Available online at www.scholarsresearchlibrary.com

Scholars Research Library

European Journal of Applied Engineering and Scientific Research, 2012, 1 (4):207-215
(http://scholarsresearchlibrary.com/archive.html)

ISSN: 2278 – 0041

Investigation of attack types in Ad Hoc networks and simulation of wormhole avoidance routing protocol

[1]Kimia Moradi , [2]Majid Rahiminasab

[1]Computer Department, Azad University, Boroujerd, Iran
[2]Technical and engineering department, Lorestan university, khorramabad, Iran

---

ABSTRACT

Progress in networks and technologies of communication have led to emergence of wireless devices in most of our works. People use wireless networks in a wide range of their activities. Among these networks are Ad-Hoc networks that are very applicable. In these networks nodes do the routing operation by their own and there is possibility that the network unexpectedl, experience some changes in topology. One of the most important problems in these networks is security establishment. Various attacks are imposed on these networks and every attack, in its way, threatens the network's security. This article pays attention to divisions of attack types and that How much change every attack can make on the network or How can they threaten it? Following using NS2 simulator, we simulate warp algorithm to confront wormhole attack.

Key words: Ad-Hoc network, Routing, Security, Attack, Simulator

---

INTRODUCTION

Progress in networks and technologies of communication have led to emergence of portable wireless devices in most of our works. Most of the people use laptops, pagers and mediums which enjoy the benefit of mobile technologies. Among these networks we can name Ad-Hoc networks. Ad-Hoc networks are joined together via wireless hosts that use wireless links. These networks are not obligated to use constant and pre-structured substructures such as central station, router and switch, but there are simply some wireless nodes that using the connection with neighbor nodes are linked to non-neighbor nodes. In these networks routing operation is done by the node themselves and , indeed, every node works as a router forwarding data packages for other existing nodes in the network[1]. It is possible that the network quickly and accidentally experiences changes in topology. These networks, because of fast and simple implementation and also independence from pre-structured frameworks, have much usages in connecting laptops together, military settings and remote control of battles, search and rescue operation for regenerating and achieving information in unexpected accidents. Ad-Hoc networks alike other networks, whether wireless or wired, need security to perform correct operation including routing, forwarding data packages, keeping and updating information. Basically security is the essential term for correct network performance and without it. There is no guarantee for doing correct operations, consequently, attackers can easily pass through and unsettle it's unity. [2] Security issues, in these networks, are specifically under evaluation because here, in addition to all existent problems in wired networks or a wireless network having a wired substructure, there is other defects like overhearing or changes in information being transferred and various attacks are imposed on these networks and each one threatens the security in some way. In this article we focus on categorizing attack types and that how much change every attack can make on the network or how can they threaten it? One of attacks that is dictated to wireless networks, specially to Ad-Hoc ones is wormhole attack and itself has different types which are studied here.

1. Attack kinds

Attacks against Ad-Hoc networks can be categorized from some aspects: external attacks and internal attacks. External attacks are made by one or more node outside the network and the most security actions are exercised against these attacks. Internal attacks are made by valid nodes inside the network and it is difficult to prevent these attacks. Attacks, in other war, are divided into active and inactive categories. In active attacks, the attacker simply does overhearing in data that are transferring. But in active attacks, in addition to overhearing data, the attack can change them to gain their benefits to it‘s own interest. Another category is from the viewpoints of layers that are under attacks, that is the attacks can happen in physical, application, data link or network layers. A different kind of attack such as non-attending in routing operation or disconnection are also found that can lead to prevention of service and the only way for preventing them is finding attacking node. The next attack is the integrity attack which in it, the attack can introduce itself in behalf of a correct node [3]. One more kind of attacks is denial of service attack. In this kind, the attacker injects a large number of useless packages which consume a major part of networks resources [3]. Two more kind of attacks are routing disruption attack and routing consumption attack. In routing disruption attack, the attacker tries to send his own packages, as a valid one, on the network until they are used in inefficient ways. The attacker, in routing consumption attack, tries to utilize the bandwidth and /or memory and accounting ability of the node with sending an invalid package. One more kind of attack is rushing attack. In this one the attacking node, in the path discovery operation, sends his request very quicker than the valid node, hence it‘s package will be most likely accepted than the valid one. The attacker can more probably construct a path which itself is found in it. Some attacks such as wormhole attacks are peculiar to Ad-Hoc networks. Wormhole was taken from a physic hypothesis stated by John Whiler in 1957. This attack is considered as a cleverly one which in this one, two nodes construct a private virtual tunnel connecting the current streams of messages with short links and consequently adjoin two non-neighbor nodes. Wormhole attack can cause a serious threat against Ad-Hoc networks routing. In fact we can say: this attack has a spatial-chronological topologic property which is a shortcut between time and place. As a consequent of this attack, this private network can cover a long distance of the path without rising of hop count and the package can reach the destination with only two hops. Henceforth this path will be certainly chosen as the shortest path. Wormhole is a virtual shortcut path which connects distant nodes and generates two vague attacking nodes in which two separated nodes are connected together through nodes in a way that seems they are neighbors but they are actually far away of each other.

