AODVB: Ad hoc On-Demand Distance Vector with Black Hole Avoidance

Venkatesh
Department of Computer Science and Engineering,University Visvesvaraya College of Engineering Bangalore, India

Raj Mohammed
Department of Computer Science and Engineering,University Visvesvaraya College of Engineering, Bangalore, India

ABSTRACT
A Wireless ad-hoc network is a temporary network set up by nodes moving arbitrary in the places that have no network infrastructure. The nodes find a path to the destination node using routing protocols. However, due to security vulnerabilities of the routing protocols, wireless ad-hoc networks are unprotected to attacks of the malicious nodes. One of these attacks is the Black Hole Attack against network integrity absorbing all data packets in the network. Since the data packets do not reach the destination node on account of this attack, data loss will occur. In this paper work, we propose A AODVB (Ad hoc On-Demand Distance Vector with Black-hole Avoidance) protocol for avoiding black-hole attack. AODVB forms link disjoint multi-path during path discovery to provide greater path selection in order to avoid malicious nodes in the path using legitimacy table maintained by each node in the network. Non-malicious nodes gradually isolate the black-hole nodes based on the values collected in their legitimacy table and avoid them while making path between source and destination. We simulated AODV protocol with and without Black-hole attack and our solution AODVB protocol. From our simulation results AODV network has normally 3.21 % data loss and if a Black Hole Node is introducing in this network data loss is increased to 92.59 %. When we used AODVB protocol in the same network, the data loss decreased to 65 %.

Keywords
Black Hole Attack, link disjoint multi-path, legitimacy table.

1. INTRODUCTION
As wireless ad-hoc networks lack an infrastructure, they are exposed to a lot of attacks. One of these attacks is the Black Hole attack. In the Black Hole attack, a malicious node absorbs all data packets in itself, similar to a hole which sucks in everything in. In this way, all packets in the network are dropped. A malicious node dropping all the traffic in the network makes use of the vulnerabilities of the route discovery packets of the on demand protocols, such as AODV. In route discovery process of AODV protocol, intermediate nodes are responsible to find a fresh path to the destination, sending disovery packets to the neighbor nodes. Malicious nodes do not use this process and instead, they immediately respond to the source node with false information as though it has fresh enough path to the destination. Therefore source node sends its data packets via the malicious node to the destination. assuming it is a true path. Black Hole attack may occur due to a malicious node which is deliberately misbehaving, as well as a damaged node interface.

2. RELATED WORK
The methods proposed to avoid blackhole earlier fall broadly into two categories. The first category is of those which modify specific well known routing protocols such as AODV, DSR and OLSR to avoid/detect blackhole attack during route reply [1] [2]. The second category is of those which adopt an extra monitoring system such as a watchdog, confidant protocol or intrusion detection system [3] [4]. There are several methods proposed to add security measures for routing protocols to avoid attacks [5] - [12].

Shurman et. al. [1] proposed two different approaches to solve the blackhole attack problem. First, the sender node verifies the authenticity of the node that initiates the RREP packet by utilizing the redundancy of the network. Second, each node stores the last and received sent packet sequence number. If there is any mismatch then an ALARM indicates the existence of a black hole node. However, this approach unable to detect multiple blackhole attacks.

Tamilselvan et. al. [2] proposed an enhancement of the AODV protocol by introducing fidelity table. The RREPs are collected in the response table and the fidelity level of each RREP is checked and one is selected having the highest level. After acknowledgement is received, the fidelity level of the node is updated proving it safe and reliable. However, updating the fidelity table of each node by broadcasting it to other nodes results in congestion and also the selection of wrong RREP from the response table cause another route request flooding.

Marti et. al. [3] described the misbehavior detection using the watchdog and the pathrater. The watchdog identifies misbehaving nodes by listening promiscuously to the next node transmission whereas the pathrater uses the knowledge from the watchdog to choose a path that is most likely to deliver packets. This technique is imperfect due to limited transmit power, collision and partial dropping.

