION

In recent years, Mobile Ad Hoc Network (MANET) has become highly popular and lots of research has been carried out on its different aspects. MANETs are a network of mobile nodes (laptops, sensors, etc.) interacting together without a stable central infrastructure [1]. High degrees of freedom and self-made potentials have made ad hoc networks different from others. Users can create an ad hoc network easily and cheaply.

Security is a critical issue to protect the connection among the mobile nodes. Due to unique feature of MANETs, these networks' security faces with many challenges: open network architecture, shared wireless mediums, high constraints of resources and high dynamics of the network topology [2]. The presence of intruder nodes in ad hoc networks can cause reduction of the network's efficiency to a great deal. Therefore, it is quite essential to use mechanisms in order to secure routing protocols especially when 1.There is the possibility of the existence of intruder nodes and 2. Safety and efficiency of the ad hoc network is critical (especially for military use).

In this study, the kind of attack in focus contains intruder nodes passing the first security barrier, the protective methods of attack, and infiltrating into the network. Intruder nodes participate in routing processes, but when they are placed in the forward route of the data packets, they start to eliminate them. There are two types of attack or malicious behavior in question: 1. the intruder node has infiltrated into the network. This node apparently takes part in routing process, route maintenance and cleaning but intentionally removes data packets needed to be pushed forward. 2. Node has got a selfish behavior, meaning that it is part of the network and must cooperate in forward-running process of the packet to destination. However, since this process requires energy, consumption and processing burden, it refuses to do it. An intruder or selfish node is likely to eliminate all or some data packets.

In ad hoc protocols, most of the routing protocols use two different designing approaches for dealing with inherent properties of ad hoc networks. These

two approaches are table-driven approach and on-demand approach [3]. Some examples of first group are Destination Sequenced Distance Vector (DSDV)[4], Optimized Link State Routing (PLSR) [5] and some of second group are: Dynamic Source Routing Protocol (DSRP) [6], Ad Hoc On-Demand Distance Vector (AODV) [7] and Temporally-Ordered Routing Algorithm (TORA) [8].

TORA is one of the famous routing algorithms of the ad hoc networks. This is a multi-path routing protocol which finds various routes towards the destination in the routing process. Nevertheless, the protocols act in this way that one of the routes (the shortest one) is always used to run the packets forward and doesn't react to data packets' removal from the main route. [9] deals with the TORA protocol reaction versus control packets' removal of QRY,UPD and CLR, whereas there is no word of the removal of the data packets. In [10], the use of back-up routes has been stressed as one of the healing strategies of malicious effects of intruder nodes on routing protocols.

This article is aimed at modifying the basic algorithm of TORA to reduce the impact of intruder nodes on the removal of data packets using back-up routes. Resulting from the modification of basic algorithm of TORA, SecTORA reduces the impacts of data packets' removal through reading TCP packet header and using the main and back-up routes. In the following, this algorithm's performance process is mentioned and compared with the main one using simulation.

2 TORA ROUTING PROCESS IN A NETWORK WITHOUT INTRUDER NODES

TORA is an on-demand routing protocol, which means that a route is created only if there is a request around [8].creating a route from a request node towards the destination requires making a series of directed links from the source node towards the destination. The method used to achieve this goal is a request/response process creating a directed acyclic graph directed to the destination (i.e. the destination is the only node bearing no output arrow).

Every time an organized quintuple $H_i = (\tau_i , oid_i , r_i , \delta_i , i)$ is assigned to each of the network nodes and it is called node height [11]. Route creation requires control packets of QRY and UPD. Figure 1 represents the ultimate DAG following route request from node C to F.

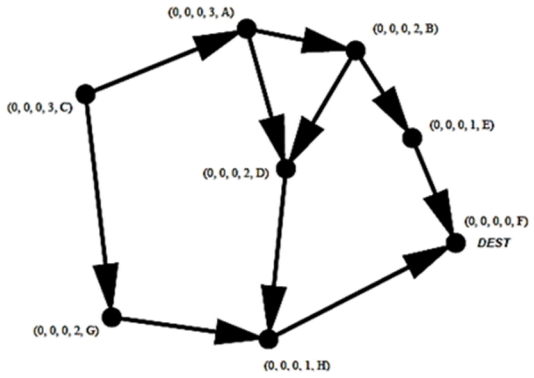

Fig. 1. Final DAG after route request from C to F [11].

In route selection, each node picks out the neighboring one, the shortest among the neighbors. For instance, in Figure 1, among nodes A and G, nodes C selects G for its less height to transfer the packets towards the F destination.

Figure 2 represents a trace file relating to a network of fifty nodes. This file is an output of running ns2 simulation software and shows the undertaken actions at the network and transport layers. The routing protocol used at the network layer, TORA protocol and the employed protocol at the transport layer is TCP. In this case, there is no intruder or selfish node and all nodes are doing duties relevant to routing and packet forward-running actions very well.