2. Routing stages in Ad-Hoc networks

3.1. Rout request stage: Source node spreads a controlling rout request package, briefly called RREQ, in the network and every node hearing it for the first time starts responding it.

3.2. Rout reply stage: The destination, as soon as receiving a RREQ message, sends the rout reply package for the source node in the opposite way. Rout reply package is briefly called RREP.

3. Wormhole attack

Wormhole attack extremely influences the routing operation in the network. For example, as it is shown in the following figure , if attacking node C transfers rout request package S, using it‘s high speed link with k, to one of the nodes J,D,H or A then the target nodes suppose the node K is their neighbor or is only one hop away of them. Therefore, it transfers the package through generated tunnel between node C and node K.

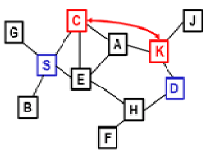

Fig 1: wormhole attack [3]

3.1. Classification of wormhole attack

4.1.1. Wormhole attack using Encapsulation

According to figure 2, X and Y are two attacking nodes. When node A sends rout request package, it reaches to node X. Then node X generates a virtual capsule between itself and next attacking node and transfers the package inside this capsule. After receiving the package in Y, the node Y directs the package towards destination B. The point is, because of Encapsulating the package, the length of hop while passing nodes U, V, W, Z doesn't increase. The rout request simultaneously reaches the destination D through the path C, D, E. The node B has two paths, one is through C, D, E with the length 4 and the other through attacking node X and Y with length 3 [4]. The node B chooses the short path but, in reality the length of chosen path is 7, so we can say: every routing protocol which uses the criteria of shortest path as the best path is vulnerable to wormhole attack.

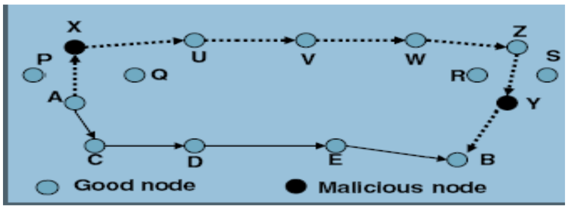

Fig 2: Wormhole via Encapsulation

43.1.2. Wormhole via a channel outside the band

According to figure 3, this attack occurs between attacking nodes through a high bandwidth channel outside the band. The happening of this attack is less possible than the former one, because it needs a special hardware capability. There is also two path here: a path through C (A-C-D-E-F-B) with the length 5 and a path through attacking nodes with length 3. In this one, the destination node chooses a short length path too, and the attack performs successfully. [4]

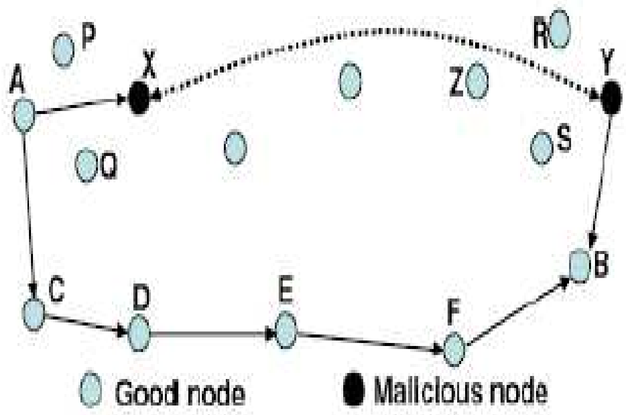

Fig3: Wormhole via outside the band channel [4]