Burchegger et. al. [4] described the confidant protocol where each node monitor the behavior of its next hop and this information is given to the reputation system which makes decisions based on ratings about providing or accepting route from it. However, the use of reputation system makes this protocol impractical to include in adhoc network.

In D. P. Agrawal et. al. [5], the authors discuss a protocol that requires the intermediate nodes to send RREP message along with the next hop information. When the source node gets this information, it sends a RREQ to the next hop to verify that the target node (i.e. the node that just sent back the RREP packet) indeed has a route to the intermediate node and to the destination. When the next hop receives a FurtherRequest, it sends a Further Reply which includes the check result to the source node. Based on information in FurtherReply, the

source node judges the validity of the route. In this protocol, the RREP control packet is modified to contain the information about next hop. After receiving RREP, the source node will again send RREQ to the node specified as next hop in the received RREP.

Sanjay Ramaswamy, et al [6] proposed a method for identifying multiple black hole nodes. They are first to propose solution for cooperative black hole attack. They slightly modified AODV protocol by introducing data routing information table (DRI) and cross checking. Every entry of the node is maintained by the table. They rely on the reliable nodes to transfer the packets.

Hesiri Weerasinghe et. al. [7] proposed the solution which discovers the secure route between source and destination by identifying and isolating cooperative black hole nodes. This algorithm uses a methodology to identify multiple black hole nodes working collaboratively as a group to initiate cooperative black hole attacks. This protocol is a slightly modified version of AODV protocol by introducing Data Routing Information (DRI) table and cross checking using Further Request (FREQ) and Further Reply (FREP).

All material on each page should fit within a rectangle of 18 x 23.5 cm (7" x 9.25"), centered on the page, beginning 2.54 cm (1") from the top of the page and ending with 2.54 cm (1") from the bottom. The right and left margins should be 1.9 cm (.75"). The text should be in two 8.45 cm (3.33") columns with a .83 cm (.33") gutter.

3. PROPOSED AODVB ROUTING PROTOCOL

The routing protocol AODVB is based on AODV and it can efficiently avoid multiple blackhole attacks during path setup between source and destination. When intermediate nodes reply to source node, few nodes in the path may have multiple paths to the destination but it eventually chooses only one path to destination node. In AODVB, every node maintains the legitimacy of their neighbor nodes to form the correct path to destination node. In the path discovery of AODVB, an intermediate node will attempt to create a route that does not go through a node whose legitimacy ratio crosses the lower threshold level. Therefore, malicious nodes will be gradually avoided by other non-malicious nodes in the network. Compared with AODV, the proposed AODVB has the following differences in message format and type. RREQ Packet: RREQ in AODVB has additional $first\_hop$ field shown in Fig. 3. This field is used to store the IP address of the first hop after it left the originator. Intermediate nodes would not process the RREQs which has the same $first\_hop$ field value. AODVB creates link disjoint multiple paths in path discovery phase using $first\_hop$ field but during path setup (RREP) it chooses only single link which has the higher legitimacy ratio among multiple links towards source and destination discussed later.

Types	J	R	D	G	U	Reserved	Hop Count
$first\_hop$						RREQ ID	
Originator IP Address							
Originator Seq Number							
Destination IP Address							
Destination Seq Number							

Figure 1. RREQ in AODVB

REP Packet: RREP in AODVB has one additional field called $originator$ as shown in Fig. 4 This field is used to store the identity of the node (can be intermediate or destination node) who is claiming a path to the destination. This field value is being stored in the $First\_hop$ field of routing table when node receives RREP.