In line 1, node 6 intends to transmit some data to node 19. TCP protocol to node 6 generates a packet and transmits it to the network layer. Since there is no route to the node 19 at the moment with network layer, the packet is initially placed in queue (line2) and then TORA creates route request in order to find a route to node 19 (line 3). This packet is spread out generally so that the node containing a route to the destination responds to the packet in need of the route. The entire process of routing has not been shown in the figure due to being lengthy and we assume that routing process is complete in line 6 and the found route is as 19→1→16→27→6.Having detected the route, network layer of node 6 transmits the packet (line7) and the intermediate nodes of 27,16, and 1 run the transmitted packet forward (lines 8, 9, 10), node 19 receives the packet fine (line 11) and creates ack packet to verify the received packet and delivers it to the network layer to be sent towards the source node, that is node 6. TORA protocol in node 19 transmits the route request packet in order to transmit ack (line 14). It shall be noted that ack packet transmitting is not in the opposite direction of the TCP packet transmit route. Rather, the found route of ack back reply is different from transmit

routes of TCP packets. Transmit route of ack packet is as 6→27→21→12→19 in which the intermediate nodes return ack packet to the source node, that is node 6 (lines 19, 20, 21). Finally, node 6 receives the acknowledgment for ntthe transmitted packet (line22).

3 TORA ROUTING PROCESS IN A NETWORK WITH INTRUDER NODES

In the previous example, all nodes are well acted and there is no malicious or selfish node in the network. In the next example, we assume that node 16 is an intruder one infiltrating into the network. The network's performance state has been demonstrated in Figure 3. Node 16 is placed on the forward route of the packet from node 6 to node 19 and eliminates the packet instead of forward-running (line 9). Due to packet elimination, node 19 receives no packet so ack packet won't be produced. Having transmitted the packet, node 6 does not receive any acknowledgment; therefore time dedication for the acknowledgment of the transmitted packet is elapsed. When time duration is up, node 6 has to retransmit the packet (line 6). Retransmitting takes place on the same former route and intruder node of 16 re-eliminates the TCP packet and node 6 inevitably retransmits the packet. The same process is repeated as far as the transmitting route of the packet switches in a way that intruder node 16 is no longer on the same route.

In Figure 3, the transmitter node does not do the repeated transmissions of a packet in equal intervals whose acknowledgment has not been received. Rather, in every post this time interval is doubled and this led to a sharp drop in TCP efficiency. Posting times are 6, 12, 24 and 48 respectively.

```
1   s 5.000000000 _6_ AGT --- 3950 tcp 40 [0 0 0 0] ------- [6:0
2   19:0 32 0] [0 0] 0 4
3   T 5.000000000 _6_ tora enq 6->19
4   T 5.000000000 _6_ tora sendQRY 19
5   T 5.006458085 _37_ tora sendQRY 19
6     ……..
7   T 5.038696131 _32_ tora sendQRY 19
8   s 5.084920420 _6_ RTR --- 3950 tcp 60 [0 0 0 0] ------- [6:0
9   19:0 32 27] [0 0] 0 4
10  f 5.147486852 _27_ RTR --- 3950 tcp 60 [13a 1b 6 800] -------
11  [6:0 19:0 31 16] [0 0] 1 4
12  f 5.167766553 _16_ RTR --- 3950 tcp 60 [13a 10 1b 800] -------
13  [6:0 19:0 30 1] [0 0] 2 4
14  f 5.178361237 _1_ RTR --- 3950 tcp 60 [13a 1 10 800] -------
15  [6:0 19:0 29 19] [0 0] 3 4
16  r 5.183296029 _19_ AGT --- 3950 tcp 60 [13a 13 1 800] -------
17  [6:0 19:0 29 19] [0 0] 4 4
18  s 5.183296029 _19_ AGT --- 5368 ack 40 [0 0 0 0] ------- [19:0
19  6:0 32 0] [0 0] 0 4
20  T 5.183296029 _19_ tora enq 19->6
21  T 5.183296029 _19_ tora sendQRY 6
22  T 5.191820384 _2_ tora sendQRY 6
      ……
    T 5.237823041 _3_ tora sendQRY 6
    s 5.331676081 _19_ RTR --- 5368 ack 60 [0 0 0 0] ------- [19:0
    6:0 32 12] [0 0] 0 4
    f 5.357902981 _12_ RTR --- 5368 ack 60 [13a c 13 800] -------
    [19:0 6:0 31 21] [0 0] 1 4
    f 5.362818058 _21_ RTR --- 5368 ack 60 [13a 15 c 800] -------
    [19:0 6:0 30 27] [0 0] 2 4
    f 5.364796228 _27_ RTR --- 5368 ack 60 [13a 1b 15 800] -------
    [19:0 6:0 29 6] [0 0] 3 4
    r 5.367234077 _6_ AGT --- 5368 ack 60 [13a 6 1b 800] -------
    [19:0 6:0 29 6] [0 0] 4 4
```

Fig. 2. ns2 trace file relevant to the performance of the network layer (TORA) and transport layer (TCP) in the attacker free networks.

These intervals are called retransmission time out (RTO) intended to reduce congestion in the network but it may also cause a sharp drop in the network's efficiency in packet transmission (reminder: TCP accounts any sort of packet loss as for the network congestion).

