ISSN:0975-9646

Denial of Service Attacks in Wireless Mesh Networks

Monika
Department of computer science
Lovely Professional University
Phagwara (PB), India

Abstract **- A Wireless mesh network is a wireless communication between different nodes which are dynamically self-organized and self-configured. The nodes in the network automatically establishing an ad-hoc network and maintaining the mesh connectivity, nodes communicate with each other by forwarding data packets to other nodes in the network. Thus the nodes find a path to the destination node using routing protocols. However, due to security vulnerabilities of the routing protocols, wireless mesh networks are unprotected to attacks of the malicious nodes. One of these attacks is the Black Hole Attack and gray hole attack , against network integrity absorbing all data packets in the network and gray hole random drop the packets. Since the data packets do not reach the destination node on account of this attack, data loss will occur. There are lots of detection and defense mechanisms to eliminate the intruder that carry out the black hole attack. In this thesis, I have simulated the black hole attack in various wireless mesh network scenarios and have tried to find a response system in simulations. Denial of service attacks is one of the most common types of attack which is possible in WMNs. DoS attacks are most common in networks which connect to internet and since WMNs are mainly designed for fast and long distance internet access this type of attacks are common in the network. Wireless mesh networks consist of both mesh routers and mesh clients. I have confined my studies to mesh routers which are stationary. I have implement both Gray Hole attack and black hole attack in mesh routers and study the delivery ratio of the network with and without the presence of attack routers. By simulating the scenario with AODV protocol I have study the delivery ratio of packets and find out how it is affecting the network in the presence of an attack router.**

***Keywords*-** **Wireless Mesh Network (WMN), Network Simulator(NS), Denial Of Service Attack(Dos), AD-Hoc On Demand Distance Vector(Aodv).**

I.INTRODUCTION

Wireless mesh network: Wireless mesh networks (WMNs) are a multi-hop wireless communication among different nodes are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad-hoc network and maintaining the mesh connectivity. WMNS are emerged as a promising concept to meet the challenges in wireless networks such as flexibility, adaptability, reconfigurable architecture etc. A wireless mesh network enables ad-hoc mode peer to peer interconnection among mesh clients are is called client meshing [2]. With client meshing, mesh routers that stay outside the radio coverage of a mesh router can rely on other intermediate clients to relay packets to them to get WMN access network connections. Thus packets from a mesh client which lies far away from the mesh router has to travel multi hop client-to-client and client-to-router wireless link before reaching its destination. The number of hops is determined by the geographical location of the client and also the organization structure of the access network. The architecture of wireless mesh networks can be classified in to three main groups based on the functionalities of the nodes namely infrastructure/backbone WMNs, client WMNs and Hybrid WMNs. In infrastructure WMNs wireless mesh routers will form a mesh of self-configuring, self healing links among themselves. With gateway functionality these routers can be connected to the internet. This approach provides backbone for conventional clients and enables integration of WMNs with existing wireless networks, through gateway/bridge functionalities in mesh routers. In client meshing the client devices will form a mesh to perform routing and configuration functionalities as well as providing end-user applications to users. In this architecture no mesh routers are present and thus are same as the conventional ad-hoc network. Hybrid WMNs is the combination of infrastructure and client meshing and a mesh network is formed between the clients and as well as the routers. Mesh clients can access the network through mesh routers as well as directly meshing with each other. Wireless mesh network in which nodes send and receive data by using mesh network.[5] Wireless mesh network is a wireless communication between different nodes which are self organized and self configured with nodes in network and construct Ad hoc network maintain connectivity of mesh network.

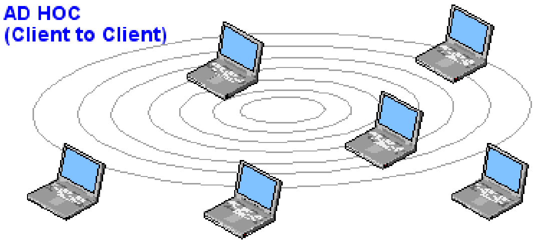


Figure 1 wireless mesh network

A wireless mesh network is a mesh network established through the connection of wireless access points installed for each network user's local. Each network user sends data to the next node. The wireless mesh network infrastructure is decentralized because each node need only transmit data to the next node. Wireless mesh networking can be used in remote areas and small business operating in rural neighborhoods to connect their networks together for affordable Internet connections. The Wi-Fi card in your laptop might become an access point to it perform role as network client. Wireless mesh network consist of two nodes: mesh router and mesh client. In the mesh router, stationary part of mesh network with less power constraint and form of backbone network. Mesh router can perform all bridges function as used in conventional wireless router. They support multiple wireless interfaces and technologies. Mesh router are dedicated and stationary node route data with less power constraint. Mesh client are route by forwarding packets from one node to next node. Mesh Client used simple form of Hardware and software as compare to mesh router. Mesh client no use of gateway and bridges function, only use one wireless interface.

A. Denial of Service Attacks in WMNS

1) Rushing attacks: Rushing attacks targeting the on-demand routing protocols were amongst the first exposed attacks on the network layer of multi-hop wireless networks. These attacks exploit the route discovery mechanism of on-demand routing protocols. In these protocols, the node requiring the route to the destination floods the Route Request message, which is identified by a sequence number.[4].