Types	R	A	Reserved	Prefix Size	Hop Count
Source IP Address					
Destination IP Address					
Destination Seq Number					
Lifetime					
$Originator$					

Figure 2: RREP packet in AODVB

Legitimacy Table: In AODVB, each node maintains a legitimacy table as shown in Fig.3(a) to choose the most legitimate node (among the multiple backward disjoint link to source node and next hop to destination) while sending RREP back to source node. Legitimacy table contains three fields: $NodeID$, $Pathcount$ and $Sentcount$. $NodeID$ Field stores the IP address of the node whose legitimacy is being recorded. $Pathcount$ Field specifies the number of times the node has been chosen in the route and the $Sentcount$ field describes the number of times connection to destination have been successful node through the $NodeID$.

These two count field are also used to define the Legitimacy Ratio = ($Sentcount$/($Pathcount$+1)) of a $NodeID$ which indicates the confidence of node in performing its intended function of correct routing. A higher legitimacy ratio means higher possibility of a node being non-malicious.

Node ID	Pathcount	Sentcount
-----	-----	-----
A	3	3
B	4	2
-----	-----	-----

Figure 3. Illustration of (a) Legitimacy Table

Source Address
Destination Address
changeBit

Figure 3 (b) Route_Change Packet

Route_Change Packet: This is an additional packet used in the AODVB protocol shown in Fig. 5 (b). The packet has only three fields and it is used by nodes (a) to inform the first node in the backward path (having multiple entries for destination) to change the route to another path which has next highest legitimacy ratio and (b) to flush the counter of all nodes in the backward path. $changeBit$ in the packet has special purpose i.e it will be set to 1 by the first node in the backward path which has multiple entries to the destination node, so that other nodes in the backward path would not switch the route to another path.

Routing Table: Routing table in AODVB has three additional fields $First\_hop$, $validBit$ and $Count$ shown in Fig. 6 $First\_hop$ field is used to store the value of $first\_hop$ field of RREQ to avoid loop in the path formation. However, when a node receives an RREP, this field is used to store the value of $originator$ field of RREP. $validBit$ field has only three values 0, 1 and -1. Value 0 indicates that the path to the

destination through next hop may not be correct; value 1 specifies that the path to the destination is free from malicious nodes and value -1 indicates entry has not been chosen for data transfer. $count$ field denotes the number of RREPs received with same sequence number for the entry but its value would be -1 if the entry has been created after RREQ arrival.

Destination Sequence Number
Destination IP Address
$First\_Hop$
$validBit$
$Count$
Hop Count
Next Hop

Figure 4: Routing Table in AODVB

HELLO Packet: AODVB modifies the function of HELLO packet. In AODVB, HELLO packet are also used to broadcast the $NodeID$ whose legitimacy ratio cross the lower threshold level among its 1-hop away neighbors. If a node's neighbor has an entry to $NodeID$ in its legitimacy table and legitimacy ratio of sending node is higher than the upper threshold level, then neighbors will update their legitimacy table for $NodeID$ so that malicious node will not be able to grab the route through them.

4. PROCEDURE FOR RECEIVING RREQ IN AODVB PROTOCOL

In AODVB, each node uses three fields: source IP, sequence number and $first\_hop$ to determine whether an RREQ is duplicate or not whereas AODV uses only first two fields. For an intermediate node, if the hop count in the RREQ is larger than the hop count of the entry in the routing table which has the same sequence number and source IP, then RREQ is directly dropped. A node would create multiple entries (multiple link disjoint path) when the sequence number is same; hop count is smaller and the $first\_hop$ field value is different from the existing reverse entries. However, the destination node replies to each RREQ inspite of the values in hop count and $first\_hop$ field of RREQ when the sequence number is equal or larger than the existing entry.