Because of TORA performance process in which the routes of transmitting TCP packet and its acknowledgment return (ack) might be different, the second state in which one intruder or selfish node can cause the reduction network efficiency through the absence forward running packet happens when the packet is delivered safe and sound but its ack takes a return route bearing an intruder node causing the elimination of ack packet. Since the transmitter does not receive the acknowledgment again, it assumes that the package has not been delivered and starts to retransmit it.

```
1   s 5.000000000 _6_ AGT --- 3950 tcp 40 [0 0 0 0] ------- [6:0
2   19:0 32 0] [0 0] 0 4
3   T 5.000000000 _6_ tora enq 6->19
4   T 5.000000000 _6_ tora sendQRY 19
5   T 5.006458085 _37_ tora sendQRY 19
6     ……..
7   T 5.038696131 _32_ tora sendQRY 19
8   s 5.084920420 _6_ RTR --- 3950 tcp 60 [0 0 0 0] ------- [6:0
9   19:0 32 27] [0 0] 0 4
10  f 5.147486852 _27_ RTR --- 3950 tcp 60 [13a 1b 6 800] -------
11  [6:0 19:0 31 16] [0 0] 1 4
12  D 5.167766553 _16_ RTR ATT 3950 tcp 60 [13a 10 1b 800] ----
13  --- [6:0 19:0 30 1] [0 0] 2 4
14    ……..
15  s 11.000000000 _6_ AGT --- 10915 tcp 40 [0 0 0 0] ------- [6:0
16  19:0 32 0] [0 0] 0 4
17    ……..
      s 23.000000000 _6_ AGT --- 20846 tcp 40 [0 0 0 0] ------- [6:0
      19:0 32 0] [0 0] 0 4
        ……..
      s 47.000000000 _6_ AGT --- 40709 tcp 40 [0 0 0 0] ------- [6:0
      19:0 32 0] [0 0] 0 4
        ……
      s 95.000000000 _6_ AGT --- 80430 tcp 40 [0 0 0 0] ------- [6:0
      19:0 32 0] [0 0] 0 4
```

Fig. 3. ns2 trace file relevant to the performance of the network layer (TORA) and transport layer (TCP) in the network bearing attacker.

4 SECTORA

TORA is a multi-path routing protocol. During routing process, it finds multiple routes from the source to destination. Each node selects a node with

the lowest height among its neighbors as the next step when it decides to forward a packet to the destination and overlooks the other nodes. SecTORA exploits the potential of various routes to drop the impact of the nodes infiltrated into the network and deleted data packets. If there is a mechanism informing SecTORA on the intruder nodes deleting data packets, this protocol can change the forward-running route of the data packets. For example, instead of the node with the shortest height, the second one, in case it exists, could be chosen as the next hop. This causes a switch in the forward-running route of the packet with the hope that the new route would not have any intruder nodes. This method would be effective when 1. There is more than one node as the next step at the time of forward-running the packet to the destination, and 2. The new route bears no intruder node.

Node with the lowest height path and the node with the second lowest height form the main and alternate paths respectively. Unlike most multi-path routing methods in which various paths are used simultaneously to run the packets forward, in SecTORA only one path is used at a time. As mentioned in [12], simultaneous forward-running of the packets through various paths will lead to two major problems:

1. When various paths for transmitted packets (due to unequal length of the paths or other reasons) are used, they arrive at the destination out of order. This causes producing repeated ack packets, retransmitting them, reducing the congestion window and subsequently dropping the TCP efficiency (e.g. congestion window is cut into half after three received repeated acks).

2. Parameter estimation of "average round-trip time (RTT)" is not done with precision since the time is varied in different paths and may be far apart. This parameter's value is used in calculating TCP expectations to receive the acknowledgment for a packet and is considered as a significant parameter in TCP efficiency.

In order to avoid the reduction of TCP efficiency for the reasons discussed, back-up path routing in SecTORA is used in which one path is just used at a time. At the same time, some back-up paths are also maintained so they can be swiftly switched into other paths if necessary (when the main path is under attack). In [11], this method is used so that other paths are employed when the main path is temporarily out of order. Here the following questions arise:

1 when is the time to use back-up paths?

The answer is: when it is figured out that there are intruder or selfish nodes on the main path which eliminate data packets.

