An Approach on the Distributed Detection Algorithm for Wormholes in Wireless Network Systems

***P. Ashmy*[1] *and S. Devendran*[2]**

[1]*PG Scholar, Department of Electronics and Communication Engineering, Chandy College of Engineering, Tuticorin, TamilNadu, India*

[2] *Assistant Professor, Department of Electronics and Communication Engineering, Chandy College of Engineering, Tuticorin, TamilNadu, India*

---***---

Abstract - *Wireless sensor network are vulnerable to various type of attacks and hence security is a necessary in mobile ad-hoc network to provide protected communication between mobile nodes. Wormhole is one type of routing attack where malicious nodes promote as it has shortest straight path to all nodes in the environment by conveying fake route reply and their by destabilize wireless sensor networks. In a distinctive wormhole attack, the attacker obtain packets at one point in the network, forwards them through a wired or wireless link with less latency. In this paper, the techniques dealing with wormhole attack are investigated and an approach for wormhole prevention is proposed. We used four different metrics like DAWN, AODV, LEECH, DSDV were used in the evaluation of the Sequence Number Attack Detection and countermeasure mechanism. Among the four metrics the DAWN protocol has improved on many fronts. Our method has been shown to detect local as well as distributed attacks. This depicts a distributed wormhole detection algorithm for wireless sensor networks and main advantage of the algorithm is that it can provide the approximate location of wormholes, which is useful in implementing countermeasures. Simulation results reveals that DAWN has algorithm has low false toleration and detection rates.*

Key Words: Wireless sensor networks, wormhole detection, distributed algorithm, Sensor Nodes, DAWN

1.INTRODUCTION

Wireless sensor network (WSN) is a spatially dispersed autonomous sensor helps to monitor physical or environmental conditions and to helpfully pass their data through the network to a main location. (Fig. 1]. The networks are bi-directional facilitating the control of sensor activity and are initially used for the military applications later it has many industrial and consumer applications [1].

Wormhole attack is one of the Denial-of-Service attacks that can affect network routing and it may be begin by a single or a pair of collaborating nodes. It is generally found two ended wormhole, one end overhears the packets and forwards them through the tunnel to the other end, where the packets are replayed to local area. Wormhole attack does not require MAC protocol information and it is immune to cryptographic technique [2]. This makes it very difficult to detect.

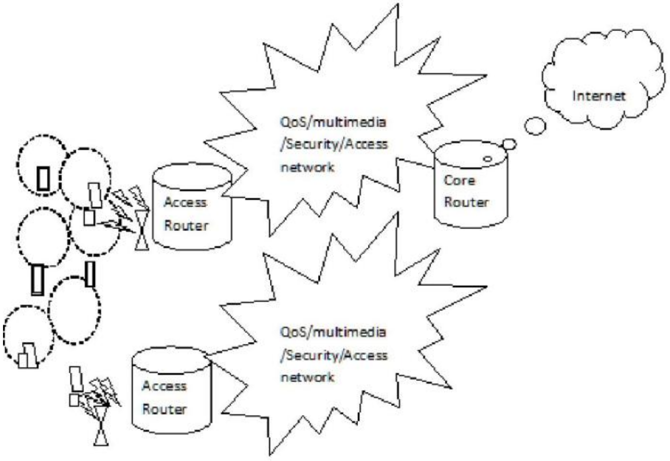


Fig 1: Mobile Ad hoc Networks [3]

Hence to define challenging attacks and to defend against, which we call number of metrics. Here, we present a new, general mechanism for detecting and, thus defending against wormhole attacks. In this attack, an attacker records individual bits from a packet, at one location in the network, tunnels the packet (possibly selectively) to another location, and replays it there. Moreover, four metrics (DAWN, AODV, LEECH, DSDV) were used in the evaluation of the sequence number attack detection and countermeasure mechanism like delivery ratio, the number of false routing packets sent by the attacker, false positive We also show that efficiency of DAWN protocol with other metrics.

2. PROBLEM STATEMENT

2.1 Denial of Service Attack

A Denial of Service (DoS) attack is an attempt to make a computer system (server or client) or some other resource unavailable to legitimate users. In general, it aims to prevent some services from functioning efficiently either temporarily or indefinitely. Hardware failures, environmental conditions, software bugs or resource exhaustion can lead to Denial of Service attacks. Various types of DoS attacks work at different layer and affect differently to the network, where one kind of DoS attack is wormhole attack [3].