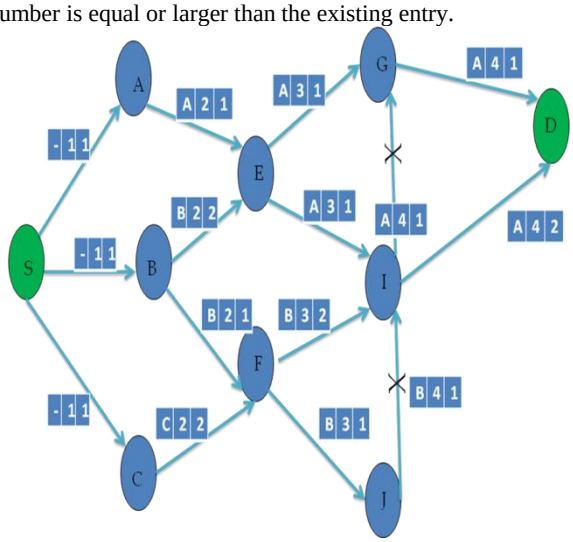

Figure 5: RREQ processing in AODVB

Fig. 5 illustrates a situation, in which S and D represent the source and the destination respectively. The arrow represents the RREQ broadcasting traces; the first box represents the $first\_hop$ of the RREQ; second box represents the hop count and the third box represents the sequence of the RREQ arrivals at the node. As the nodes A, B and C received RREQs, they find hop count in the RREQ is 1. Therefore, they place their own IP addresses into the $first\_hop$ field of the RREQ and then broadcast the packet after adding 1 to the hop count.

Accordingly, node E receives the first RREQ from node A and creates a reverse entry with hop count=2, $first\_hop$=A, $validBit$=1 and $count$=-1. Then, it continues to broadcast the RREQ after adding 1 to the hop count. Later, node E receives another RREQ from node B, the RREQ has the same source IP and sequence number as the previous RREQ from A and the hop count is smaller or equal to the entry in the routing table but a different $first\_hop$. Therefore, node E would create another reverse entry in its routing table and drop the RREQ. Similarly, node F process both RREQ arrival from node B and C. In sequence, node I will receive RREQ from node E, F and J. The first two arrivals of RREQ are processed by node I in a similar way to node E in previous case. But on receipt of last RREQ from node J, node I would drop the RREQ without creating a reverse entry because the hop count (i.e. 4) is larger than the hop count in the existing entries created by the previous two RREQs although the $first\_hop$ is same. Node G receives two RREQ from node E and node I respectively with the same hop count; however it creates reverse entry for the first arrival of RREQ and drops the second RREQ because the $first\_hop$ fields of both are same.

5. PROCEDURE FOR RECEIVING RREP IN AODVB ROUTING PROTOCOL

Regardless of the number of RREQs received, the destination node will reply to each RREQ unless the sequence number of RREQ is not smaller than the existing sequence number in the routing entry. Intermediate nodes will reply to RREQ only when they have an entry to the destination node with $validBit$=1 in the routing table. Node receiving an RREP from any of its neighbour will first check the legitimacy ratio of the neighbour. If the legitimacy ratio crosses the lower threshold level than the node will drop the RREP, otherwise will forward it to the neighbour which has the highest legitimacy ratio among multiple backward entries and delete the other backward entries. Whenever a node creates an entry in the routing table after receiving RREP, it sets the $validBit$ as 0 and and $count$ field as 1. A node who is generating RREP must store its identity into the $originator$ field of the RREP packet. The intermediate node will create single forward entry in the routing table regardless of the RREPs recieved (having same sequence number) with next hop set to the neighbour which has the least hop count to the destination and set the $count$ field of the entry to the number of RREP received. Malicious node may send the RREP with its own identity or with the identity of the destination node (spoofing) in the $originator$ field of the RREP. Malicious node replies with higher sequence number because they do not know the exact sequence number of the destination node. Intermediate nodes forward only the first RREP arrival and drop the others. Intermediate node store the identity of the replying node in the $First\_hop$ field of the routing entry. Fig. 8 illustrates an example, in which S and D represents the source and the destination respectively. The arrow represents the neighbour

chosen among multiple backward entries because of its high legitimacy ratio; the first box represents the *originator* of the RREP; second box represents the hop count and the third box represents the sequence of the RREP arrivals at the node. B1 is one blackhole node where B1 replies with its own identity and B2 with the destination node identity (spoofing). We have assumed that the node G knows a path to the destination node via node M. Node G receives two RREP from the destination. On receipt of RREP, G will create forward entry towards destination and forwards the packet to the only backward entry node E. After second RREP arrival, node G finds an entry to the destination having same sequence number. Therefore, it retains a single entry towards destination which has least hop count i.e first entry and sets the *count* field in the routing entry to 2. When node I receives the reply, it would check its legitimacy table to find which backward entries (E or F) has higher