2. How can it be identified that there are intruder nodes on a path?

In the proposed method, the network layer (SecTORA) uses the transport layer (TCP) information and guesses the attackers' presence on a path. This is the method: SecTORA in the source node (not the middle ones) reads the TCP header of the packet to be transmitted and records the field of the sequence number, representative of the sequence number of the transmitted packet, in its memory. If there is an intruder node on the path, it eliminated the packets continuously. By the way, it is clear that TORA algorithm takes no efforts to switch the path of transmitting the packets because of its implementation. However, given that SecTORA records the sequence number of the packets TCP has transmitted in one path, one can say that lots of packets are unregularly retransmitted and it may partly go to the malicious or selfish nodes. Packets' retransmission may occur for other reasons though which is dealt with in the questions. In this case, while changing the algorithm of the path selection (in TORA, there is always a neighbor with the lowest height chosen to run the packet forward), we can get assistance from the alternate path: if SecTORA figures out that, given the sequence number of the packets it transmits through a given path, retransmission is beyond the normal, it stops transmitting through the main path and switches quickly to the next path (if there is a second node with lowest height among neighbors) and uses it to transmit the packets through. Moreover, still in the destination, SecTORA records the TCP packets' sequence number which receives in a TCP communication. When SecTORA realized that it received repeated TCP packets beyond the normal given the sequence numbers recorded, it may guess that the round trip of ack packets contain intruder or selfish nodes. In this case,it reads the acknowledgement number filed in the header of the ack packets and switches the transmission path of them (in case of any back-up path). Generally, TORA routing algorithm has been only modified at the source and destination nodes (not at the intermediate ones) and the decision to re-switch the path is taken only at the extreme nodes (source and destination). When the decision about route switch is taken, the threshold number must be considered, whenever the number of retransmission of a packet at the source or the number of repeated delivery of a packet at the destination goes beyond the threshold, changing path must carry out.

3. How many backup paths had better be used? In other words, what is the optimal value for the number of alternate paths?

According to the analysis conducted in [12] and [13], it is appropriate to use a main path along with an alternate one and there is no significant higher efficiency of the network when there are more backup paths around.

4. Switching to the backup path, how much data should be transmitted through the path?

Backup route can be used temporarily and it is likely the main route (the shortest path to destination) is debugged (due to mobility of the nodes, intruder nodes may have been derailed) or the alternate path might face the risk of intruder nodes itself. Therefore, SecTORA uses main and alternate paths intermittently and switches between them. Doing lots of simulations using various parameters, it seems that in case of switching from the main route into the alternate one, transmission of three packets and then re-switching into the main route are more optimized. In [14], there is a multi-path routing algorithm provided with changing AODV protocol in which the number of packets transmitted in each of the backup routes is called "frequency". This research maintains that this parameter is out of order of data packets getting to the destination in order to reduce the possibility (because main and backup paths' length is not equal. Transmitted packets through these paths may reach to the destination one after the other and we try to reduce this possibility using frequency parameter). We also call this parameter frequency and consider its value for each backup route as 3.

5. There might be no reason for retransmitting the packets but the presence of intruder nodes. Doesn't the unintended switch of the packets reduce TCP efficiency?

In addition to attacks or selfish behavior, data packets may be eliminated by any node in the network (even those having a desirable behavior) for some reasons [15]:

* Collision at MAC layer: TORA protocol provides no guarantee for data delivery (like IP protocol).therefore, data packets will not buffer for retransmission. In the collision facing a packet, it is simply considered as deleted. The responsibility for retransmission of the packet goes to higher layers of protocol stack.

* saturation of interface queues: TORA implements network interface queue (IFQ) to buffer the packets ready for transmission and received by the networks' protocol stack. There is a limitation on both the number and time these packets can keep waiting in the queue. As a result, it is quite probable that some of the packets waited for a long time in the queue or didn't find a place due to congestion, are deleted without any notice.

* Due to nodes' mobility and also interfering signals of ad hoc networks, it is likely that some connections between the nodes are temporarily eliminated and some packets are also lost. This state might not be counted as attack or selfish behavior.

It is necessary to note that even when the main path contains no intruder node and some packets are eliminated only for the reasons mentioned above, SecTORA avoids the problematic paths. Also, after switching to the backup path and transmitting several packets through, the main path is used again. As a result, if the main path contains no node from the start, packet transmission continues normally through which.