4.1.3. Wormhole via relaying the package

In this kind of attack, an attacking node relays packages between two remote nodes, to convince them that they are neighbors. Every attacking node do this action. The minimum number of attacking node in two previous procedure was two nodes, but in this one this number is one. [4]

4.1.4. Categorizing from the view point of Mahajan

In 2009, Mahajan devides wormhole attacks into some categories:

4.1.4.1. Wormhole attack inside the band

This type requires an overlap through out the existing wireless medium.

4.1.4.2. Wormhole attack outside the band
As it was mentioned in the earlier section, this attack requires a hardware channel for connecting the nodes that through creation of a virtual tunnel are going to generate a wormhole attack and causing it to seen shorter. [5]

The first kind of attacks are devided into two groups:
A) Self-sufficient wormhole attack: in this one the attack is limited to attacking nodes.
B) Extended wormhole attack: the wormhole attack is extended beyond attacking nodes.
The second kind of attacks are also devided into two groups:
A) Hidden attack: in this one, the network is not aware of the presence of attacking nodes, intending to generate a wormhole attack.
B) Obvious attack: the network is aware of the presence of attacking nodes but can not recognize them in other existing nodes. [5]

4.1.5. Categorizing wormhole attack based on the view point of Wang :

4.1.5.1. Closed wormhole: in this attack, according to figure 4, attacking nodes are external elements which target the process of discovering neighbors.
4.1.5.2. Open wormhole: in this kind, both attacking nodes M1 and M2 are internal nodes which participate in routing protocol.
4.1.5.3. Half-open wormhole: only one attacking node is in endangered node which participates in routing attack and the other node is simply an external element.

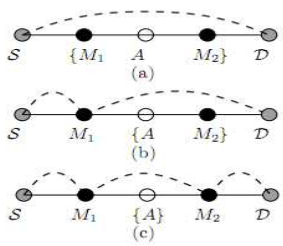

Fig 4: a) closed b) half open c) open

Half-open wormhole attack is devided into two groups:
Weak open wormhole and strong open wormhole [7]. In the weak one, the virtual tunnel connects an attacking node at D hop distance of source node, to another attacking node, at (D+1) hop distance of source node. In strong open wormhole attack, if one of the attacking nodes is at D hop distance of source node then the next attacking node is at least at D+2 hop distance of source node. When comparing these two kinds of open wormhole attacks we can say: if weak open wormhole attack can not necessarily give a shorter path to destination then the strong open wormholes certainly do this job.

Various algorithms are introduced for defending wormhole attacks. One of them is Warp algorithm which we simulate its performance against wormhole attack.

4. Warp algorithm
One of the protocols for avoiding wormhole attack while routing process is warp protocol which was first stated by Ming-yang su at 2009. This routing protocol is on the basis of AODV routing algorithm and can keep wormhole nodes away from interfering routing course. This protocol investigates multiple separate paths which are found between source and destination and at last it chooses only one path to transmit data packages. The work of warp is fundamentally based on the principle that neighbor nodes should be aware of wormhole node‘s high ability for detecting the path and wormhole nodes are occasionally reserved by their neighbor nodes. In comparison of warp with AODV algorithm we can count this difference: rout request message, in warp, has an extra field named first hop which registers the code of first message receiving node. Furthermore, the warp protocol has an extra message which is called the rout request decision, indicated by RREQ-DEC, and has some field similar to RREP. After receiving the RREP message, the sender of routing package should send a RREP-DEC along the path and state that the middle node is located in the path. Another difference of warp and AODV algorithm is the format of routing

table. The entry of routing table in warp has 3 extra fields: 1- first hop field : for illustrating the RREQ needs. 2- RREP counter field: for counting the number of received RREPs. 3- RREP-DEC counter field: for counting the number of received decision making messages. The warp protocol uses the anomaly value for recognizing wormhole attacking nodes. This number illustrates the possibility for location of one node inside the nodes along multiple separate paths. The high anomaly number means that the node is most likely a wormhole. The formula for calculation of anomaly number is as follows:

$$\text{Anomaly} = \frac{\text{RREP DEC COUNT}}{\text{RREP COUNT} + 1}$$

5. Simulation of warp against wormhole attack

Here, we have choose 40 nodes with different coordinates. The routing process was performed based on warp protocol and then the wormhole nodes were omitted from routing operation.