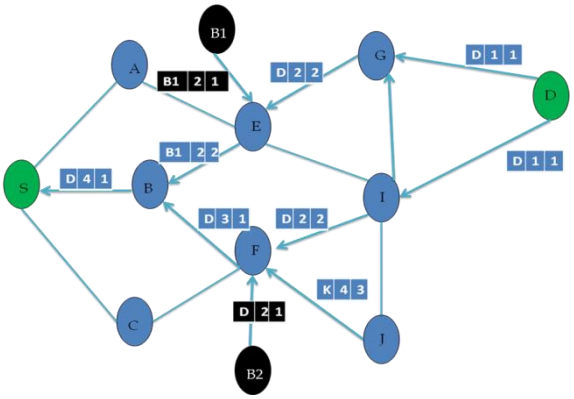

Figure 6: RREP processing in AODVB

legitimacy ratio and eventually chooses node F. In sequence, node F receive three replies, i.e from B2, I and J. Node B2 replies with higher sequence number and spoofs the destination node identity in the *originator* field, whereas, node K replies with its own identity. Next hop would have been chosen on the following criteria:

- If reply from I and J are having same sequence number, then node F would create two forward entries; first towards either I or J based on their legitimacy ratio if they have same hop count to destination, otherwise to the node which has lower hop count to destination with *count* field set to 2; and the second entry towards B2. Node F will choose first entry for forwarding data packets by setting its $validBit$=0 and others $validBit$=-1 because first entry *count* field is higher than second entry which means the higher possibility of correct path through first entry.

If reply of node K has an older sequence number of node D, then each reply has different sequence number. Node F would create three entries for each reply comes from B2, I and J respectively. The RREPs having destination address in the *originator* field had a higher probability of correct path to destination than other RREPs because RREP originated by other nodes (than destination node) may claiming an older path to destination. Since two replies had different sequence number with the *originator* field containing destination address (it means any of them comes from malicious node), node F would choose forward entry whose next hop has higher legitimacy ratio inspite of their hop count by setting its $validBit$=0 and others with -1. If both entries have similar legitimacy ratio, then node F randomly choose any of them.

Accordingly, node E receives the first reply from B1 and it copies the reply content into the routing entry and forward the reply to the backward entry having higher legitimacy ratio i.e . When node E received another reply from node G which had *originator* field filled with the destination address and different sequence number with an existing entry, then node E sets the second entry for data transfer regardless of the legitimacy ratio of next hop by setting its $validBit$=0 and others $validBit$=-1. But, it will not delete previous entry, because the second RREP could have been generated by malicious node by spoofing destination address (not in this case). Node B will perform in a similar way to node E. After the source node receives the reply, it starts sending data packet to the destination node. While forwarding data packet, each node in the path will set the counter to an interval so that it would get data packet reply or *Route_Change* packet within the interval time. Otherwise, as the node counter interval period expires, it would increment the *Pathcount* field of the next hop in the legitimacy table and send the *Route_Change* packet to the backward entry node. Counter interval is $t = (15 - HopC)(2\{d/v\} + \delta)$ where v is the velocity of the light, d is the maximum transmission range, δ is the processing time of the node, $HopC$ is the hopcount between the source node and the intermediate node which set the counter and 15 is assumed as maximum hop count in the adhoc network.