6. Does the proposed method identify intruder nodes?

No, SecTORA only identifies a kind of route in which unreasonable and over the base removal of the packets happen and avoids it. It cannot identify which node (s) on this route removes the packets unreasonably (route identification vs. node identification).

```
s 255.283679584 _6_ AGT  --- 295715 tcp 1500 [0 0 0 0] ------- [6:0 19:0 32 0] [6901 0] 0 2
s 255.283679584 _6_ RTR  --- 295715 tcp 1520 [0 0 0 0] ------- [6:0 19:0 32 37] [6901 0] 0 2
f 255.299076386 _28_ RTR  --- 295547 tcp 1520 [13a 1c 2e 800] ------- [6:0 19:0 30 22] [6900 0] 2 2
f 255.317246319 _37_ RTR  --- 295715 tcp 1520 [13a 25 6 800] ------- [6:0 19:0 31 30] [6901 0] 1 2
D 255.330923908 _30_ RTR  ATT 295715 tcp 1520 [13a 1e 25 800] ------- [6:0 19:0 30 33] [6901 0] 2 2
f 255.350069442 _22_ RTR  --- 295547 tcp 1520 [13a 16 1c 800] ------- [6:0 19:0 29 19] [6900 0] 3 2
r 255.363686798 _19_ AGT  --- 295547 tcp 1520 [13a 13 16 800] ------- [6:0 19:0 29 19] [6900 0] 4 2
s 255.363686798 _19_ AGT  --- 295956 ack 40 [0 0 0 0] ------- [19:0 6:0 32 0] [6900 0] 0 2
s 255.363686798 _19_ RTR  --- 295956 ack 60 [0 0 0 0] ------- [19:0 6:0 32 41] [6900 0] 0 2
f 255.371548916 _41_ RTR  --- 295956 ack 60 [13a 29 13 800] ------- [19:0 6:0 31 36] [6900 0] 1 2
f 255.373766142 _36_ RTR  --- 295956 ack 60 [13a 24 29 800] ------- [19:0 6:0 30 6] [6900 0] 2 2
r 255.375764060 _6_ AGT  --- 295956 ack 60 [13a 6 24 800] ------- [19:0 6:0 30 6] [6900 0] 3 2
s 255.375764060 _6_ AGT  --- 295979 tcp 1500 [0 0 0 0] ------- [6:0 19:0 32 0] [6902 0] 0 2
s 255.375764060 _6_ RTR  --- 295979 tcp 1520 [0 0 0 0] ------- [6:0 19:0 32 37] [6902 0] 0 2
f 255.389441739 _37_ RTR  --- 295979 tcp 1520 [13a 25 6 800] ------- [6:0 19:0 31 30] [6902 0] 1 2
D 255.404741327 _30_ RTR  ATT 295979 tcp 1520 [13a 1e 25 800] ------- [6:0 19:0 30 33] [6902 0] 2 2
s 261.295764060 _6_ AGT  --- 302359 tcp 1500 [0 0 0 0] ------- [6:0 19:0 32 0] [6901 0] 0 2
T 261.295764060 _6_ tcp Packet 6901 Change route from 37 to 46:3
s 261.295764060 _6_ RTR  --- 302359 tcp 1520 [0 0 0 0] ------- [6:0 19:0 32 46] [6901 0] 0 2
f 261.309637824 _46_ RTR  --- 302359 tcp 1520 [13a 2e 6 800] ------- [6:0 19:0 31 28] [6901 0] 1 2
f 261.323735693 _28_ RTR  --- 302359 tcp 1520 [13a 1c 2e 800] ------- [6:0 19:0 30 22] [6901 0] 2 2
f 261.342971760 _22_ RTR  --- 302359 tcp 1520 [13a 16 1c 800] ------- [6:0 19:0 29 19] [6901 0] 3 2
r 261.358114032 _19_ AGT  --- 302359 tcp 1520 [13a 13 16 800] ------- [6:0
```

```
19:0 29 19] [6901 0] 4 2
s 261.358114032 _19_ AGT  --- 302539 ack 40 [0 0 0 0] ------- [19:0 6:0
32 0] [6901 0] 0 2
s 261.358114032 _19_ RTR  --- 302539 ack 60 [0 0 0 0] ------- [19:0 6:0
32 41] [6901 0] 0 2
f 261.364034128 _41_ RTR  --- 302539 ack 60 [13a 29 13 800] -------
[19:0 6:0 31 36] [6901 0] 1 2
f 261.367662203 _36_ RTR  --- 302539 ack 60 [13a 24 29 800] -------
[19:0 6:0 30 6] [6901 0] 2 2
r 261.370019971 _6_ AGT  --- 302539 ack 60 [13a 6 24 800] -------
[19:0 6:0 30 6] [6901 0] 3 2
s 261.370019971 _6_ AGT  --- 302593 tcp 1500 [0 0 0 0] ------- [6:0
19:0 32 0] [6902 0] 0 2
T 261.370019971 _6_ tcp Packet 6902 Change route from 37 to 46:2
s 261.370019971 _6_ RTR  --- 302593 tcp 1520 [0 0 0 0] ------- [6:0 19:0
32 46] [6902 0] 0 2
s 261.370019971 _6_ AGT  --- 302594 tcp 1500 [0 0 0 0] ------- [6:0
19:0 32 0] [6903 0] 0 2
T 261.370019971 _6_ tcp Packet 6903 Change route from 37 to 46:1
s 261.370019971 _6_ RTR  --- 302594 tcp 1520 [0 0 0 0] ------- [6:0 19:0
32 46] [6903 0] 0 2
f 261.385170689 _46_ RTR  --- 302593 tcp 1520 [13a 2e 6 800] -------
[6:0 19:0 31 28] [6902 0] 1 2
f 261.398869044 _46_ RTR  --- 302594 tcp 1520 [13a 2e 6 800] -------
[6:0 19:0 31 28] [6903 0] 1 2
f 261.412946912 _28_ RTR  --- 302593 tcp 1520 [13a 1c 2e 800] -------
[6:0 19:0 30 22] [6902 0] 2 2
f 261.436307685 _22_ RTR  --- 302593 tcp 1520 [13a 16 1c 800] -------
[6:0 19:0 29 19] [6902 0] 3 2
r 261.450244990 _19_ AGT  --- 302593 tcp 1520 [13a 13 16 800] -------
[6:0 19:0 29 19] [6902 0] 4 2
s 261.450244990 _19_ AGT  --- 302809 ack 40 [0 0 0 0] ------- [19:0 6:0
32 0] [6902 0] 0 2
s 261.450244990 _19_ RTR  --- 302809 ack 60 [0 0 0 0] ------- [19:0 6:0
32 41] [6902 0] 0 2
f 261.452402502 _41_ RTR  --- 302809 ack 60 [13a 29 13 800] -------
[19:0 6:0 31 36] [6902 0] 1 2
f 261.454379747 _36_ RTR  --- 302809 ack 60 [13a 24 29 800] -------
[19:0 6:0 30 6] [6902 0] 2 2
r 261.460187891 _6_ AGT  --- 302809 ack 60 [13a 6 24 800] -------
[19:0 6:0 30 6] [6902 0] 3 2
```