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>
#    Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel
type
set val(prop)   Propagation/TwoRayGround   ;# radio-
propagation model
set val(netif)  Phy/WirelessPhy            ;# network
interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface
queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna
model
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel
#    Mobile node parameter setup
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
                -phyType       $val(netif) \
                -channel       $chan \
                -topoInstance  $topo \
                -agentTrace    ON \
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     44                         ;# number of
mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      1501                       ;# X dimension of
topography
set val(y)      100                        ;# Y dimension of
topography
set val(stop)   10.0                       ;# time of simulation
end
#       Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]
#Setup topography object

                -routerTrace   ON \
                -macTrace      ON \
                -movementTrace ON

#       Nodes Definition
#===================================
#Create 44 nodes
set n0 [$ns node]
$n0 set X_ 410
$n0 set Y_ 293
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 1066
$n1 set Y_ 280
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 1054
$n2 set Y_ 351
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
```

```
$n3 set X_ 1139
$n3 set Y_ 365
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 1158
$n4 set Y_ 300
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 1167
$n5 set Y_ 210
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 1106
$n6 set Y_ 172
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 1055
$n7 set Y_ 152
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
$n8 set X_ 1010
$n8 set Y_ 202
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 1003
$n9 set Y_ 359
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20
set n10 [$ns node]
$n10 set X_ 1075
$n10 set Y_ 432
$n10 set Z_ 0.0
$ns initial_node_pos $n10 20
set n11 [$ns node]
$n11 set X_ 1146
$n11 set Y_ 436
$n11 set Z_ 0.0
$ns initial_node_pos $n11 20
set n12 [$ns node]
$n12 set X_ 1237
$n12 set Y_ 324
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20
set n13 [$ns node]
$n13 set X_ 1261
$n13 set Y_ 236
$n13 set Z_ 0.0
$ns initial_node_pos $n13 20
set n14 [$ns node]
$n14 set X_ 1203
$n14 set Y_ 83
$n14 set Z_ 0.0
$ns initial_node_pos $n14 20
set n15 [$ns node]
$n15 set X_ 1094
$n15 set Y_ 43
$n15 set Z_ 0.0
$ns initial_node_pos $n15 20
set n16 [$ns node]
$n16 set X_ 414
$n16 set Y_ 353
$n16 set Z_ 0.0
$ns initial_node_pos $n16 20
set n17 [$ns node]
$n17 set X_ 338
$n17 set Y_ 367
$n17 set Z_ 0.0
$ns initial_node_pos $n17 20
set n18 [$ns node]
$n18 set X_ 334
$n18 set Y_ 293
$n18 set Z_ 0.0
$ns initial_node_pos $n18 20
set n19 [$ns node]
$n19 set X_ 326
$n19 set Y_ 194
$n19 set Z_ 0.0
$ns initial_node_pos $n19 20
set n20 [$ns node]
$n20 set X_ 406
$n20 set Y_ 167
$n20 set Z_ 0.0
$ns initial_node_pos $n20 20
set n21 [$ns node]
$n21 set X_ 493
$n21 set Y_ 179
$n21 set Z_ 0.0
$ns initial_node_pos $n21 20
set n22 [$ns node]
$n22 set X_ 446
$n22 set Y_ 418
$n22 set Z_ 0.0
$ns initial_node_pos $n22 20
set n23 [$ns node]
$n23 set X_ 405
$n23 set Y_ 458
$n23 set Z_ 0.0
$ns initial_node_pos $n23 20
set n24 [$ns node]
$n24 set X_ 307
$n24 set Y_ 456
$n24 set Z_ 0.0
$ns initial_node_pos $n24 20
set n25 [$ns node]
$n25 set X_ 250
```