We have assumed that S, B, F and B2 is the path formed for the RREQ sent by the source node. Nodes in the path i.e S, B and F will set the counter as they forward the data packet. Since B2 is the blackhole node, it will drop the packet. When the counter interval of node F expires, it would increment the *Pathcount* field of node B2 in its legitimacy table and delete the corresponding entry in the routing table and send the *Route_Change* packet to node B with *changeBit* set to 1 (because node F has change the path to node I). Node B flushes the counter and does not make a decision to switch to next path because the node ahead in the route already had changed the path by setting *changeBit*=1, then each node send the *Route_Change* packet along reverse route unless source node reached.

Assume remaining entries in the routing table of node F towards destination node results in to dropping, then Node F on last entry would send *Route_Change* packet to node B with *changeBit*=0. After the complete data has been transferred, destination node will send final data acknowledgement (REP_ACK) packet back to source node. Each node in the path will change the sequence number of the forwarding entry as listed in the REP_ACK packet and set the $validBit$=1 in the routing entry; delete other entries in the table towards destination; and increment the *Pathcount* and *Sentcount* field of the previous hop and next hop of REP_ACK in the legitimacy table.

6. PERFORMANCE MATRICS

We choose the following parameters to give an idea of behavior and reliability of AODVB protocol:

a. Packet Delivery Ratio: it is ratio of number of packets delivered to destination to the total number of packets sent by the source in presence of blackhole nodes.

b. Route Formation Delay: It is time taken to form a candid path from source to destination

c. Node Speed: it is the speed of nodes moving in the network and we shall check the performance of AODVB on different node speed

d. Pause Time: it is used as mobility metric that expresses the period of node in pause but cannot reflect other information such as a node location or velocity. It is used to find the behavior of AODVB before and after the node start movingly.

7. SIMULATION RESULTS

In our example we have taken 7 nodes. All nodes are randomly distributed in area of 500*500 square meter with each node having radio range of 250 meters. All nodes are having the same configuration. Node-2 will act as sender whereas node-5 will be the receiver. node-0 will behave as the black-hole node in the network. Node 0 being a Black Hole AODV Node absorbs the packets in the connection from Node 2 to Node 5. Figure 9 shows how the Black Hole AODV Node absorbs the traffic.

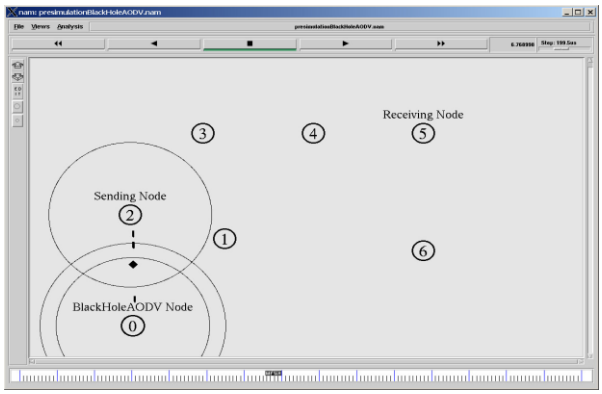

Figure 7: Node 0 (Black Hole Node) absorbs the connection Node 2 to Node 5

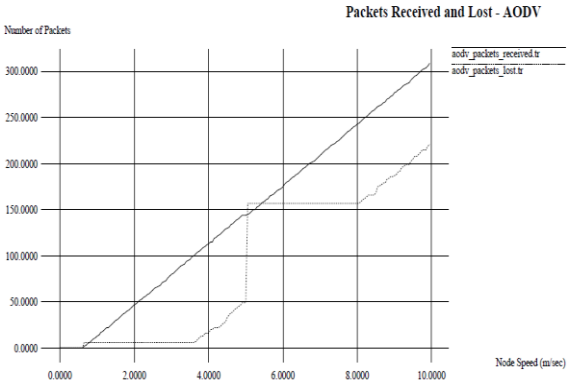

Figure 8: Packets received and lost in AODV

The above graph is plotted for packet received and lost parameters for AODV protocol where there is no black-hole in the network. The loss is because of network and high mobility of the nodes. Whereas the below graph is for same parameters but in the presence of black-hole attack. Here the packets received and lost are shown as zero because the legitimate. receiver did not receive any packets the black-hole node consumed all data packets.