To limit the flooding, each node only forwards the first message that it receives and drops remaining messages with the same sequence number. To avoid collusion of these messages, the protocols specify a specific amount of delay between receiving the Route Request message by a particular node and forwarding it. The malicious node launching the rushing attack forwards the Route Request message to the target node before any other intermediate node from source to destination. This can be easily achieved by ignoring the specified delay.

2) Worm Hole attacks: A wormhole attack's objective is similar to rushing attack but the technique used is different. During a wormhole attack, two or more malicious nodes collude together by establishing a tunnel using an efficient communication medium (wired connection or high-speed wireless connection, etc.). During the route discovery phase of on-demand routing protocols, the Route Request messages are forwarded between the malicious nodes using the tunnel [5]. Therefore, the first Route Request message that reaches the destination node is the one forwarded by the malicious node. Consequently, the malicious nodes are added in the path from source to destination. Once the malicious nodes are included in the routing path, the malicious nodes either drop all the packets, resulting in complete denial of service, or drop the packets selectively to avoid detection.[5]

A black hole attack (or sink hole attack) also leads to denial of service in wireless mesh networks. It also exploits the route discovery mechanism of on demand routing protocols. In a black hole attack, the malicious node always replies positively to a Route Request even if it may not have a valid route to the destination. Because the malicious node does not check its routing entries, it will always be the first to reply to the Route Request message [6]. Hence almost all the traffic within the neighborhood of the malicious node will be directed towards the malicious node, which may drop all the packets, resulting in denial of service [7].

3)Gray Hole attacks: A Gray Hole attack is a variant of the black hole attack. In a black hole attack, the malicious node drops all the traffic that it is supposed to forward. This may lead to possible detection of the malicious node [11], [12]. In a Gray Hole attack, the packets are dropped selectively, thus avoiding the detection. A Gray Hole attack does not lead to complete denial of service, but it may go undetected for a longer duration of time. This is because the malicious packet dropping may be considered as congestion in the network [13]. This also leads to selective packet loss.

II. AD-HOC ON-DEMAND DISTANT VECTOR ROUTING PROTOCOL

AODV protocol is one of the commonly used in wireless mesh networks and is proposed as one of the protocol in the IEEE 802.11s standard [16]. AODV is a reactive distance vector routing protocol which will establish the path only when the router has some data to send. AODV borrows the basic route establishment and maintenance mechanisms from the Dynamic Source Routing protocol (DSR) and the hop-to-hop routing vectors from the Destination-Sequenced Distance-Vector protocol (DSDV). To avoid routing loops AODV makes use of the sequence number in the control packets. When source node intends to communicate with a destination node whose route is not known it broadcasts a Route Request packet (RREQ). Each RREQ contains an ID which uniquely identifies the RREQ packet, source and destination IP addresses and sequence numbers together with the various control flags. The sequence number maintains the freshness of the control messages and the hop count maintains the number of nodes between the source and the destination. On receiving a RREQ message by the intermediate or neighboring node that has not seen a source IP and ID pair or which doesn't contain a fresher route (larger sequence number) to destination will rebroadcast the packet after incrementing the hop count. Such intermediate nodes will also create a reverse route to the source node for a particular interval of time. When the RREQ reaches the destination node or any intermediate node which has a fresher route to the destination a Route Reply (RREP) packet is generated and uni-cast backward to the source of the RREQ. Each RREP contains the destination sequence number, source and destination IP addresses route life time and the hop count together with control flags. Each intermediate node receiving a RREP packet will increment the hop count and establishes a forward route to the source of the packet and send the RREP packet in the backward route. A Route Error (RERR) packet is send by a node to its neighboring nodes if there is a link break observed in the active route. Once the route is updated by all the nodes the source will send packet to the destination in the route. When using AODV in multiple radios in a node, the RREQ is broadcasted on all the interfaces of the node. In order to avoid broadcast storms each RREQ is send after a random time interval. Intermediate node with more than one interfaces and working on a channel will receive the RREQ and create a reverse route to the source of the packet. If the RREQ is a duplicate it is simply discarded. The first RREQ received by the destination or an intermediate node having route to the destination is selected and all the other RREQs are discarded. Then RREP is generated for the selected RREQ and is send back to the source of the RREQ in the reverse path.

AODV uses Route Request (RREQ), Route Reply (RREP) control messages in Route Discovery phase and Route Error (RERR) control message in Route Maintenance phase. AODV is a hop by hop routing protocols developed for wireless ad-hoc networks [15]. It offers quick adaptation to dynamic link conditions, low processing and memory overhead. When a host wants to find a route to a destination it broadcast a route request (RREQ) message. The RREQ contains addresses (source and destination), sequence number and a broadcast identifier. Nodes other than destination receiving RREQ message either re-broadcast or respond with route reply (RREP), depending on flags setting in RREQ message. When forwarding a RREQ node stores broadcast identifier, source address and maintains a reverse route. In order to avoid loop, RREQ are re broadcasted only when a request with the same source address and broadcast identifier has not been processed before. Concept of sequence number is used for route updating. Thus an intermediate host replies with a RREP when it has a fresh enough route to the destination. Figure 2 shows a typical example of route discovery using AODV protocol. RREQ message was broadcasted by source node. Intermediate node creates and maintains a reverse route to the source node.