Fig. 4. Changing path by SecTORA at the source node when retransmitting the deleted packets

7. what will happen if both main and backup routes contain intruder nodes?

In this case, SecTORA has no advantages. We are here dealing with the law of possibilities. If we assume that the probability of each path containing an intruder node (s) is 0.1, then by using two paths this probability is between 0.01 (0.1 × 0.1) and 0.1 (note that the main and backup paths are not completely separated). That is, the more we use backup routes, the less is the collision with the intruder nodes.

8. Why the consideration of path switch happens only at the extreme nodes (source and destination)?

This is one of the benefits of proposed method since many proposed safety algorithms on ad hoc networks require the operation of all or some nodes. Some of the nodes might be intruder, not cooperative and disruptive. SecTORA just trusts the source and destination nodes and makes sure they are not intruder nodes. So the intermediate nodes are ineffective in the decision to switch the paths.

Figure 4 demonstrates SecTORA routing process using the simulation done by ns2 simulation environment. In line (1), a TCP packet with sequence number 6901 is produced by node 6 whose ultimate destination is node 19. Node 30 is an intruder one and deletes data packets instead of running-forward them.Packet 6901 on line (4) has been run forward with node 37 but it is removed by node 30 on line (5). As a result, the destination node, 19, does not receive the packet and there would be no ack for it. In line (16), packet 6902 suffers the same fate and is removed by the node 30. After the time expiry and not receiving the relevant ack of the packet 6901, it is retransmitted on line (17) by node 6. Regarding that SecTORA records sequence number of the packets transmitted to the destination, it realizes that the packet is retransmitted. Thus, it decides in line (18) that it switches the path for its retransmission and transmits it to node 46 instead of node 37 (switching to the backup path). The number at the end of line (18) shows the frequency parameter, meaning that the next three packets will be transmitted from the backup path.

The packets 6902 and 6903 on lines (30) and (33) are transmitted from backup route due to the same frequency parameter. Because the backup path contains no intruder nodes, these three packets get the destination node intact (packet 6901 on line 23, packet 6902 on line 39).

Parameter value decreases by each packet transmission. Once it reaches zero, packet transmission through backup route is stopped and the main route is reused to transmit the packets.

```
1   s 76.395774508 _6_ AGT  --- 92645 tcp 1500 [0 0 0 0] ------- [6:0 19:0
    32 0] [3379 0] 0 2
2
    s 76.395774508 _6_ RTR  --- 92645 tcp 1520 [0 0 0 0] ------- [6:0 19:0
3   32 33] [3379 0] 0 2
4   f 76.409707840 _33_ RTR  --- 92645 tcp 1520 [13a 21 6 800] -------
    [6:0 19:0 31 10] [3379 0] 1 2
5
    f 76.423426133 _10_ RTR  --- 92645 tcp 1520 [13a a 21 800] -------
6   [6:0 19:0 30 19] [3379 0] 2 2
7   r 76.437082336 _19_ AGT  --- 92645 tcp 1520 [13a 13 a 800] -------
    [6:0 19:0 30 19] [3379 0] 3 2
8
    s 76.437082336 _19_ AGT  --- 92750 ack 40 [0 0 0 0] ------- [19:0 6:0
9   32 0] [3379 0] 0 2
    s 76.437082336 _19_ RTR  --- 92750 ack 60 [0 0 0 0] ------- [19:0 6:0
10  32 35] [3379 0] 0 2
11  f 76.439198960 _35_ RTR  --- 92750 ack 60 [13a 23 13 800] -------
    [19:0 6:0 31 30] [3379 0] 1 2
12
    D 76.443629595 _30_ RTR  ATT 92750 ack 60 [13a 1e 23 800] -------
13  [19:0 6:0 30 42] [3379 0] 2 2
14  s 78.855774508 _6_ AGT  --- 95445 tcp 1500 [0 0 0 0] ------- [6:0 19:0
    32 0] [3379 0] 0 2
15
    s 78.855774508 _6_ RTR  --- 95445 tcp 1520 [0 0 0 0] ------- [6:0 19:0
    32 33] [3379 0] 0 2
```

```
16  f 78.869427971 _33_ RTR  --- 95445 tcp 1520 [13a 21 6 800] -------
    [6:0 19:0 31 19] [3379 0] 1 2
17
    r 78.885844580 _19_ AGT  --- 95445 tcp 1520 [13a 13 21 800] -------
18  [6:0 19:0 31 19] [3379 0] 2 2

19  s 78.885844580 _19_ AGT  --- 95446 ack 40 [0 0 0 0] ------- [19:0 6:0
    32 0] [3379 0] 0 2

    T 78.885844580 _19_ ack Packet 3379 Change route from 35 to 29:3

    f 78.890956267 _29_ RTR  --- 95446 ack 60 [13a 1d 13 800] -------
    [19:0 6:0 31 27] [3379 0] 1 2

    f 78.896129499 _27_ RTR  --- 95446 ack 60 [13a 1b 1d 800] -------
    [19:0 6:0 30 45] [3379 0] 2 2

    r 78.899909499 _6_ AGT  --- 95446 ack 60 [13a 6 3 800] ------- [19:0
    6:0 31 6] [3379 0] 3 2

    s 113.330905457 _6_ AGT  --- 95447 tcp 1500 [0 0 0 0] ------- [6:0 19:0
    32 0] [3380 0] 0 2
```