2.1 Description of wormhole

An adversary establishes wormhole link between two points in the network. This link can be established via a wireline, a long-range wireless transmission, or an optical link. Once the wormhole link is operational, the adversary eavesdrop messages at one end, referred as the origin point, tunnels them through the wormhole link and replays them in a timely fashion at the other end, referred as the destination point (Fig. 2.)

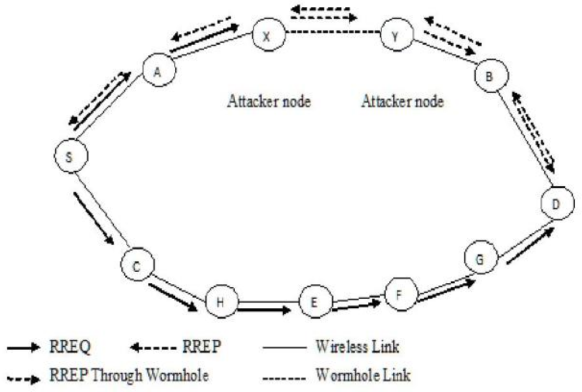


Fig. 2: Wormhole Attack [3]

In a wormhole attack, an attacker receives packets at one point in the network, "tunnels" them to another point in the network, and then replays them in other network from that point. For tunneled distances longer than the normal wireless transmission range of a single hop, it is simple for the attacker to make the tunneled packet arrive with better metric has a normal multihop route, for example, through use of a single long-range directional wireless link or through a direct wired link to a colluding attacker. It is also possible for the attacker to forward each bit over the wormhole directly, without waiting for an entire packet to be received before beginning to tunnel the bits of the packet, in order to minimize delay introduced by the wormhole. Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to itself, since it can over hear them in wireless transmission and tunnel them to the colluding attacker at the opposite end of the wormhole.

3. RELATED WORK

Hu and Perrig [4] presented an approach using Packet Leashes, where in geographic leash and temporal leash put upper bound on location of the receiver and maximum time a packet takes to travel respectively. TIK protocol is proposed for defense against temporal leash, but the knowledge of geographic location or tight time synchronization is required.

Thaier et al., [5] used leashes approach with modified packet transmission methodology to decrease calculation overhead.

In transmission time based mechanism (TTM), Tran, Hung and Lee brothers [6] proposed an approach where each node on path notes time of sending RREQ packet and receiving RREP packet. Here, also time consideration is the main factor.

Hu and Evans [7] proposed a location based approach, where directional antenna is used to check the validity of neighbor. Considering the direction from which the response of HELLO message comes and using verifiers, the neighbors are authenticated. The approach can detect insider attack also by establishing authentication with pair wise secret keys, but hardware support is required here. Additionally, only types of wormholes with fake neighbors can be detected with this methodology.

Khalil, Bagchi and Shroff [8] proposed a lightweight countermeasure (LITEWORP) for wormhole attack detection using guard nodes. After detecting wormhole, LITEWORP leaves network in that open mode only, causing possibility of more disruption.

Chen, Lou, Sun and Wang [9] presented a secure localization approach that can detect simplex and duplex wormhole attacks.

Nait-Abdesselam, Bensaou and Taleb [10] proposed detection and avoidance method that focuses on the load carrying by various routes. When a route is loaded heavily, it may be because of packet congestion etc., so it may signal alarm even when wormhole is not present.

4. TECHNICAL PRELIMINARIES

In this section, we describe the technical preliminaries needed in this paper.

4.1. Wormhole Attack Model

Wormhole attack is one of the Denial-of-Service attacks that can affect the network even without the knowledge of cryptographic techniques implemented. This is the reason why it is very difficult to detect. It may be launched by one, two or more number of nodes. In two ended wormhole, packets are tunnelled through wormhole link from source to destination node. On receiving packets, destination node replays them to the other end.

Designing prevention and detection methods of Wormhole attack requires the classification of Wormhole attacks. Depending on whether the attackers are visible on the route, packet forwarding behaviour of wormhole nodes as well as their tendency to hide or show the identities, wormholes is classified into three types: closed, half open, and open. In the following cases S and D are the source and destination nodes respectively. Nodes M1 and M2 are malicious nodes (Fig. 3)

Open Wormhole

Source(S) and destination (D) nodes and wormhole ends M1 and M2 are visible. Nodes A and B on the traversed path are kept hidden. In this mode, the attackers include themselves in the packet header following the route discovery procedure. Nodes in network are aware about the presence of malicious nodes on the path but they would imitate that the malicious nodes are direct neighbours.

