© 2015 IJSRSET | Volume 1 | Issue 6 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Classification and Intimation of Black Hole Attack with AODV Routing Protocol in MANET Using NS-3

Deependra Patel, Sapna Choudhary
Shri Ram Group of Institutions, Madhotal, Jabalpur, Madhya Pradesh, India

ABSTRACT

This paper breaks down the blackhole assault which is one of the conceivable assaults in specially appointed systems. In a blackhole assault, a vindictive node imitates destination nodes by sending a ridiculed course answer parcel to source nodes that starts a course disclosure. By doing this, the malevolent nodes can deny the activity from the source nodes. With a specific end goal to keep this sort of assault, it is significant to identify the variation from the norm happens amid the assault. In traditional plans, peculiarity discovery is accomplished by characterizing the typical state from static preparing information. Be that as it may, in versatile specially appointed systems where the system topology progressively changes, such static preparing strategy couldn't be utilized effectively.

Keywords: AODV, anomaly detection, blackhole attack, MANET.

I. INTRODUCTION

Mobile ad hoc network (MANET) is a collection of mobile hosts without the required intervention of any existing infrastructure or centralized access point such as a base station. The applications of MANET range from a one-off meeting network, emergency operations such as disaster recovery to military applications due to their easy deployment. However, due to their inherent characteristics of dynamic topology and lack of centralized management security, MANET is vulnerable to various kinds of attacks. Blackhole attack is one of many possible attacks in MANET. In this attack, a malicious node sends a forged Route REPly (RREP) packet to a source node that initiates the route discovery in order to pretend to be a destination node. By comparing the destination sequence number contained in RREP packets when a source node received multiple RREP, it judges the greatest one as the most recent routing information and selects the route contained in that RREP packet.

In case the sequence numbers are equal it selects the route with the smallest hop count. If the attacker spoofed the identity to be the destination node and sends RREP with destination sequence number higher than the real destination node to the source node, the data traffic will flow toward the attacker. Therefore, source and destination nodes became unable to communicate with each other. In [1], the authors investigated the effect of blackhole attack when movement velocity and a number connection toward the victim node are changed, and proposed the detection tech nique at the destination node. However, we can effectively avoid the attack for example by selecting the detour route during route reconstruction which achieved by detecting the attack at the source node rather than at the destination node. Thus, taking into account the detection at the source node is indispensable.

Regarding the detection of blackhole attack at the source node, [2, 3] have proposed methods in which still they are using the same training data to define the normal state. However, in MANET where the network state changes frequently, the pre-defined normal state may not accurately reflect the present network state. Therefore, using this normal state may degrade the detection accuracy.

II. METHODS AND MATERIAL

A. Related work

Adhoc On-demand Distance Vector (AODV). is a routing protocol for (MANETs) and other wireless ad-hoc networks. It establishes a route to a destination only on demand. AODV is, as the name indicates, a distance vector routing protocol. AODV avoids the *counting-to-infinity* problem of other distance-vector protocols by using sequence numbers on route updates. Each node has its own sequence number and this number increases when links change.

Each node judges whether the channel information is new according to sequence numbers. Node S is trying to establish a connection to destination D. First, the source node S refers to the route map at the start of communication. In case where there is no route to destination node D, it sends a Route Request (RREQ) message using broadcasting. RREQ ID increases one every time node S sends a RREQ. Node A and B which have received RREQ generate and renew the route to its previous hop. They also judge if this is a repeated RREQ. If such RREQ is received, it will be discarded. If A and B has a valid route to the destination D, they send a Route Reply (RREP) message to node S. By contrast, in case where the node has no valid route, they send a RREQ using broadcasting. The exchange of route information will be repeated until a RREQ reaches at node D. When node D receives the RREQ, it sends a RREP to node S. When node S receives the RREP, then a route is established. In case a node receives multiple RREPs, it will select a RREP whose the destination sequence number (Dst Seq) is the largest amongst all previously received RREPs. But if Dst Seq were same, it will select the RREP whose hop count is the smallest.