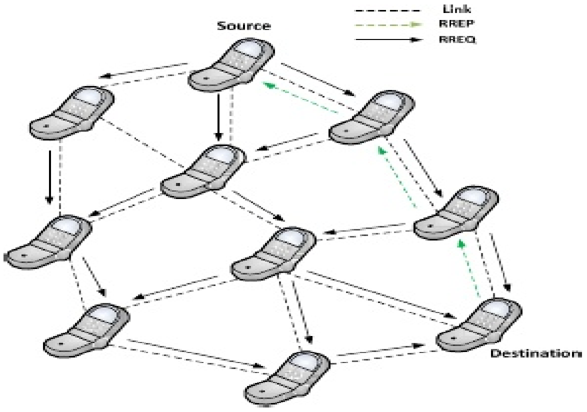


Figure 2 Route discovery using AODV protocol

Destination node, on receiving RREQ sends a uni-cast RREP to the source node on the same path that was created during RREQ. The incoming RREPs from the source node are processed. After a source node receives a RREP message, it calls Receive Reply (Packet P) method - one of the crucial functions of AODV. For every RREP control message received, the source node would first check whether it has an entry for the destination in the route table or not. If it finds one, the source node would check whether the destination sequence number in the incoming control message is higher than one it sent last in the RREQ or not. If the destination sequence number is higher, the source node will update its routing table with the new RREP control message; otherwise the RREP control message will be discarded. In Route Maintenance phase, if a node finds a link break or failure, then it sends RERR message to all the nodes that uses the route.

III. NEW SECURITY EXTENSION TO AODV IN NS-2

A) NS2 Overview

NS2 is an object-oriented, discrete event driven network simulator which was originally developed at University of California-Berkley. The programming it uses is C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT). The usage of these two programming language has its reason. The biggest reason is due to the internal characteristics of these two languages. C++ is efficient to implement a design but it is not very easy to be visual and graphically shown. It's not easy to modify and assembly different components and to change different parameters without a very visual and easy-to-use descriptive language. Moreover, for efficiency reason, NS2 separates control path implementations from the data path implementation. The event scheduler and the basic network component objects in the data path are written and compiled using C++ to reduce packet and event processing time. OTcl happens to have the feature that C++ lacks. So the combination of these two languages proves to be very effective. C++ is used to implement the detailed protocol and OTcl is used for users to control the simulation scenario schedule events.

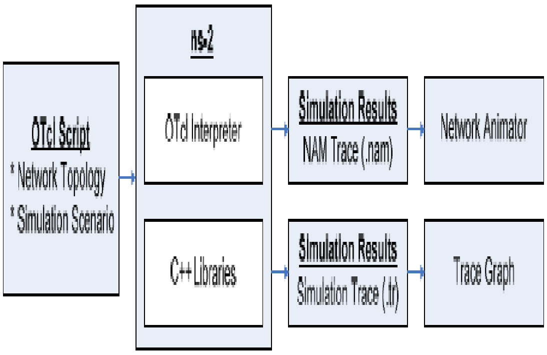


Figure 3 Network Simulator 2

A simplified user's view of NS2 is shown in figure 3. The OTcl script is used to initiate the event scheduler, set up the network topology, and tell traffic source when to start and stop sending packets through event scheduler. The scenes can be changed easily by programming in the OTcl script. When a user wants to make a new network object, he can either write the new object or assemble a compound object from the existing object library, and plumb the data path through the object. This plumbing makes NS2 very powerful. Another feature of NS2 is the event scheduler. In NS2, the event scheduler keeps track of simulation time and release all the events in the event queue by invoking appropriate network components. All the network components use the event scheduler by issuing an event for the packet and waiting for the event to be released before doing further action on the packet.

1) Tcl Language in NS: TCL is a very powerful and dynamic programming language. It has a wide range of usage, including web and desktop applications, networking, administration, testing etc. Tcl is a truly cross platform, easily deployed and highly extensible. The most significant advantage of Tcl language is that it is fully compatible with the C programming language and Tcl libraries can be interoperated directly into C programs. I will describe the Tcl code and designed to implement the black hole attacks.

2) Testing the Black Hole AODV: I have tested my implementation of the Black Hole to see whether it is correctly working or not. To be ensure the implementation is correctly working, I have used the NAM (Network Animator) application of NS. To test the implementation I have used two simulations. In the first scenario i did not use any Black Hole AODV Node (the malicious node that exhibits the Black Hole Attack will be called "Black Hole Node"). In the first scenario explain, simple packets forward between nodes. In the second scenario added a Black Hole AODV Node to the simulation. Then I have compared the results of the simulations using NAM.