Half-Open Wormhole

Malicious node M1 near the source (S) is visible, while second end M2 is set hidden. This leads to path S-M1-D for the packets sent by S for D. The attackers do not modify the content of the packet. Instead, they simply tunnel the packet form one side of wormhole to another side and it rebroadcasts the packet

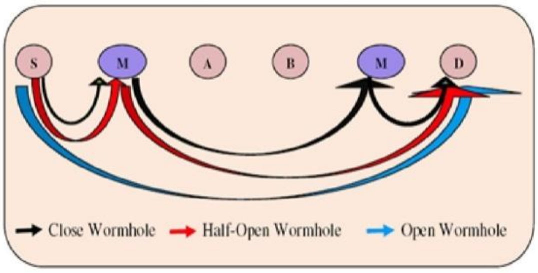


Fig. 3: Representation of Open, Half-Open and Closed Wormhole

Close Wormhole

Identities of all the intermediate nodes (M1, A, B, M2) on path from S to D are kept hidden. In this scenario both source and destination feel themselves just one-hop away from each other. Thus fake neighbours are created.

4.2 Simulation Environment

Operating System	:	Feroda 13 (Linux)
Scripting Language	:	Object Oriented tool command language (OTCL)
NS2	:	Network Simulator Software
NAM	:	Network Animator

For the simulations, we use NS-2 (v-2.34) network simulator. NS-2 provides useful implementations of the different network protocols. The implementation of the protocol has been done using C++ language in the backend and tcl language in the frontend on the Ferado 13 (Linux)operating system. At the physical and data link layer, we used the IEEE 802.11 algorithm. The channel used is Wireless Channel with Two Ray Ground radio propagation model. At the network layer, we use AODV as the routing algorithm. Finally, UDP is used at the transport layer. All the data packets are CBR (continuous bit rate) packets. The size of the packet is 512 bytes. The packets transmission rate is 1 Mbps. The connection pattern is generated using cbrgen and the mobility model is generated using setdest utility. Setdest generates random positions of the nodes in the network with specified mobility and pause time. The terrain area is 750m X 750m with 20 nodes with chosen maximum speed 20 m/s. The duration of time is 500sec.

4.3 Module Implementation

NS (version 2) is an object-oriented, discrete event driven network developed at DC Berkley written in OTCL. NS is primarily useful for simulating local and wide area networks. The purpose is to give a new user some basic idea of how the simulator works, how to create new network components, etc., mainly by giving simple examples and brief explanations based on our experiences. NS is an event driven network simulator developed at UC Berkley that simulates variety of IP networks. It implements network protocols such as TCP and DDP, traffic source behavior such as FTP, Tclnet, web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms.

NS also implements multicasting and some of the MAC layer protocols for LAN simulations. The NS is now a part of the VINT Project that develops tools for simulation results display, analysis and converters that convert network topology generated by well-Known generators to NS formats. Currently, NS (version 2) written in OTCL (TCL script language with object-oriented extensions developed at MIT) is available. This document explains briefly about basic structure of NS and explains in detail how to use NS mostly by giving examples. As shown in figure, in a simplified user's view, NS is object-oriented TCL (OTCL) script interpreter that has a simulation event scheduler and network component, and object libraries, and network setup (plumbing) module libraries (actually, plumbing modules are implemented as member function of the base simulator object).

In other words, to use NS, you program in OTCL script language. To setup and run a simulation network, a user should write a OTCL script that initiates an event schedule, setups the network topology using the network objects and plumbing functions in the library, and Tcls traffic sources when to start and stop transmitting packets to the event scheduler. Another major component of NS beside network objects is the event scheduler. An event in NS is a packet ID that is unique for a packet with scheduled timing and the pointer to an object that handles the event. Timers use event schedulers in a similar manner that delay does. The only difference is that timer measures a time value associated with the packet and does an appropriate action related to the packet after a certain time goes by, and does not simulate a delay.