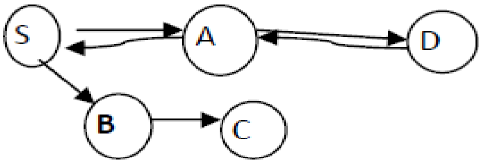


Figure 1: Route discovery process

If there is any disconnection in the route then a Route Error (RERR) message is generated and this information is sent to source [4].

B. Proposed approach

Our approach is based on metrics are different. For this we have adopted the idea of behavior (delivery, misrouted and modified) with conjunction of author's method's metrics like **hop count** and **sequence no**. As well as our method used the idea of threshold mechanism for for better approximation of black hole nodes in MANET AODV scenario. Following metrics will be used in black hole detection and prevention (Author only proposed prevention) they are listed below-

1. Packet Delivery Ratio (pkt_dr)
2. Packet Modification Ratio (pkt_mr)
3. Packet miss routed ratio (pkt_mir)
4. Hop count (hc)
5. Timestamp (ts)
6. No. of RREQ transmitted by node
7. No. of RREP transmitted by node

In networking, black holes refer to places in the network where incoming traffic is silently discarded (or "dropped"), without informing the source that the data did not reach its destination. These black hole nodes are invisible and can only be detected by monitoring the lost traffic. So, it is named as black hole. A black hole attack or packet drop attack is a type of denial of service attack accomplished by dropping packets. The attack can be accomplished either selectively (e.g. by dropping packets for a particular network destination, a packet every *n* packets or every *t* seconds, or a randomly selected portion of the packets, which is called "Gray hole attack") or in bulk (by dropping all packets). [4]

Two properties of Black Hole Attack:

1. The node exploits the ad hoc routing protocol to advertise itself as having a shortest valid route to a destination node, even though the route is spurious.
2. The node consumes the intercepted packets. [5]

Why AODV Is Prone To Black Hole Attack.

In table driven or proactive routing protocol the total routing table is shared. So, there is no chance of ondemand request or reply messages i.e. no chance of blackhole attack. Probability of black hole attack is more in reactive algorithm. AODV and DSR are the most recognized reactive (on-demand) protocol. Here black hole attack can occur. But DSR uses source routing and in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. So, AODV is much more prone to black hole attack as a black hole always responds positively with a RREP message to every RREQ, even though it does not really have a valid route to the destination node.

Comparative study can reveal that AODV is much more prone to black hole attack than other relevant attacks (like flooding attack or rushing attack). In fact the packet loss in blackhole attack is higher than any other attack under AODV protocol. The throughput of received packets in blackhole AODV decreases with the increase of number of Blackhole Nodes. Also the average End-to-end Delay without blackhole attack is increased as compared to the effect of blackhole attack. This is due to the immediate reply from the blackhole node owing to AODV protocol without checking its routing table. In blackhole attack, the attackers also have the option of manipulating only a fraction of RREP messages to reduce probability of detection.

Black Hole Attack in AODV

In AODV, Dst Seq is used to determine the freshness of routing information contained in the message from originating node. When generating a RREP message, a destination node compares its current sequence number and Dst Seq in the RREQ packet plus one, and then selects the larger one as RREP's Dst Seq. Upon receiving a number of RREP, a source node selects the one with greatest Dst Seq in order to construct a route. To succeed in the blackhole attack the attacker must generate its RREP with Dst Seq greater than the Dst Seq of the destination node. It is possible for the attacker to find out Dst Seq of the destination node from the RREQ packet. In general, the attacker can set the value of its RREP's Dst Seq base on the received RREQ's Dst Seq. However, this RREQ's Dst Seq may not present the current Dst Seq of the destination node. Figure shows an example of the blackhole attack. The value of RREQ and RREP using in the attack are shown in Table 1.