IV. Simulation parameters

To take accurate results from the simulations, I used UDP protocol. The source node keeps on sending out UDP packets, even if the malicious node drops them, while the node finishes the connection if it uses TCP protocol. Therefore, I could observe the connection flow between sending node and receiving node during the simulation. Furthermore I was able to count separately the sent and received packets since the UDP connection is not lost during the simulation. If i had used TCP protocol in my scenarios, i could not count the sent or received packets since the node that starts the TCP connection will finish the connection after a while if it has not received the TCP ACK packet. I generate a small size network that has 7 nodes and create a UDP connection between Node 2 and Node 5, and attach CBR (Constant Bit Rate) application that generates constant packets through the UDP connection. CBR packet size is chosen to be 512 bytes long, data rate is set to 1 Mbyte. Duration of the scenarios is 20 seconds and the CBR connections started at time equals to 1.0 seconds and continue until the end of the simulation, in a 79 x 659 meter flat space. I have manually defined appropriate positions of the nodes to show the data flow and also introduce a movement only to Node 1 to show the changes of the data flow in the network. The Tcl script contains a Black Hole AODV for the first simulation.

A) Evaluation of Simulation

In the first scenario where there is not a Black Hole AODV Node, connection between Node 5 and Node 4 is correctly flawed when look at the animation of the simulation, using NAM. Figure 4. A show the data flow from Node 2 to Node 5. When the Node 1 leaves the propagation range of the Node 2 while moving, the new connection is established via Node 3. The new connection path is shown in Figure 4.

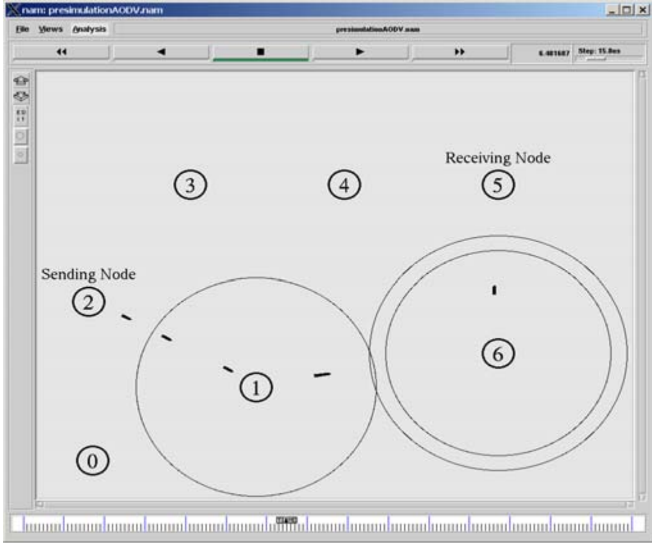


Figure 4 Data flow between Node 2 and Node 5 via Node 1 and Node 6

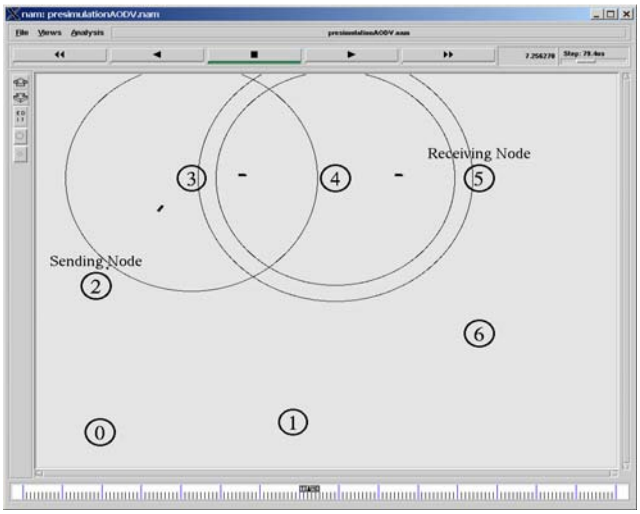


Figure 5 Data flow between Node 2 and Node 5 via Node 3 and Node 4

In the second scenario, commenting out the three statements in the Tcl script, shown in Figure 5, i could easily add the Black Hole behavior to Node 0. The first statement, "$ns node-configure ad-hoc Routing black hole AODV" is to add the Black Hole AODV behavior to the nodes created from this point on. But I have only defined Node 0 as a Black Hole AODV and i have to change to AODV protocol after Node 0 again with the third statement. The second statement just puts a notification to Node 0 defining it as a Black Hole Node. Node 0 being a Black Hole AODV Node absorbs the packets in the connection from Node 2 to Node 5. Figure 6 shows how the Black Hole AODV Node absorbs the traffic.

