International Journal of Computer Applications Technology and Research
Volume 6–Issue 5, 224-228, 2017, ISSN:-2319–8656

Optimised Proactive Link State Routing For DOS Attack Prevention

Vishnu S Kumar
Department of Computer Science and Engg.
Mangalam College of Engineering
Ettumanoor, Kerala, India

Divya. S. B
Department of Computer Science and Engg.
Mangalam College of Engineering
Ettumanoor, Kerala, India

Abstract: A Mobile Ad hoc Network is a collection of independent mobile nodes that can communicate to each other via radio waves. The mobile nodes that are in radio range of each other can directly communicate, whereas others need the aid of intermediate nodes to route their packets. Each node has a wireless interface to communicate with each other. These networks are fully distributed, and can work at any place without the help of any fixed infrastructure as access points or base stations. Routing protocols are divided into two broad classes – Reactive and Proactive. In Reactive or on demand routing protocols the routes are created only when they are needed. The application of this protocol can be seen in the Dynamic Source Routing Protocol (DSR) and the Ad-hoc On-demand Distance Vector Routing Protocol (AODV). Wherein Proactive or Table-driven routing protocols the nodes keep updating their routing tables by periodical messages. OPSR proposes a proactive mechanism in source routing.

Keywords: MANET, OPSR, DOS attack

1. INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a group of mobile devices capable of communicating wirelessly with each other without using a predefined infrastructure or centralized authority [1]. Sending packets from one node to another is done through a chain of intermediate nodes. A number of routing algorithms exist for packet transmission in networks. These algorithms can be broadly classified into two main categories: reactive routing and proactive routing protocols. In the case of proactive (table-driven) protocol, for example, DSDV[2] and OLSR [3], [4], every node constantly maintains a list of all possible destinations in the network and the optimal paths routing to it. Reactive protocols, such as DSR [5] and AODV [6], find a route only on demand.

The essential requirement of MANET's is its ability to have all its nodes recognized by other node in the network, even in motion. A route between two nodes can be broken due to intermediate nodes that dynamically change their position. Mobile nodes can join or leave the network at any time.

The Optimized Link State Routing (OLSR) protocol [3], [4], has become one of the algorithms widely used today [7]. Although OLSR is quite efficient in bandwidth utilization and in path calculation, it is vulnerable to various attacks [8], [9]. As OLSR relies on the cooperation between network nodes, it is susceptible to a few malicious nodes which can cause routing havoc. These attacks include link withholding attacks [6], link spoofing attacks [6], flooding attacks [6], wormhole attacks, replay attacks, black-hole attacks, colluding mis-relay attacks, and DOS attacks.

Denial-of-service attack (DoS attack) is a cyber-attack where the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled. Denial-of-service attacks are characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. The nodes causing denial of service attacks are mostly selfish nodes .

There can be two types of selfish attacks –selfish node attack (saving own resources) and sleep deprivation (exhaust other's resources). Routing protocol plays a crucial role for effective communication between mobile nodes and operates on the basic assumption that nodes are fully cooperative. A selfish node does not supposed to directly attack the other nodes, but is unwilling to spend battery life, CPU cycles, or available network bandwidth to forward packets not of direct interest to it. It expects other nodes to forward packets on its behalf. To save own resources there is a strong motivation for a node to deny packet forwarding to others, while at the same time using the services of other nodes to deliver own data.

At first in Route Update, each node in the network constructed a star graph centered at that node itself. i.e., at the beginning, a node is only aware of the existence of itself. In our proposed model we create selfish node who drops the the packet to next intermediate hop to reach its destination. Normal routing protocols does not detect this threat. But here we form an adjacency matrix of each node based on the network constructed for each node after that we form a spanning tree for each node to find the number of intermediate nodes, as the selfish nodes coursing DOS attack will not be having next intermediate hops their calculated values will be zero and the non attacker nodes will be having values greater than zero based upon their intermediate next hops count. This phase is done at the routing level, so before forming the routing paths the identified selfish nodes are eliminated from routing table and form proactive routes based on this.

The reminder of this paper is organized as follows. In Section 3 the protocols such s ADOV, AOMDV, OLSR, DSR, protocols are presented. A method for protecting OLSR MANET from DOS attack is described in depth in Section 4. Section 5 and describes the simulation model and presents the results achieved along with a discussion of the results. Finally, conclusions and future works are presented in Section**.**

2. BACKGROUND

Network Simulator (Version 2), widely known as NS2, is simply an event-driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols can be done using NS2. In general, NS2 provides users with a way of specifying network protocols and simulating their corresponding behaviors.

Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events.

3. ROUTING PROTOCOLS IN NS2

3.1 Destination-Sequenced Distance-Vector

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements[2]. Every mobile station maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven.

3.2 Ad Hoc On-Demand Distance Vector Routing

AODV discovers routes on an as needed basis via a similar route discovery process[5]. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate an RREP back to the source and, subsequently, to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine freshness of routing information and to prevent routing loops. All routing packets carry these sequence numbers. An important feature of AODV is the maintenance of timer-based states in each node, regarding utilization of individual routing table entries. A routing table entry is expired if not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes which use that entry to route data packets.

3.3 Dynamic Source Routing (DSR)

The key distinguishing feature of DSR is the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process to dynamically determine such a route. Route discovery works by flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path traversed across the network.

3.4 AOMDV Protocol

AOMDV stands for Ad-hoc On-demand Multipath Distance Vector Routing protocol. AOMDV is a multipath extension to the AODV protocol[10]. In AOMDV protocols multiple routes are founded between the source and destination.It uses alternate routes on a route failure. In AOMDV protocols new route discovery is needed when all the routes fail. In AOMDV protocols multipath routing is the enhancement of unipath routing in which advantage is to handle the load in network and avoid the possibility of congestion and increases reliability.

3.5 OLSR PROTOCOL

OLSR is a proactive routing protocol, that is, it is based on periodic exchange of topology information. The key concept of OLSR is the use of multipoint relay (MPR) to provide an efficient flooding mechanism by reducing the number of transmissions required. In OLSR, each node selects its own MPR from its neighbors. Each MPR node maintains the list of nodes that were selected as an MPR; this list is called an MPR selector list. Only nodes selected as MPR nodes are responsible for advertising, as well as forwarding an MPR selector list advertised by other MPRs.

4. OPTIMISED PROACTIVE LINK STATE ROUTING

OPSR proposes a proactive mechanism in source routing. Our proposed method, provides every node with a Breadth First Spanning Tree (BFST) of the entire network rooted at itself. To do that, nodes periodically broadcast the tree structure to its best knowledge in each iteration. Based on the information collected from neighbors during the most recent iteration, a node can expand and refresh its knowledge about the network topology by constructing a deeper and more recent BFST. This knowledge will be distributed to its neighbors in the next round of operation. On the other hand, when a neighbor is deemed lost, a procedure is triggered to remove its relevant information from the topology repository maintained by the detecting node.

With the adjacency matrix calculation and spanning tree we find out the nodes with zero adjacency that is nodes with no forwarding node or intermediate hopes. Attacker nodes will be off no intermediate nodes as they drop the received packets or increases the path length by wasting the bandwidth. After identifying these nodes it will not be considered for routing in our proposed method thus by ensuring a much better safer and less overhead communication.

5. SIMULATION PLATFORM CREATION

For the simulation of nodes in mobile adhoc network (MANET), we have created the platform on Ubuntu. The MANET network simulations are implemented using NS-2 simulator. For this purpose, in NS2 we need to create a topology for the project with which can be used for proactive source routing. The coding will be done using TCL (Tool Command Language). But none of current NS2 versions does not have any proactive source routing mechanism. Source routing included in NS2 is DSR.

For analysis of existing source routing we need to integrate OLSR routing protocol in NS2 which is not part of standard NS2. And it is available as patch file externally. But to integrate this OLSR into NS2 will include some work as it will now compile with the current NS versions. This is done to generate olsr object file with the GCC compiler. NS2 version here we used is NS ALL in one 2.35.

The topology creation will be done using TCL coding. But to edit AODV or DSR or to create a new protocol we cannot code with TCL. Protocol codes are core coded files which is done using C++. So in coding, first thing needs to do the topology and node creations using TCL which uses existing protocol coding within NS all in one version 2.35.

For analyzing the delay, throughput and overhead caused in the existing method we need to capture the packet drop and through put, for this we generate the trace output files of out TCL execution. From this trace output we calculate the drop and throughput using Perl and AWK scripts.