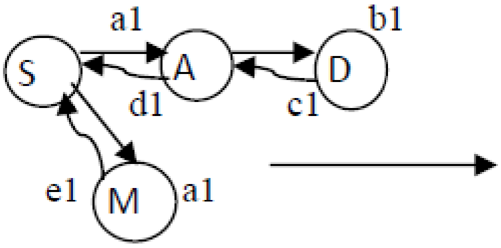


Figure 2: BlackHole Attack

Table1: Values of RREQ and RREP

	RREQ		RREP		
	a1	b1	c1	d1	e1
IP.Src	S	A	D	A	D(MD)
AODV.Dst	D		D		D(MD)
Dst Seq	60		61		65
AODV.Src	S		-		-

In Table1 IP.Src indicates the node which generates or forwards a RREQ or RREP, AODV.Dst indicates the destination node and AODV.Src indicates the source node. Here, we assume that the destination node D has no connections with other nodes. The source node S constructs a route in order to communicate with destination node D. Let the destination node D's Dst Seq that the source node S has is 60. Hence, source node S sets its RREQ (a1) and broadcasts as shown in Table. Upon receiving RREQ (a1), node A forwards RREQ (b1) since it is not the destination node. To impersonate the destination node, the attacker M sends spoofed RREP(e1) shown in Table with IP.Src, AODV.Dst the same with D and increased Dst Seq (in this case 65 as) to source node S. At the same time, the destination node D which received RREQ (b1) sends RREP (c1) with Dst Seq incremented by one to node S. Although, the source node S receive two RREP, base on Dst Seq the RREP(e1) from the attacker M is judged to be the most recent routing information and the route to node M is established. As a result, the traffic from the source node to the destination node is deprived by node M. So, blackhole node enters into the network. [4]

i. Procedure for Black Hole Detection:

Begin
Step 1: Initiate the network with two clusters and each cluster has some nodes.

```
Step 2: The cluster head is selected based on cluster election algorithm.
Step 3: Each node stores the information of its immediate neighbors in its neighbor table.
Step 4: Source node S sends a HELLO packet to the intermediate node with destination node ID and cluster ID.
Step 5: S starts timer, initializes T1
Step 6: When S get acknowledgement from destination node stop timer, T2
Step 7: The expected round trip time is computed as Te = T2 – T1
Step 8: Source provides a unique sequence number to each packet and this number is known to Source, destination and cluster head only.
Step 9: Source node S sends a packet to destination node.
Step 10: S starts timer TP1
Step 11: When S get acknowledgement from destination node stop timer, TP2
Step 12: The round trip time is calculated as Tv = TP2 – TP1
Step 13: If Tr << Te
        Step 13.1: Inform cluster head
        Step 13.2: The cluster head checks number of packet send by source node and number of packet receive by destination node .
        Step 13.3: x =no of sent packet – no of received packet.
        Step 13.4: If x >n then inform the source node to stop packet transfer.
        Step 13.5: The source node stop packet transfer and inform the CH of outer layer to inform other clusters.
        Step 13.6: CH discards that path and establishes a new path.
Step 14:Else
        Step 14.1: The cluster head calculates x.
        Step 14.2: If x is not zero then goto Step 13.1
End. [5]
```

All the nodes in an ad hoc network are categorized as *friends*, *acquaintances* or *strangers* based on their relationships with their neighboring nodes. During network initiation all nodes will be *strangers* to each other. A *trustestimator* is used in each node to evaluate the trust level of its neighboring nodes. The trust level is a function of various parameters like length of the association, ratio of the number of packets forwarded successfully by the neighbor to the total number of packets sent to that neighbor, ratio of number of packets received intact from the neighbor to the total number of received packets from that node, average time taken to respond to a route request etc. Accordingly, the neighbors are categorized into *friends* (most trusted), *acquaintances* (trusted) and *strangers* (not trusted).

In an ad hoc network, the relationship of a node *i* to its neighbor node *j* can be any of the following types:

(i) Node i is a *stranger* (S) to neighbor node j: Node i have never sent/received messages to/from node j. Their trust levels between each other will be very low. Any new node entering ad hoc network will be a stranger to all its neighbors. There are high chances of malicious behavior from stranger nodes.