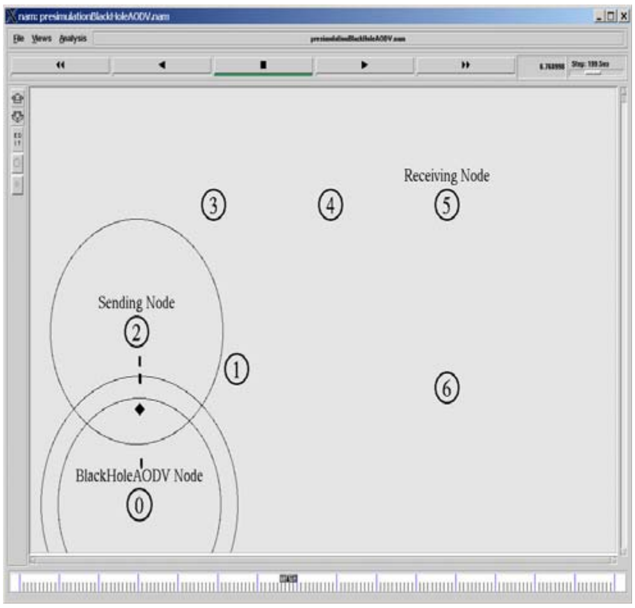


Figure 6 Node 0 (Black Hole Node) absorbs the connection Node 2 to Node 5

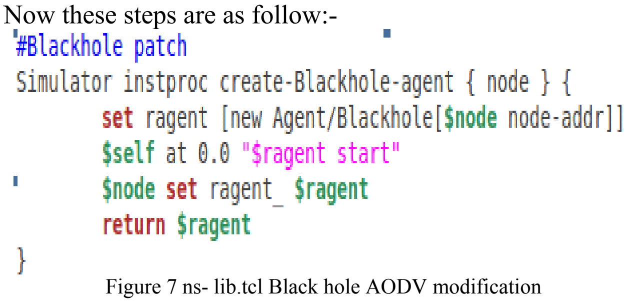
In my test, i ensured that the Black Hole AODV implementation is correctly working. Then, i performed the actual simulation i will describe in the next section. Because i cannot easily see the effects of the Black Hole AODV Node in the large number of Nodes and connections, i will carry out in the actual simulation; i had to test the implementation in a small sized simulation that has a small number of nodes.

B) Simulation of Black Hole Attack& Gray Hole Attack

I have implemented the protocol which will implement Black-hole and gray hole attack in ns2. Now i have to do simulate the scenario to check whether the protocol is working properly or not. To test whether the implementation of GAODV is working correctly or not, i have created a scenario in which 8 routers are connected initially and checked the data traffic when all routers are using the original AODV protocol. After that one of the routers is set to use GAODV protocol and compared the data traffic in both occasions. As expected the delivery ratio of data is decreased when i use GAODV protocol.

1) Steps of modification in objects files: In the first step extract the black hole aodv folder from home folder. In the second step, there are 4 object files are available for eg. Lib.tcl, agent.tcl, make file, adov.cc. now change in the first object file lib.tcl which is available in the ns-2.34. first open the ns-alline-2.34, then open ns-2.34, tcl, lib. In lib folder open the code of object file, then the search code where user want to make changes, copy of this code and make changes in code and give name such as black hole aodv lib.tcl. With the help of these steps user make new files.

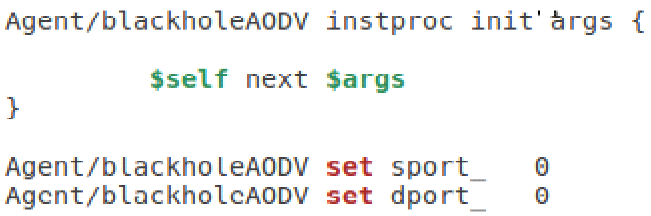
Now these steps are as follow:-



```
#Blackhole patch
Simulator instproc create-Blackhole-agent { node } {
        set ragent [new Agent/Blackhole[$node node-addr]]
        $self at 0.0 "$ragent start"
        $node set ragent_ $ragent
        return $ragent
}
```

Figure 7 ns- lib.tcl Black hole AODV modification

The first modified file is the ns − lib.tcl. It's in this file the protocol agents are coded in a procedure. So her I had to add the protocol agent for the newly created black hole AODV protocol. When a node is using black hole AODV protocol this agent is scheduled at the beginning of the simulation and is assigned to the nodes which use the protocol.



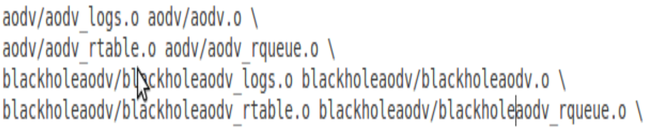
```
Agent/blackholeAODV instproc init'args {

          $self next $args
}

Agent/blackholeAODV set sport_   0
Agcnt/blackholcAODV set dport_   0
```

Figure 8 ns – agent.tcl Black hole AODV modification

The next file to be modified is the ns − agent.tcl. In this I have to set the port numbers for the new routing protocol. S port is the source port and d port is the destination port.



```
aodv/aodv_logs.o aodv/aodv.o \
aodv/aodv_rtable.o aodv/aodv_rqueue.o \
blackholeaodv/blackholeaodv_logs.o blackholeaodv/blackholeaodv.o \
blackholeaodv/blackholeaodv_rtable.o blackholeaodv/blackholeaodv_rqueue.o \
```

Figure 9 Make file.in Black hole AODV modification

The third file modified is the make file:in in the root directory of ns-2.34. This file is modified for creating the object files for the cpp coded files. After all the implementations are ready, we have to recompile NS-2 again to create the object files.