For next purpose we need to find the core code files(written in C++) related to our project in NS. We need to create a new proactive source routing cpp code along with its associate routing and header files, as there is no other proactive source routing code to modify in current NS versions we need create it a whole new one for this. Gcc Complier will be called to compile the new coding and and then will be futher bind with the TCL . This will enable TCL to call the newly created protocol code into topology. And further we can compare delay, throughput and overhead caused of the new PSR with the exixting Protocols including the newly added OLSR.

6. PERFORMANCE EVALUATION AND RESULTS

Here we present the measurement of various parameters by implementing the simulation environment. Throughput is defined as the ratio of the data delivered to the destination of the data sent out by the sources[7]. Average end-to-end delay is the avg. time a packet takes to reach its destination.

End-to-End Delay (EED): It is the time taken for an entire message to completely arrive at the destination from the source. Evaluation of end-to-end delay mostly depends on the following components i.e. propagation time (PT), transmission time (TT), queuing time (QT) and processing delay (PD). Therefore, EED is evaluated as:

$$EED = PT + TT + QT + PD.$$

Throughput: It is the measure of how fast a node can actually sent the data through a network. So throughput is the average rate of successful message delivery over a communication channel.

Packet Sent and Received: It is the total number of packets sent and received during the complete simulation timeframe.

Packet Delivery Ratio (PDR): It is the ratio of the total data bits received to total data bits sent from source to destination.

Control Overhead: It is ratio of the control information sent to the actual data received at each node.

6.1 RESULTS AND ANALYSIS

During the implementation of this project, an attempt was made to compare the performances of various protocols such as AODV, AOMDV, OLSR and PSR under the same simulation environment.

For all the simulations, the same movement models were used, the packet size is fixed to 512 bytes. For the experimental significance, here we only discuss the experimental results of simulation of 6 nodes only. The simulations environment is the same for other nodes of 10,15,20 number of nodes. The diversity of the experiments is more as we increase the number of nodes in a simulation environment.