(ii) Node i is an *acquaintance* (A) to neighbor node j: Node i have sent/received few messages from node j. Their mutual trust level is neither too low nor too high to be reliable. The chances of malicious behavior will have to be observed.

(iii) Node i is a *friend* (F) to neighbor node j: Note i sent/received plenty of messages to/from node j. The trust levels between them are reasonably high. Probability of misbehaving nodes may be very less. The above relationships are computed by each node and a friendship table is maintained for the neighbors. Fig. 1 shows the relationship of N4 with its neighbors. The corresponding friendship table maintained in N4 is given in Table I. The threshold trust level for a stranger node to become an acquaintance to its neighbor is represented by Tacq and the threshold trust level for an acquaintance node to become a friend of its neighbor is denoted by Tfri.

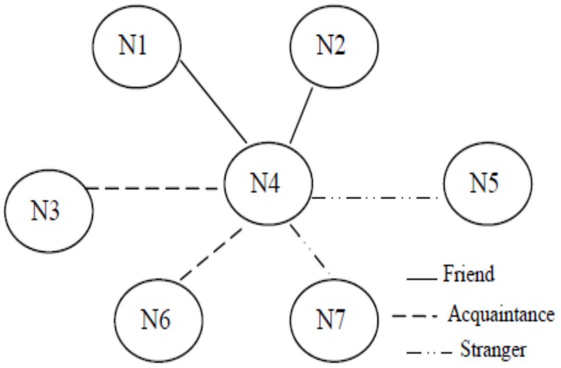


Figure 3: Trust Relationship of a node in an ad hoc network

The relationships are represented as:
R (ni →nj) = F when T ≥ Tfri
R (ni →nj) =A when Tacq ≤ T < Tfri
R (ni →nj) =S when 0 < T < Tacq

During route discovery phase of the DSR protocol, the extended system also computes the aggregate trust along different paths to the destination by the "path semiring" algorithm as proposed in [6]. From this, the most trusted path between the source and the destination is found out before establishing the data transfer. The segregation of the neighboring nodes into *friends*, *acquaintances* and *strangers* is the outcome of the direct evaluation of trust.

Table 2: FRIENDSHIP TABLE FOR NODE (N4) IN FIG. 1

Neighbors	Relationship
N1	**F**
N2	**F**
N3	**A**
N5	**S**
N6	**A**
N7	**S**

To prevent RREQ flooding, the threshold level is set for the maximum number of RREQ packets a node can receive from its neighbors. To prevent DATA flooding, the intermediate node assigns a threshold value for the maximum number of data packets it can receive from its neighbors. If rs, Xra, Xrf be the RREQ flooding threshold for a stranger, acquaintance and friend node espectively, Xrf > Xra > Xrs. If Yrs, Yra, Yrf be the DATA flooding threshold for a stranger, acquaintance and friend node respectively then Yrf > Yra > Yrs. If the specified threshold level is reached, further RREQ packets from the initiating node are ignored and dropped. Thus, flooding is prevented in the routing table.

ii. Algorithm for RREQ Flooding

```
Begin
if an intermediate node receives RREQ flooding packet from node 'i' then
1. if node 'i' is a friend and Z[i] = 0 then
2. increment X[i]
3. if X[i] > Xrf
4. drop the RREQ packet and set Z[i] = 1
5. else
6. forward the RREQ packet
7. if node 'i' is an acquaintance and Z[i] = 0 then
8. increment X[i]
9. if X[i] > Xra
10. drop the RREQ packet and set Z[i] = 1
11. else
12. forward the RREQ packet
13. if node 'i' is an stranger and Z[i] = 0 then
14. increment X[i]
15. if X[i] > Xrs
16. drop the RREQ packet and set Z[i] = 1
17. else
18. forward the RREQ packet
End
```

Let X[i] denotes the number of packets delivered from neighboring node i, where 1 ≤ i ≤ n. Xrf, Xra and Xrs are the threshold values set for *friends*, *acquaintances* and *strangers*. Let Z[i] is a Boolean array to activate or stop the prevention algorithm. The algorithm for preventing RREQ flooding is as given above. The algorithm to prevent DATA flooding is similar to the algorithm discussed in above. The threshold values for DATA flooding can be set as per the requirements of the application software.