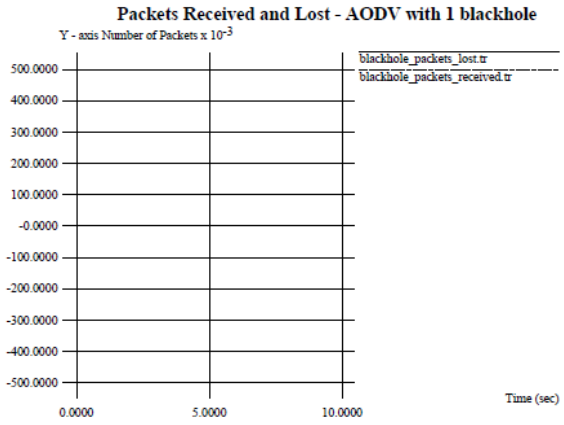

Figure 9: Packets received and lost in AODV with 1 blackhole.

The above graph is plotted for packet received and lost parameters for AODVB protocol where there is one blackhole in the network. Even in the presence of blackhole node the AODVB protocol detects it and sends data to legitimate receiver. The loss is because of network and high mobility of the nodes.

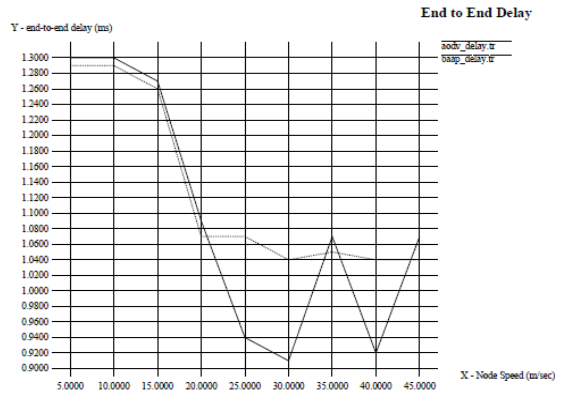

Figure 10 End to End Delay for both AODV and AODVB

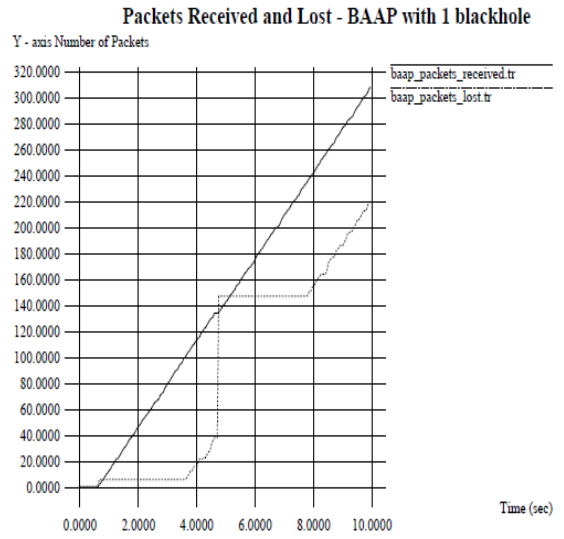

Figure 11: Packet Loss Percentage between AODV, blackhole AODV and AODVB.

This graph is plotted for packet loss percentage for all three protocol AODV, AODV with one blackhole node and the AODVB. AODV packet loss will be 45%. In presence of

blackhole it is increased to 99%. Our protocol AODVB gives less loss in the presence of even blackhole.