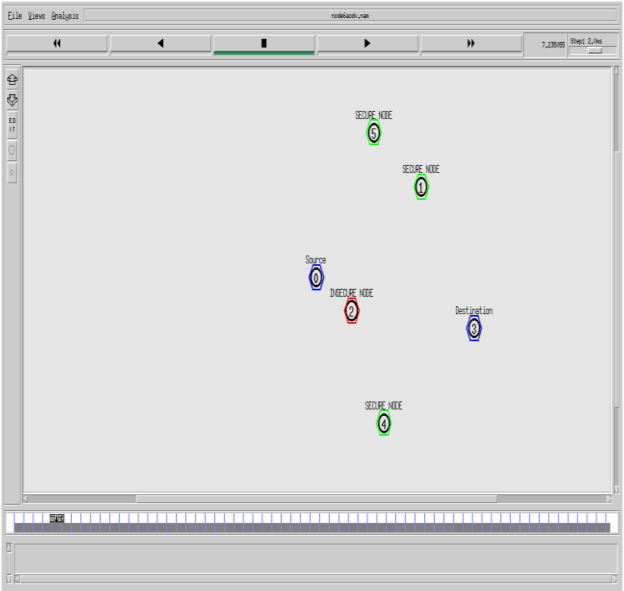

Figure 1: Simulation with 5 nodes

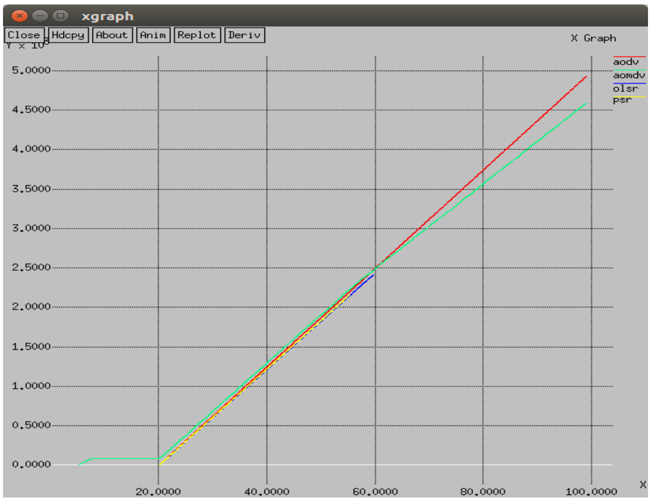

Figure 2: Number of dropped packets

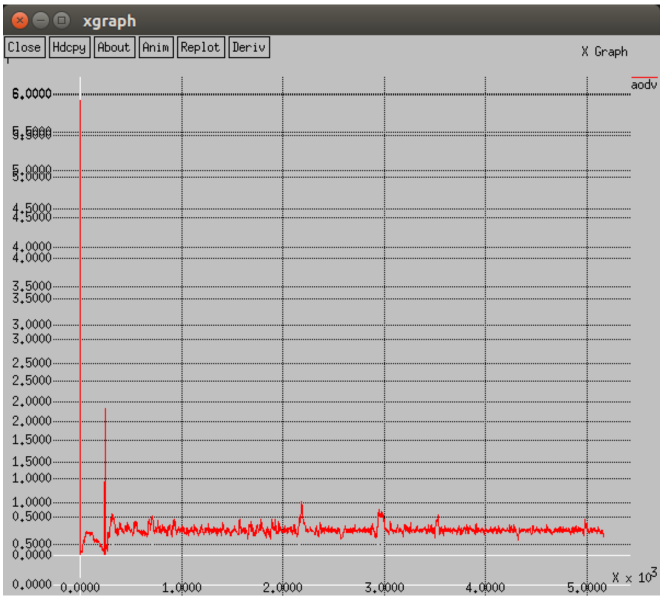

Figure 3: End-to-End Delay in AODV

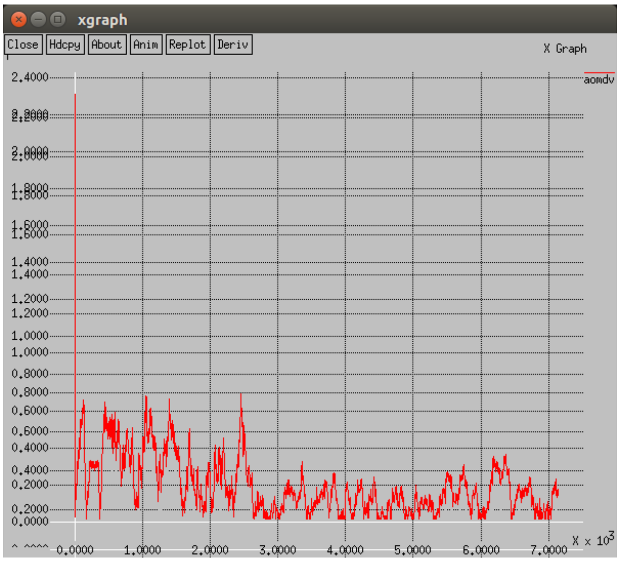

Figure 4: End-to-End Delay in AOMDV

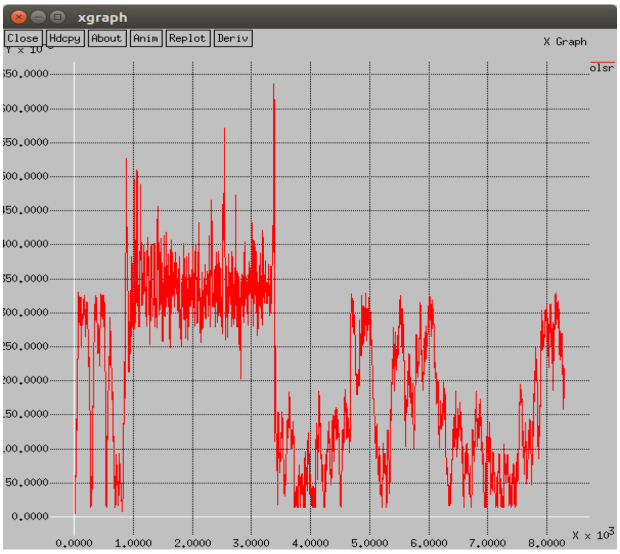

Figure 5: End-to-End Delay in OLSR

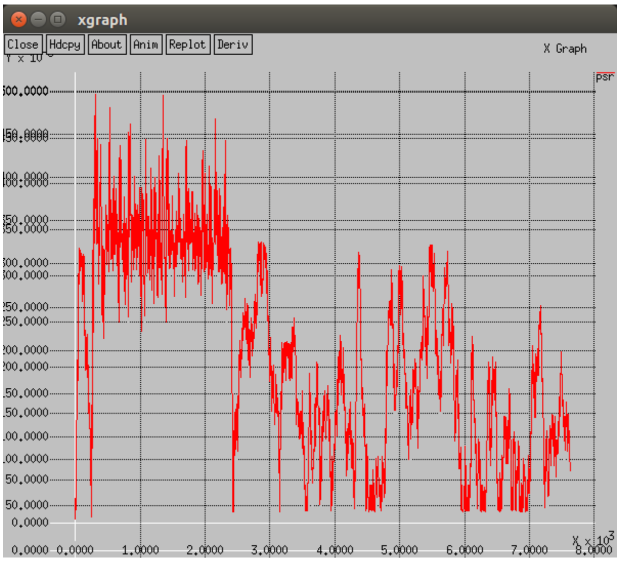

Figure 6: End-to-End Delay in OPSR

7. CONCLUSION

In this project, we evaluated the five performance measurements of various routing protocols such as AODV, AOMDV, OLSR and PSR. Routing protocols were simulated with 6,10, and 15 nodes moving randomly. In this project proposed a new routing protocol called OPSR, a secure extension for source routing protocol in Mobile Ad hoc Networks. Reviewed different routing protocols: Reactive and Proactive. Reactive protocols are on demand protocols. These Protocols do not initiate route discovery by themselves, until or unless a source node request to find a route. The major drawback of this protocol is that its initial delay in path establishment is high.

Proactive protocols are table driven which maintain up-to-date information of routes from each node to every other node in the network. These protocols continuously learn the topology of the network by exchanging topological information among the network nodes. Thus, when there is a need for a route to a destination, such route information is available immediately. Drawback of this protocol is that overhead because every node keep all possible path to every other node in the network. OPSR is introduced to overcome the drawback of reactive and proactive protocols. OPSR design includes three phases: Route Update, Neighbourhood Trimming, and node Update. In the simulation part compared the performance of OPSR with existing protocols such as AODV, DSDV, DSR and OLSR and results are analysed. Proposed model of OPSR reduces overhead and initial delay in route finding and to detect and prevent blackhole attacks in MANETs.