III. RESULTS AND DISCUSSION

Simulation Results and Analysis

The ns-3 system as a whole is a fairly complex system and has a number of dependencies on other components. Along with the systems you will most likely deal with every day (the GNU toolchain, Mercurial, you programmer editor) you will need to ensure that a number of additional libraries are present on your system before proceeding. ns-3 provides a wiki for your reading pleasure that includes pages with many useful hints and tips.

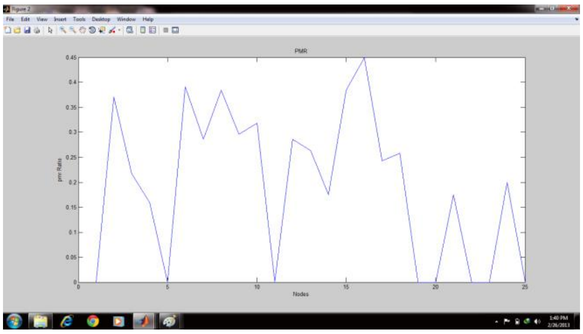
The "Prerequisites" section of this wiki page explains which packages are required to support common ns-3 options, and also provides the commands used to install them for common Linux variants. Cygwin users will have to use the Cygwin installer (if you are a Cygwin user, you used it to install Cygwin). You may want to take this opportunity to explore the ns-3 wiki a bit since there really is a wealth of information there.

From this point forward, we are going to assume that the reader is working in Linux or a Linux emulation environment (Linux, Cygwin, etc.) and has the GNU toolchain installed and verified along with the prerequisites mentioned above.

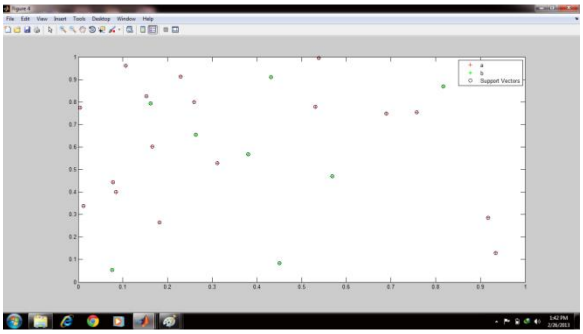
We are also going to assume that you have Mercurial and Waf installed and running on the target system. The ns-3 code is available in Mercurial repositories on the server http://code.nsnam.org. You can also download a tarball release at http://www.nsnam.org/releases/, or you can work with repositories using Mercurial. We recommend using Mercurial unless there's a good reason not to. See the end of this section for instructions on how to get a tarball release.

The simplest way to get started using Mercurial repositories is to use the ns-3-allinone environment. This is a set of scripts that manages the downloading and building of various subsystems of ns-3 for you. We recommend that you begin your ns-3 adventures in this environment as it can really simplify your life at this point.