Till now I have implemented a new routing protocol in NS-2 which is labeled as Black hole AODV. But I still didn't implement the black hole attack in this protocol. Now this protocol will act similar to the AODV protocol. To add black hole behavior in to the new protocol i had to make some changes in the Black hole aodv:cc C++ file. By explaining the working mechanism of AODV and Black-hole AODV protocol I will describe the changes made to the Black hole aodv:cc. In aodv:cc code when a packet is received it is received by a function called the recv and the received packets are processed based on the type of the packet. In this code the different control packets in AODV like RREQ, RREP and RERR packets are processed by different functions. The recv function checks whether the received packet belongs to any of these control packets. If it so then it will call the recv AODV function. If the received packet is a data packet, usually

```
//If destination address is its self
if ( (u_int32_t)ih->s addr() == index)
forward((black hole aodv_rt_entry*) 0, p, NO_DELAY);
else
// For black hole attack in the wireless ad-hoc network, after taking the path over itself, misbehaving node drops all packets
drop(p, DROP_RTR_ROUTE_LOOP);
```

Figure 10 Black-hole aodv.cc AODV modification

the AODV protocol will forward the packet to the destination address. But in Black-hole AODV protocol the code is modified such that it will drop all packets without forwarding it. This attack is implemented in the recv function of Black-hole AODV. In black hole attack malicious node drop all the packet when source node send packet to destination. In a black hole attack the malicious node will always advertise in the network that it has a fresher route to the destination by setting the sequence number to a large value and will reply to the RREQ before other routers send a reply. Thus the attacker router will attract all the traffic in its transmission range towards itself and then may drop the packets [12].

C) Assumptions

I assume that all the routers that are in the network are stationary and have no energy constraints. I also assume that the wireless interfaces support promiscuous mode operation. Promiscuous mode means that if a node A is within range of a node B, it can overhear communications to and from B even if those communications do not directly involve A. While promiscuous mode is not appropriate for all wireless mesh network scenarios (particularly some military scenarios) it is useful in other scenarios for improving routing protocol performance. I also assume that the each router is provided with an infinite buffer size so that no packets are dropped because of buffer overflow. In the case of black hole attack I have assume that the attack node will drop all the packets that it receives. Finally I also assume that each mesh router is provided with a private/public key pair and also all the public keys of other routers in the network. These keys are used to protect the packets generated while broadcasting the packet reporting the attack generated by the algorithm.

D) Attack Detection Algorithm

I present an algorithm for finding the intentional Black hole attack by a node and if all the packets are dropped will identify the attack as a black hole attack by checking the forwarding of packets by the immediate neighbor downstream node to which the data is sent. For this i have to overhear the traffic by the neighboring nodes. In my algorithm at each mesh router, the router will maintain a packet count history of the number of packets it has forwarded to the downstream node and also the number of packets it has overheard for the forwarded packets. When a router forwards a packet to the downstream node, the number of packet sent (n t) is incremented and also buffers the packet for a certain time period. Then it overhears the packet which is forwarded by the downstream node and compares with the packet in the buffer. When a match is found the number of packets forwarded by downstream node (no) is increased. Once the match is found or if the time period is over the packet is deleted from the buffer. If the packet forwarding is not heard within the time period the algorithm assumes that the packet is dropped by the downstream node. After sending out a threshold number of packets (n threshold), the number of packets dropped (n d) is calculated and is the difference of the number of packets transmitted to the number of packets overheard.

N d = n t - no

According to these observations each router will maintain a probability value called the Probability of attack (Pa), which is obtained by the number of packets dropped by the downstream node(nd) to the number of packets forwarded by the router to the downstream (nt).

Pa = nd /nt

The obtained probability of attack (Pa) is compared with a threshold value of probability called probability of black hole attack (Pb) and if Pa is greater Pb then a possibility of black hole attack is identified. if Pa > Pb , possibility of a black hole attack. When this condition fails Pa is compared with probability of gray hole attack (Pg), and if Pa is found greater than Pg then a possibility of gray hole attack is identified. if Pa > Pg , possibility of a gray hole attack. If these conditions becomes true twice within the interval an attack is identified. If Pa becomes greater than Pb twice in the interval then a packet is broadcasted to all the routers in the mesh network by the identifying router with the reporter node id, attacker node id and also the type of attack denoted by ’B’. If Pa becomes greater than Pg twice in the interval then a packet is forwarded as before with the type of attack as ’G’. If Pa is greater than Pb and Pg once in the interval then the type of attack is ’G’. At each router they maintain a table called the Attack table. When an attack is reported each router will update its attack table with the reporter node id, attacker node id and also the type of attack. If a router is reporter by two different routers then that is identified as an attack node. In AODV protocol if a node receives a RREQ it will check in the attack table and will not forward the RREQ to a attack router there by isolating them from the network.