NS is return not only in OTCL but in C++ also. For efficiency reason NS separates the data path implementation from control path implementation. In order to reduce packet and event processing time (not simulation time), the event scheduler and the basic network component objects in the data path are written and compiled using c++. In this way, the controls of the c++ object are given to Otcl, It is also possible to add member functions and variables to a c++ linked OTcl object

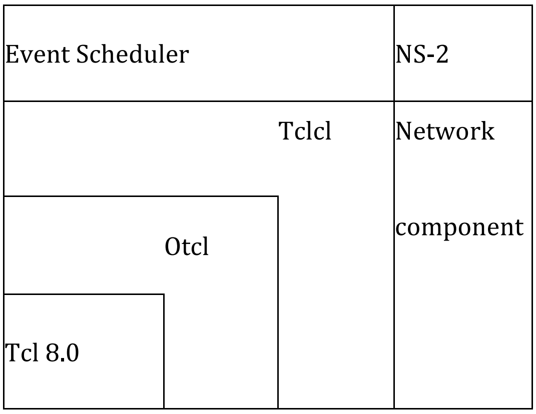


Fig. 4: Architecture view of NS-2

In this figure 4 a general user (not an NS developer) can be thought of standing at the left bottom corner, designing and running simulation in TCL using the simulator objects in the OTCL library. The event scheduler and most of the network components are implemented in c++ and available to OTCL through an OTCL linkage that is implemented using tclcl. The whole thing together makes NS, which is a 00 extended TCL interpreter with the network simulator libraries. The data can be used for simulation analysis or as an input to a graphical simulation display tool called Network Animator (NAM). NAM has a nice graphical user interface similar to that of a CD player (play, fast forward, rewind, pause and so on), and also has a display speed controller.

Furthermore it can graphically present information such as through put and number of packets drops at each link, although the graphical information can't be used for accurate simulation and analysis.

Table 1: Performance comparison is made on the basis of above four metrics (DAWN, LEECH, AODV, DSDV) at 15 nodes and DAWN at 5, 10, 20 and 30 nodes

Simulator	NS-2 (Version 2.34)
Simulation Time	500 (s)
Number of Nodes	15 Nodes (DAWN, LEECH, AODV, DSDV) 5, 10, 20 and 30 nodes (DAWN)
Simulation Area	1000 x 1000 m
Routing protocol	DAWN, LEECH, AODV, DSDV
Traffic	Constant Bit Rate
Pause Time	10 (m/s)
Maximum speed	20 (m/s)

4.3 Metrics used for simulation

To analyze the performance of our solution, various contexts are created by varying the number of nodes and node mobility. In these simulations we used evaluation metrics

4.4 Algorithm for distributed detection algorithm for wormholes in wireless network coding systems (DAWN)

Algorithm

```
DAWN on Node u
Input: R: the set of reports received in the last batch;
       N(u):the set of u's neighbors; sj :the local observation
       result of each neighbor j £N(U);₰:the threshold
output: Detected wormhole attackers in N(U);if any
1.for Each report  r(i,j,k) £ R do
2.     if ETX(j)-ETX(i)≤₰ OR i€ N(J) then{
3.        Discard this report;
4.  else
5.      if j€ N(u) then
6           sj←sj†1;
7.     end if
8     if k<2 then
9          Forward this report r(I,j,k†1);
10          end if
11.   end if
12 .end for
13.for each v€ N(u) do
14              Let C(v)={i\i€ N(v) s.t.ETX(v)-ETX(i) ₰
15             if sv ≥[|c(v)+1/2 ]then
16. Mark v as detected wormhole attacker, and block any
traffic from or to node v in future batches.
17.     end if
18 .end for
```

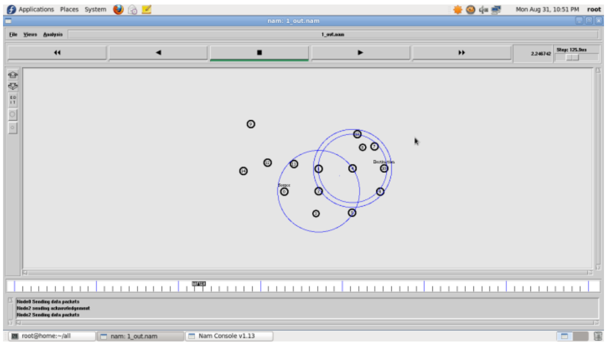
The Distributed Detection Algorithm for Wormholes in Wireless Network Coding Systems proposed is shown in the figure 4.