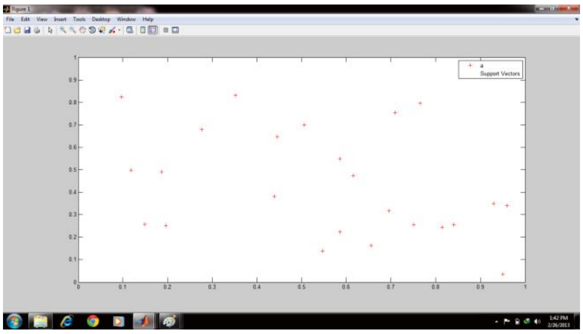
(a) Pmr result



(b) Blackhole node and authenticated node



(c) Support vector machine classifier



IV. CONCLUSION AND FUTURE WORK

With the actuality that the default AODV convention is powerless to the Blackhole assaults, in this exploration exercise, we endeavor at exploring the current answers for their reasonability. Having supported a requirement for further enhancements, we propose a calculation to counter the Blackhole assault on the steering conventions in MANETs. We effectively investigate and exhibit that with inconsequential extra overhead regarding a new MOS_WAIT_TIME variable and another Cmg_RREP_Tab table, we have the capacity to counter the Blackhole assaults on the AODV convention. From the exploratory results, we presume that the proposed arrangement accomplishes a decent ascent in PDR with satisfactory ascent in end-to-end delay. Additionally, the proposed calculation does not involve any concealed overhead on either the halfway hubs or the destination hubs. We additionally stress that however the proposed calculation is executed and reenacted for the AODV directing calculation, it can likewise be further insignificantly reached out for utilization by whatever other steering calculations, also. As a component of our future attempt, we intend to contemplate the effect of changing delay time on the convention effectiveness. Likewise, we would likewise endeavor to examine the effect of fluctuating network size and node mobility on Normalized Routing Overhead in the protocol.

V. REFERENCES

[1] W. Wang, Y. Lu, and B. K. Bhargava, “On vulnerability and protection of ad hoc on-demand distance vector protocol,” in The 10th International Conference on Telecommunications (ICT’03), vol. 1, pp. 375-382, French Polynesia, Feb. 2003.

[2] Y. A. Huang, W. Fan, W. Lee, and P. S. Yu, “Cross-feature analysis for detecting ad-hoc routing anomalies,” in The 23rd International Conference on Distributed Computing Systems (ICDCS’03), pp. 478487, May 2003.

[3] Y. A. Huang and W. Lee, “Attack analysis and detection for ad hoc routing protocols,” in The 7th International Symposium on Recent Advances in Intrusion Detection (RAID’04), pp. 125-145, French Riviera, Sept. 2004.

[4] Sanjay Ramaswamy, Huirong Fu, Manohar Sreekantaradhya, John Dixon and Kendall Nygard, “Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks” IACC 258 North Dakota State University, Fargo, ND 58105,2010.

[5] Debdutta Barman Roy, Rituparna Chaki, Nabendu Chaki, “A New Cluster-Based Wormhole Intrusion Detection Algorithm for Mobile Ad-Hoc Networks” International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 1, April 2009 6]http://en.wikipedia.org/wiki/Personal_area_network , last visited 12, Apr, 2010.

[6] Yih-Chun Hu, Adrian Perrig, David B. Johnson, ‘‘Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks”, in: Proc. of the ACM Conference on Mobile Computing and Networking (MobiCom), pp. 12–23, 2002.

[7] Panagiotis Papadimitratos, Zygmunt J. Hass, ‘‘Secure Routing for Mobile Ad Hoc Networks”, in: Proc. of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), January 2002.

[8] Semih Dokurer, Y.M. Erten, Can Erkin Acar, ‘‘Performance Analysis of Ad-hoc Networks under Black Hole Attacks”, in: Proc. of the IEEE SoutheastCon, pp. 148–153, 2007.

[9] Latha Tamilselvan, Dr. V. Sankaranarayanan, ‘‘Prevention of Blackhole Attack in MANET”, in: Proc. of the International Conference on Wireless Broadband and Ultra Wideband Communication, 2007.

[10] Latha Tamilselvan, Dr.V. Sankaranarayanan, Prevention of co-operative black hole attack in MANET, Journal of Networks 3 (5) (2008) 13–20.

[11] Satoshi Kurosawa, Hidehisa Nakayama, Nei Kato, Abbas Jamalipour, Yoshiaki Nemoto, Detecting blackhole attack on AODV-based Mobile Ad Hoc Networks by dynamic learning method, International Journal of Network Security 5 (3) (2007) 338–346.

[12] Junhai Luo, Mingyu Fan, Danxia Ye, ‘‘Black Hole Attack Prevention Based on Authentication Mechanism”, in: Proc. of the IEEE Singapore International Conference on Communication Systems (ICCS), pp. 173–177, 2008.