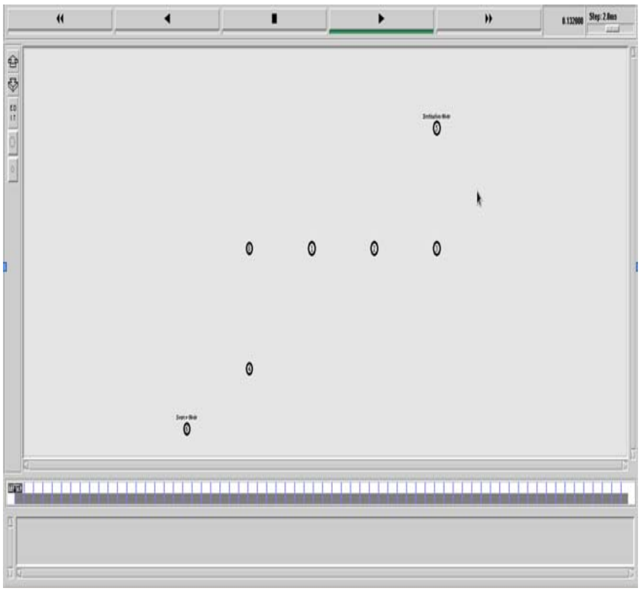


Figure 11 Initial Configuration of nodes in NAM

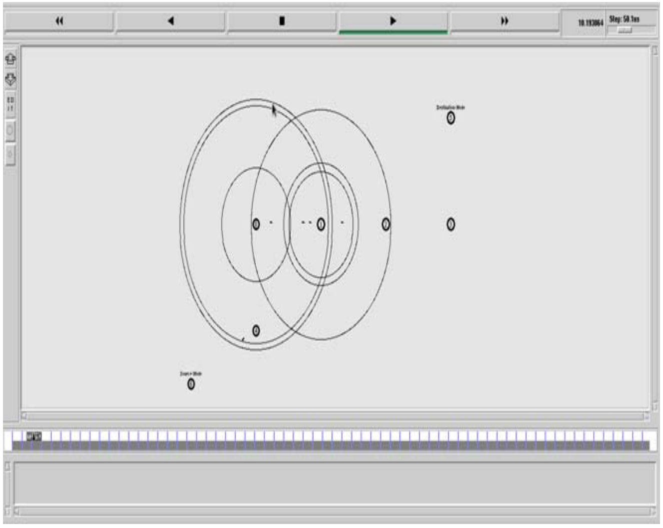


Figure12 Data Transmission source to destination via intermediate nodes

RESULT &GRAPH

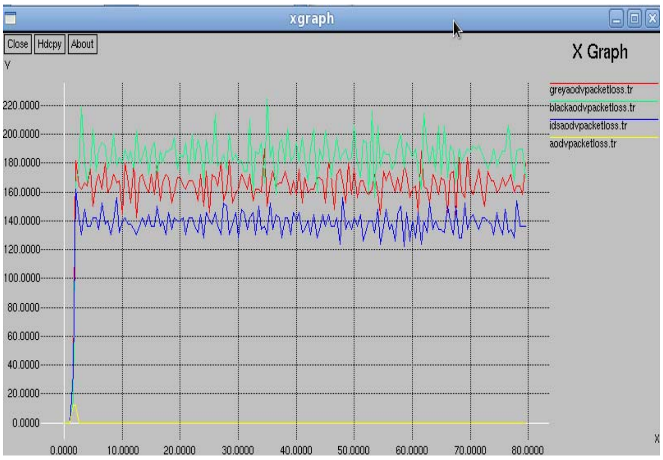


Figure13 :-idsaodvpacketloss.tr

Packet Drop Rate:-Analysis: This figure shows a high packet drop due to Black Hole & Gray Hole attack.

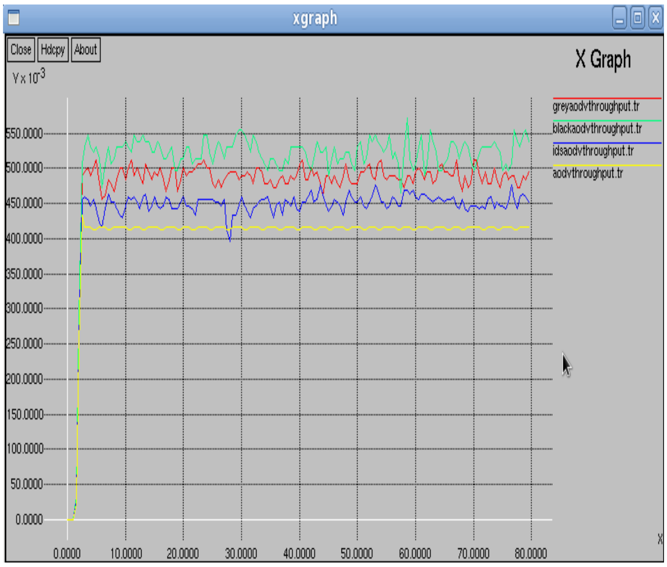


Figure14:- idsaodvthroughput.tr

Throughput:-Analysis Node 1 starts transmitting at time T =1.4 sec [1.4 sec, 10 sec] Node 1 is the only transmitting node using the entire available bandwidth.