Fig. 5. ack packet redirection by SecTORA at the destination node when duplicate packets of TCP are received.

As mentioned earlier, since the transmission route of TCP packets and their ack transmission route might be different, there is this possibility that the packet gets to the destination but its due ack might be removed by intruder or selfish nodes on the path and causes packet retransmission. The scenarios related to the performance process of SecTORA protocol at the destination node and re-switch of the ack packets have been represented in Figure 5 (ack packet redirection on line 15).

5 SIMULATION AND COMPARISON

In order to analyze the results and effects, intruder nodes apply to networks' efficiency and performance and also compare TORA and SecTORA protocols' efficiency, simulation is used. In doing so, ns2 network simulator is used.

5.1 Simulation model and Parameters

In the current study, the presented simulations are under version 2.28 of the simulator and have been carried out by Enterprise Linux Readhat 9(u7) operating system.

We use a scenario with the Table 1 parameters:

Table 1: Simulation parameters

Number of Nodes	Size of Network	Connection Type	TCP Connection Number
50	670×670square meter	TCP	1
Number of Intruder nodes	MAC	Nodes' move model	Routing algorithm
5	IEEE 802.11	Random	SecTORA and TORA
Size of Packets	Simulation Time		
1460 byte	1200 seconds		

5.2 Simulation Results

Before analyzing simulation results, the following definition is presented:

TCP goodput: is the number of consecutive bits that a TCP receiver receives per second. Break down or duplicate packets are not considered in this enumeration [14].

According to the above definition, the criterion under question in the analysis of the simulation results is the goodput value. Figure 6 shows a comparison between TORA and SecTORA protocols in a network free from attackers. This can be a response to question 5. The question was: if , indeed, there is no attacker in the network and continuous retransmission of the TCP packets happens for reasons other than the presence of attackers, doesn't unreasonable packets' redirection by SecTORA cause its efficiency loss comparing to TORA?

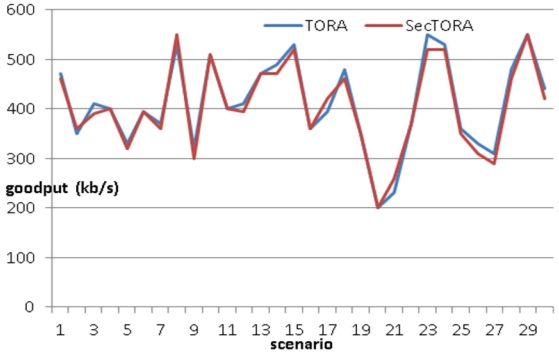

Fig. 6.: TORA and SecTORA performance process in a non-attacker network with random scenarios

Regarding simulation results shown in Figure 6, in a network free from attackers, SecTORA does well in some scenarios and some other not. And, the question of which one acts better in which rate is completely random. On average, in a non-attacker network, SecTORA decreases goodput value by 1.47% so its performance is weaker than TORA's. The point is since the main path is the closest one to the destination, it basically performs better on packet transmission comparing to alternate path which is the second closest to the destination. Moreover, unreasonable switch of the packets' routes into alternate ones would result in the drop of goodput parameter.

However, the reason why SecTORA performs better than TORA in some scenarios in a non-attacker network is, as there is packet removal on major route, using alternate route would cause to decrease the congestion on the main route and the number of packets in the buffers and also to resolve the cuts or temporary loops. The other major reason is that SecTORA finds it unreasonable to switch the route if the packets are transmitted intact and uses the main route most of the time.

Figure 7 represents a comparison between SecTORA and TORA protocols within a network of 50 nodes within which there are 5 intruder nodes which in turn remove the data packets. Goodput improvement value is palpable in SecTORA protocol and noticeable in some scenarios. The average improvement is about 17.4%.