In Future works and development we can add cross layer security to futher improve the security under an attack. And further more parameters like range , bandwidth , assigning trustworthy values by neighboring(which has routing overhead delays and pother drawbacks) in improved ways to enhance our proposed method OPSR .

8. REFERENCES

[1] Nadav Schweitzer, Ariel Stulman, Asaf Shabtai, and Roy David Margalit "Mitigating Denial of Service Attacks in OLSR Protocol Using Fictitious Nodes" IEEE Transactions On Mobile Computing, Vol. 15, No. 1, January 2016

[2] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (dsdv) for mobile computers," in Proc. Conf. Commun. Archit., Protocols Appl., 1994, pp. 234–244.

[3] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in Proc. IEEE Int. Multi Topic Conf. Technol., 2001, pp. 62–68.

[4] T. Clausen and P. Jacquet, "RFC 3626-Optimized Link State Routing Protocol (OLSR)," p. 75, 2003. [Online]. Available:http://www.ietf.org/rfc/rfc3626.txt

[5] C. Perkins and E. Royer "Ad-hoc on-demand distance vector routing," in Proc. 2nd IEEE Workshop Mobile Comput. Syst. Appl., Feb. 1999, pp. 90–100.

[6] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," IEEE Wireless Commun., vol. 14, no. 5, pp. 85–91, Oct. 2007.

[7] Samyak Shah, Amit Khandre, Mahesh Shirole and Girish Bhole, "Performance Evaluation of Ad Hoc Routing Protocols Using NS2 Simulation" Mobile and Pervasive Computing (CoMPC–2008).

[8] Mahesh K. Marina, Samir R. Das "On-Demand Multipath Distance Vector Routing in Adhoc Networks" , 1092-1658/01 $17.00 2001 IEEE.

[9] Ankur Sharma1, Er. Rakesh Kumar, "Performance Measurement and Analysis of OLSR Routing Protocol Based On Node Scenarios Using NS2 Simulator" International Journal of Engineering Research and Applications (IJERA) ISSN:2248-9622 Vol. 3, Issue 4, Jul-Aug 2013, pp.1067-1073.

[10] Preeti Aggarwal, Er. Pranab Garg, "AOMDV Protocols in MANETS :A Review", International Journal of Advanced Research in Computer Science & Technology (IJARCST 2016) 32 Vol. 4, Issue 2 (Apr. - Jun. 2016)