```
$n25 set Y_ 419
$n25 set Z_ 0.0
$ns initial_node_pos $n25 20
set n26 [$ns node]
$n26 set X_ 223
$n26 set Y_ 312
$n26 set Z_ 0.0
$ns initial_node_pos $n26 20
set n27 [$ns node]
$n27 set X_ 187
$n27 set Y_ 230
$n27 set Z_ 0.0
$ns initial_node_pos $n27 20
set n28 [$ns node]
$n28 set X_ 173
$n28 set Y_ 132
$n28 set Z_ 0.0
$ns initial_node_pos $n28 20
set n29 [$ns node]
$n29 set X_ 269
$n29 set Y_ 78
$ns initial_node_pos $n33 20
set n34 [$ns node]
$n34 set X_ 1030
$n34 set Y_ 482
$n34 set Z_ 0.0
$ns initial_node_pos $n34 20
set n35 [$ns node]
$n35 set X_ 976
$n35 set Y_ 504
$n35 set Z_ 0.0
$ns initial_node_pos $n35 20
set n36 [$ns node]
$n36 set X_ 866
$n36 set Y_ 516
$n36 set Z_ 0.0
$ns initial_node_pos $n36 20
set n37 [$ns node]
$n37 set X_ 755
$n37 set Y_ 515
$n37 set Z_ 0.0
$ns initial_node_pos $n37 20
set n38 [$ns node]
$n38 set X_ 669
$n38 set Y_ 522
$n38 set Z_ 0.0
$ns initial_node_pos $n38 20
set n39 [$ns node]
$n39 set X_ 557
$n39 set Y_ 516
$n39 set Z_ 0.0
$ns initial_node_pos $n39 20
set n40 [$ns node]
$n40 set X_ 465
$n40 set Y_ 477
$n40 set Z_ 0.0
$n29 set Z_ 0.0
$ns initial_node_pos $n29 20
set n30 [$ns node]
$n30 set X_ 601
$n30 set Y_ 187
$n30 set Z_ 0.0
$ns initial_node_pos $n30 20
set n31 [$ns node]
$n31 set X_ 707
$n31 set Y_ 179
$n31 set Z_ 0.0
$ns initial_node_pos $n31 20
set n32 [$ns node]
$n32 set X_ 779
$n32 set Y_ 149
$n32 set Z_ 0.0
$ns initial_node_pos $n32 20
set n33 [$ns node]
$n33 set X_ 908
$n33 set Y_ 177
$n33 set Z_ 0.0
$ns initial_node_pos $n40 20
set n41 [$ns node]
$n41 set X_ 915
$n41 set Y_ 294
$n41 set Z_ 0.0
$ns initial_node_pos $n41 20
set n42 [$ns node]
$n42 set X_ 697
$n42 set Y_ 271
$n42 set Z_ 0.0
$ns initial_node_pos $n42 20
set n43 [$ns node]
$n43 set X_ 518
$n43 set Y_ 279
$n43 set Z_ 0.0
$ns initial_node_pos $n43 20

#===================================
#        Agents Definition
#===================================
set agent(0) [new Agent/TCP]
set app(0) [new Application/FTP]
set sink(0) [new Agent/TCPSink]
$app(0) attach-agent $agent(0)
$ns attach-agent $n26 $agent(0)
$ns attach-agent $n13 $sink(0)
$ns connect $agent(0) $sink(0)
#        Applications Definition
#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
```

```
   close $tracefile
   close $namfile
   exec nam out.nam &
   exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
   $ns at $val(stop) "\$n$i reset"
}
$ns at 1.0 "$app(0) start"
$ns at 30.0 "$app(0) stop"
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

According to figure 5 node 9, 16 are wormhole nodes.