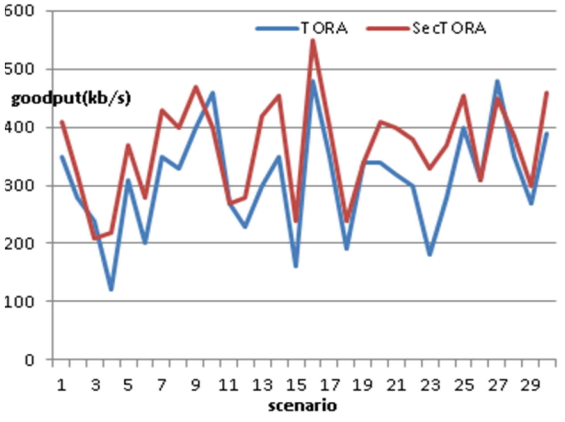

Fig. 7. TORA and SecTORA performance process in a network having attackers with random scenarios.

As it is observed in Figure 7, only in a few numbers of scenarios, goodput value shows "decrease in SecTORA protocol comparing with main TORA protocol and as it was mentioned earlier the reason is that both the main and alternate routes may have intruder nodes or there may not be any alternate routes in case of urgency.

According to the simulations, if SecTORA protocol is employed for TORA protocol, there is a price to pay for 17.4% improvement of the goodput value in the intruder network and that is the 1.47% drop of goodput value in a non-attacker network.

The conclusion one can draw from the simulation is that it is better to use TORA protocol if all nodes are trusted and no intruder nodes exist. However, if the presence of intruder nodes is probable, SecTORA is preferred.

6 CONCLUSION

This study dealt with a well-known protocol of ad hoc network, TORA and tried to empower this protocol to detect the attackers' infiltration and reduce their malicious acts with its algorithm modification. SecTORA is a proposed protocol for ad hoc networks into which intruder nodes have infiltrated and started to remove data packets.

It was suggested that if we realize that there are intruder nodes on the main routes and data packets' removal happen intermittently, it is better to use other existing routes other than the main one to transmit data. The method used to identify routes with attackers was the header of TCP packets. Simulations proved higher capability of SecTORA to TORA's main algorithm within the networks with attackers.

SecTORA uses TCP, its higher layer, to detect the routes with attackers. This algorithm can be empowered using feedbacks from lower layers like IMEP and MAC as these layers can provide more information on data packets' removal.

7 REFERENCES

[1] Dr.S.S.Dhenakaran, A.Parvathavarthini, "An Overview of Routing Protocols in Mobile Ad-Hoc Network", International Journal of Advanced Research in Computer Science an Software Engineering", Volume 3, Issue 2, February 2013

[2] Pawan Bhadana, Ritu Khurana, Chanchal, Manisha, "secure Adhoc Network", International Journal of Computational Engineerin Research, Vol 03, Issue 6, June 2013

[3] Argyroudis Patroklos, Mahony Donal, "Secure Routing for Mobile Ad hoc Networks",Department of Computer Science University of Dublin, Trinity College, 2004

[4] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers", ACM SIGCOMM Conference on

Communications Architectures, Protocols and Applications, Vol. 24, pp. 234-244 , 1994

[5] M. Abolhasan, T.A. Wysocki, and E. Dutkiewic, "A Review of Routing Protocols for Mobile Ad hoc Networks", Ad hoc Networks, Vol. 2, pp. 1-22. , 2004

[6] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", Ad Hoc Networking, Addison-Wesley, pp. 139-172 , 2001

[7] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing (AODV) ", IETF RFC 3561, 2003

[8] Park V., Corson S., "Ahighly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Proceeding of IEEE INFOCOM '97, pp.1405-1413. IEEE Computer Society Press, Los Alamitos , 1997

[9] Chee V.L., Yau W.C., "Security Analysis of TORA Routing Protocol", ICCSA 2007, LNCS 4706, Part II,pp. 975-986, Springer-Verlag Berlin Heidelberg, 2007

[10] R. Ramanujan, A. Ahamad, and K. Thurber, "Techniques for Intrusion Resistant Ad hoc Routing Algorithms (TIARA)," Proc. Military Communications Conf. (MILCOM 2000), Los Angeles, CA, pp. 660-664, 2000

[11] Park V., Corson S., "Temporally-Orderd Routing Algorithm(TORA) Version 1", Functional Specification, Internet Draft, draft-ietf-manet-tora-spec-04.txt, 2001

[12] H. Lim, K. Xu, M. Gerla, "TCP Performance over Multipath Routing In Mobile Ad-Hoc Networks", In Proc. IEEE Int. Conf. Communications , Vol.2, Anchorage, AK, USA, pp.1064—1068, 2003

[13] Alvin Valera, Winston K.G. Seah, "Cooperative Packet Caching and Shortest Multipath Routing in Mobile Ad hoc Networks", IEEE INFOCOM, 2003

[14] Z. Ye, S. V. Krishnamurthy, S. K., Tripathi, "Effects of multipath routing on TCP Performance in Ad Hoc Networks", in. Proc. of IEEE GLOBECOM , 2004

[15] Pirzada A. A., McDonald C., Datta A., "Reliable Link Reversal Routing for Mobile Ad-hoc Wireless Networks", pp. 234-239, IEEE 2005.