8. CONCLUSION

Having simulated the Black Hole Attack, we saw that the packet loss is increased in the ad-hoc network. graphs of simulation results show the difference between the number of packets lost in the network with and without a Black Hole Attack. This also shows that Black Hole Attack affects the overall network connectivity and the data loss could show the existence of the Black Hole Attack in the network. If the number of Black Hole Nodes is increased then the data loss would also be expected to increase.

We can understand from simulation results AODV network has normally 3.21 % data loss and if a Black Hole Node is introducing in this network data loss is increased to 92.59 %. As 3.21 % data loss already exists in this data traffic, Black Hole Node increases this data loss by 89.38 %. When we used AODVB protocol in the same network, the data loss decreased to 65 %. These two results show that our solution reduces the Black Hole effects by 24.38 % as packet loss in a network using AODVB and where there is no black holes increases to 75,62 %.

9. REFERENCES

[1] M.A. Shurman, S.M. Yoo, and S. Park, "Black hole attack in mobile adhoc networks," 42nd ACM Southeast Regional Conf., 2004, pp. 11-14.

[2] L. Tamilselvan, and V. Sankaranarayanan, "Prevention of cooperative black hole attack in manet", Journal of Networks, Vol. (5), 2008, pp.13-20.

[3] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile adhoc networks", Proceedings of the ACM Conf. on Mobile Computing and Networking (Mobicom), 2000, pp. 255-265.

[4] S. Buchegger, and J. Le Boudec, "A testbed for misbehavior detection in mobile adhoc networks-how much can watchdogs really do", Technical Report IC/2003/72 EPFL-DI-ICA, 2003. pp. 32-41.

[5] H. Deng, W. Li, and D. P. Agrawal. "Routing Security in Adhoc Networks." In: IEEE Communications Magazine, Vol. 40, No. 10, pp. 70-75, Oct. 2002.

[6] Sanjay Ramaswamy, Huirong Fu, Manohar Sreekantaradhya, John Dixon, and Kendall Nygard, "Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks", 2003. International Conference on Wireless Networks (ICWN-03), Las Vegas, Nevada.

[7] Hesiri Weerasinghe and Huirong Fu, Member of IEEE, Preventing Cooperative Black Hole Attacks in Mobile Adhoc Networks: Simulation Implementation And Evaluation, IJSEA,Vol2,No.3,July 2008.

[8] Y.C. Hu, D.B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks," Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, June 2002, pp. 3-13.

[9] Y.C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks," Proc. 8th ACM Int'l. Conf. Mobile Computing and Networking (Mobicom'02), Atlanta, Georgia, September 2002, pp. 12- 23.

[10] Kimaya Sanzgiti, Bridget Dahill, Brian Neil Levine, Clay shields, Elizabeth M, Belding-Royer, "A secure Routing Protocol for Ad hoc networks" In Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP' 02), 2002.

[11] S. Yi, P. Naldurg, and R. Kravets, "Security-Aware Ad hoc Routing for Wireless Networks," Proc. 2nd ACM Symp. Mobile Ad hoc Networking and Computing (Mobihoc'01), Long Beach, CA, October 2001, pp. 299-302.

[12] M. Zapata, ―Secure Ad Hoc On-Demand Distance Vector (SAODV), Internet draft, draft-guerrero-manet-saodv-01.txt, 2002.

[13] The Network Simulator Wiki. [Online]. http://nsnam.isi.edu/nsnam/index.php/

[14] The Network Simulator – ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[15] M. Greis. Tutorial for the Network Simulator NS2. [Online]. Available: http:// www.isi.edu/nsnam/ns/tutorial/

[16] C.Perkins, "(RFC) Request for Comments – 3561", Category: Experimental, Network, Working Group, July 2003.

[17] F. J. Ros and P. M. Ruiz, "Implementing a New Manet Unicast Routing Protocol in NS2", December, 2004, http://masimum.dif.um.es/nsrt-howto/pdf/nsrt-howto.pdf