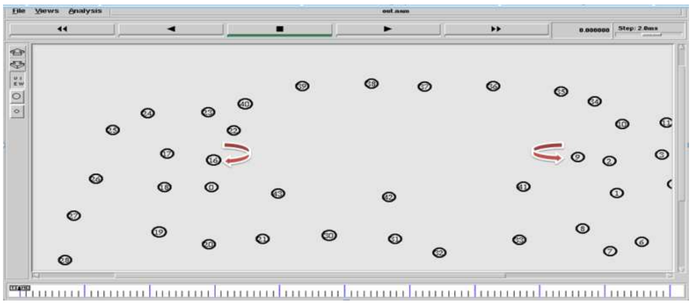

Fig 5: worm hole nodes (9 , 16)

If we consider the node 26 as the source and the node 13 as the destination, then we will see that package is routed between source and destination via various middle nodes and the destination node, after receiving the package, sends RREP message through reversed path to the destination side. The routing process from source to destination is done through different paths and nodes, but the important point is no package is sent to attacking nodes. According to figure 6, the package passes through the node 41 and reaches the node 2. That is, it doesn't reach the node 9.

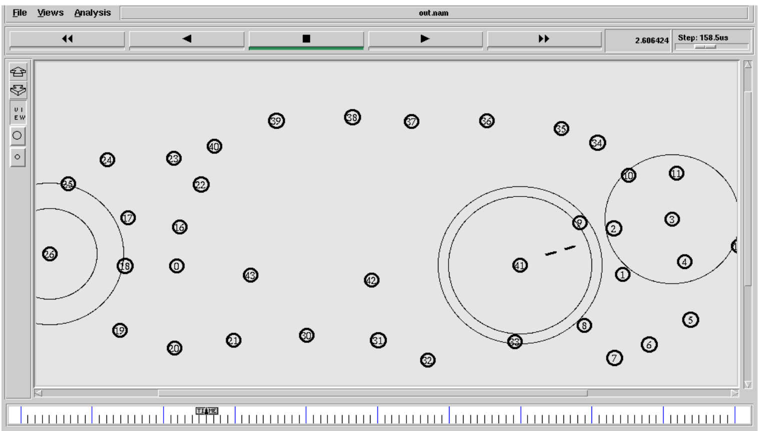

Fig 6: Nonattendance of attacking node 9 in routing

According to figure 7, the package passes through the node 17 and reaches the node 0 and the node 16 is neglected.

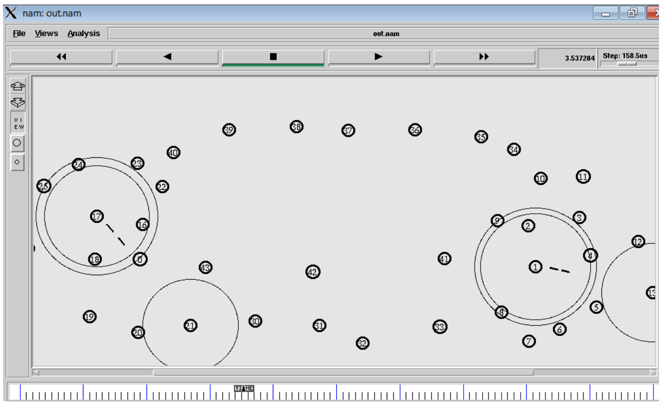

Fig 7: Nonattendance of attacking node 16 in routing

6. Benefits of Warp protocol

Among the benefits of this protocol we can point to the items below: this protocol doesn't need an extra hardware and also it doesn't require simulation of receiver and sender and is always successful in recognizing wormhole nodes.

CONCLUSION

According to results from simulation by NS2 simulator we can say: the warp protocol without the need to only extra hardware support, significantly reduces the missing percentage of the packages and also their deviation from the main path.

REFERENCES

[1] Prayag Narula, Sanjay Kumar Dhurandher, Sudip Misra, Isaac Woungang. *computer communication* 31,**2008** ,pp 760-769

[2] A.A.A. Radwan, T.M. Mahmoud , E.H. Houssein. *12th Egyption Informatics Journal*, **2011**, pp 95-106

[3] Hoang Lan Nguyen,uyon Trang, *Ad Hoc Networks* volume 6, issue 1, January **2008**

[4] Mahdi Nouri, Somayeh Abazari Aghdam, Sajjad Abazari Aghdam. "Collaborative Technique s for Detecting Wormhole Attack in Manet

[5] Mahajan, V.Natu, M.sethi, A, "Analysis of wormhole intrusion attacks in MANETs", in IEEE Military Communications conference,(MILCOM), pp 1-7, **2008**

[6] Ming-Yang Su, *Computer &Security* 29, **2010**, 208-224