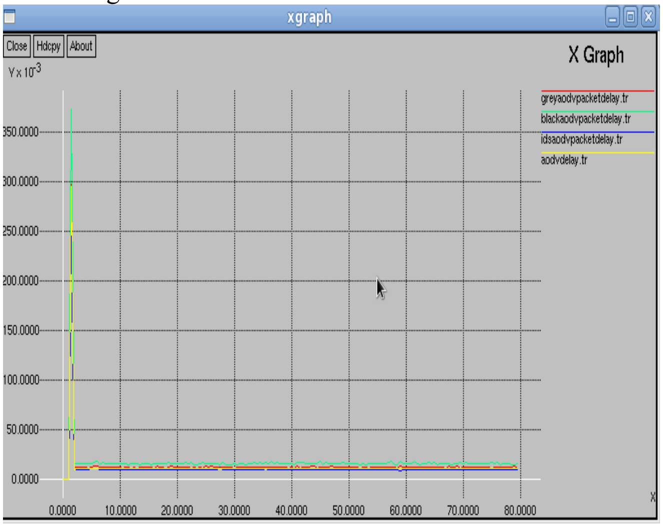


Figure15:- idsaodvpacketdelay.tr

Average Packets End to End Delay:-In my sample n/w there is only single transmitting node so there is no delay in network, hence straight line.

Conclusion And Future Work

In future I have plan to make the threshold values dynamic in the presence of normal loses due to wireless channel and MAC layer collisions and to work on the attacks when the attack routers collude together. Since routers in WMNs work in a fully wireless environment the packet can be lost due to different factors. So finding an appropriate threshold value for detecting the gray hole attack in real environment is really difficult. Wireless mesh networks is having an open architecture and more prone to Denial of Service attacks due to its use in broadband internet access. Thus more research work has to be done to reduce the Denial of Service attacks and improve the network.

References

[1] Shafiullah Khan, Kok-Keong Loo, Tahir Naeem, and Mohammad Abrar Khan."Denial of service attacks and challenges in broadband wireless networks"

[2] I.F. Akyildiz and Xudong Wang" A survey on wireless mesh networks", *Communications Magazine",* IEEE, 43(9):S23 – S30, sept. 2005.

[3] Choong Seon Hong Muhammad Shoaib Siddiqui," Security issues in wireless mesh networks"," *International Conference on Multimedia and Ubiquitos Engineering*", IEEE Computer Society, IEEE, 2007.

[4] S Seth and A. Gankotiya," Denial of service attacks and detection methods in wireless mesh networks","*Recent Trends in Information, Tele-communicatioan Computing (ITC)",2010 International Conference on*, pages 238 –240,march 2010.

[5] M. Arora, R.K. Challa, and D. Bansal," Performance evaluation of routing protocols based on wormhole attack in wireless mesh networks", "*Computer and Network Technology (ICCNT), Second International Conference on*" 2010,pages 102 –104, april.

[6] M. Medadian, M.H. Yektaie, and A.M. Rahmani," Combat with black hole attack in aodv routing protocol in manet", *Internet, 2009. AH-ICI 2009,"First Asian Himalayas International Conference on*", pages 1 –5, nov. 2009.

[7] A. Patcha and A. Mishra," Collaborative security architecture for black hole attack prevention in mobile ad hoc networks","*Radio and Wireless Conference, 2003. RAWCON '03.Proceedings*", aug. 2003 pages 75 – 78.

[8] L. Lazos and M. Krunz,"Selective jamming/dropping insider attacks in wireless mesh networks. *Network",* IEEE, 25(1):30 – 34, January-february 2011.

[9] D.M. Shila and T. Anjali, "Defending selective forwarding attacks in wmns" "*Electro/Information Technology, 2008.EIT 2008. IEEE International Conference on*" may 2008 pages 96 –101.

[10] Guorui Li, Xiangdong Liu, and Cuirong Wang,"A sequential mesh test based selective forwarding attack detection scheme in wireless sensor networks", *Networking, Sensing and Control (ICNSC), 2010 International Conference on*" april 2010 pages 554 –558.

[11] S. Ghannay, S.M. Gammar, F. Filali, and F. Kamoun,"Multi-radio multichannel routing metrics in IEEE 802.11s-based wireless mesh networks and the winner is"," *Communications and Networking, 2009.ComNet 2009. First International Conference on*", pages 1 –8 nov 2009

[12] A. Prathapani, L. Santhanam, and D.P. Agrawal. Intelligent honeypot agent for black hole attack detection in wireless mesh networks. In Mobile Adhoc and Sensor Systems, 2009. MASS '09," *6th International Conference on*, IEEE, pages 753 –758, oct. 2009.

[13] K. Fall and K. Varadhan, NS notes and documentation, The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC, 1997.

[14] Chris Karlof and David Wagner,"Secure routing in wireless sensor networks"," attacks and countermeasures" Ad Hoc Networks, Sensor Network Protocols and Applications,1(2-3):293 – 315, 2003.

[15] Sergio Marti, T. J. Giuli, Kevin Lai and Mary Bakerm," Mitigating routing misbehavior in mobile ad hoc networks","In *Proceedings of the 6th annual international conference on Mobile computing and networking*",MobiCom New York, NY, USA, 2000. ACM.'00, pages 255–265.

[16] D.M. Shila, Yu Cheng, and T. Anjali," Mitigating selective forwarding attacks with a channel-aware approach in wmns, Wireless Communications", IEEE Transactions on 9(5):1661 – 1675, may 2010. [17] D.M. Shila, Yu Cheng, and T. Anjali",Channel-aware detection of gray hole attacks in wireless mesh networks. In "*Global Telecommunications Conference"*IEEE *2009.* GLOBECOM 2009. IEEE, pages 1 –6, 30 2009-dec. 4 2009.