Fig. 4: Analyze DAWN Message

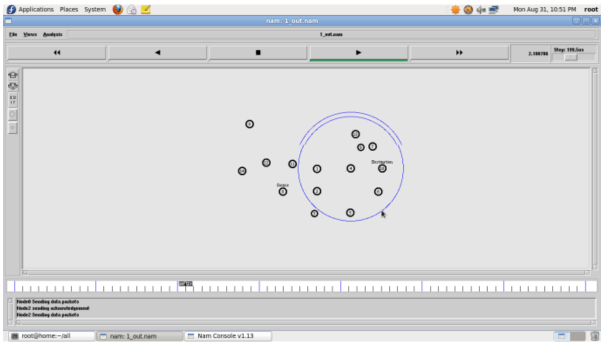
5. RESULTS AND DISCUSSION

5.1 Simulation Results

The arrangement of the 15 nodes is shown in figure 5. The nodes are arranged in a flat grid manner. In figure 5 a large number of packets are flooded into the network by a malicious node. Figure shows the flow of data from a malicious node to the other node in the network. Initially the node broadcast the route request message in the network.



```
Node0 Sending data packets
Node2 sending acknowledgement
Node2 Sending data packets
```

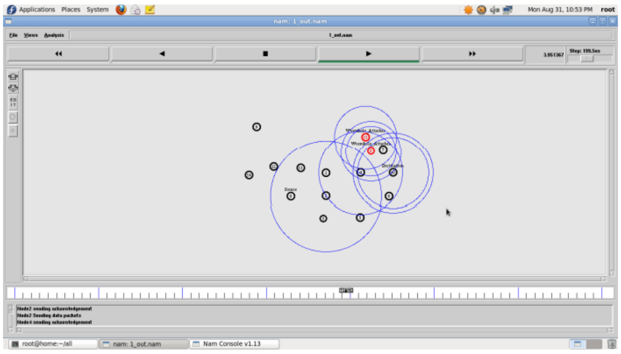


```
Node0 Sending data packets
Node2 sending acknowledgement
Node2 Sending data packets
```

Fig. 5: Source and destination node

Fig 6: Selecting first intermediate node

The figure 6 shows that the node again and again forwards the request message and thus attacked on the network consuming network bandwidth, battery power.



```
Node2 sending acknowledgement
Node2 Sending data packets
Node4 sending acknowledgement
```

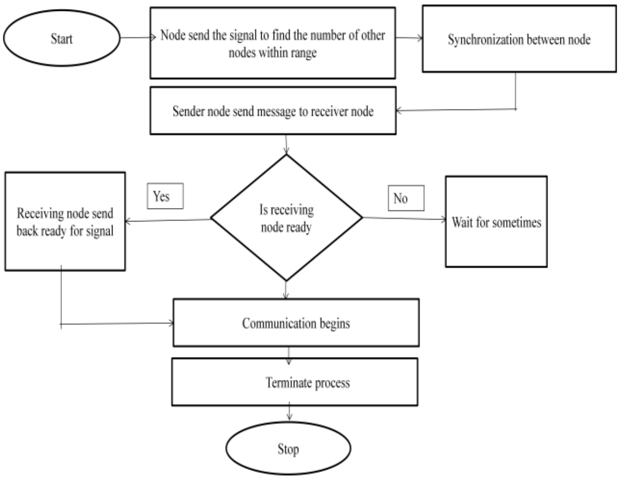

Start
Node send the signal to find the number of other nodes within range
Synchronization between node
Sender node send message to receiver node
Is receiving node ready
Yes
No
Receiving node send back ready for signal
Wait for sometimes
Communication begins
Terminate process
Stop

Fig. 7: Attack of wormhole

The figure 7 describes the attacker occurrence in the network. The attacker occurs during the communication between the nodes. The attacker represented by unique color and unique function .It also labeled as an attacker. The attacker node occurs during the interval 30 ms and 70 ms

Evaluation of sequence number attack detection using different protocol

The four metrics (DAWN, AODV, LEECH, DSDV) that were used in the evaluation of the Sequence Number Attack Detection and countermeasure mechanism are the delivery ratio, the number of false routing packets sent by the attacker, false positive We show that our work has improved on many fronts. Our method has been shown to detect local as well as distributed attacks.

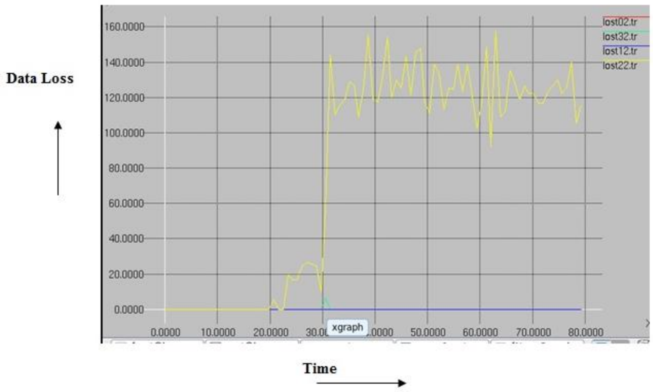


Fig. 8. Evaluation of sequence number attack detection using different protocol (lost12.tr-DAWN; lost22.tr AODV; lost32.tr LEECH, lost32.tr DSDV)

In a work, Stamouli *et al* (13) and Ashmy and Devendran (14) conclude that AODV performs well at all mobility rates and movement speeds. Our conclusions are the same; however, we argue that their definition of mobility (pause time) does not truly represent the dynamic topology of MANETs. Our mobility factor is based on actual relative movement pattern. The only node speeds have shown are 5 meters/ second and 20 meters/ second which, in our opinion, do not cover the complete range. Our mobility factor has a speed range from 0 meters/ second (static scenario) up to 20 meters/ second, and we show how our protocol behaves in the complete range.

According to the analysis that we performed, the most serious attacks are carried out by 'insiders' who carry out their attacks via an attached terminal, not via the network. Consequently, network-based IDS will fail to detect the most damaging attacks. Moreover, the most pervasive network-based IDSs are signature-based and are only able to detect known attacks.

Evaluation of sequence number attack detection using DAWN protocol at different nodes

The evaluation of sequence number attack detection using DAWN protocol at 5, 10, 15 and 20 nodes in terms of efficiency in time is shown in the figure 8.

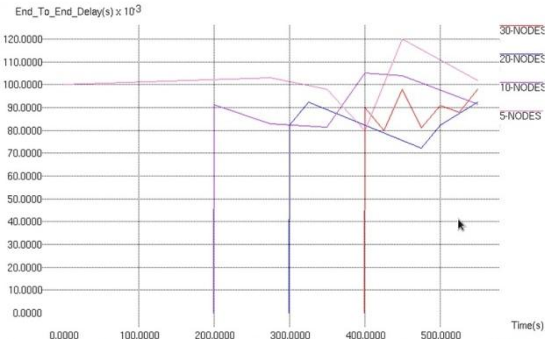


Fig. 8: End To End Delay using DAWN

The evaluation of sequence number attack detection using DAWN protocol at 5, 10, 15 and 20 nodes in terms of Jitter in time is shown in the figure 9.

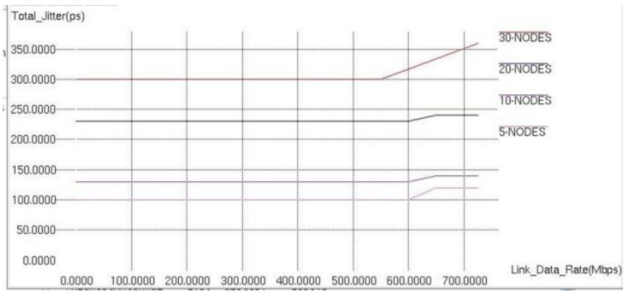


Fig. 9: Evaluation of Jitter using DAWN protocol at different nodes

We presented new techniques that advance the field of intrusion detection in several areas. We have designed novel mechanisms to detect and mitigate aberrant behaviors encountered in Mobile Ad Hoc Networks (MANETs). Since MANETs are comprised of resource-constrained devices, we designed our intrusion detection mechanisms as protocols that monitor network state rather than system state. We also experimented with reactive protocols for MANETs, extending prior research to work with all mobile Ad Hoc routing protocols, not just DAWN. Our experiments and simulations have demonstrated that our protocol is functionally feasible given limited resources.

This study shows the impact and countermeasures of wormhole attacks in wireless network coding systems and the harmful impact of wormholes on system performance and regional nodes' resource utilization. The results via simulations on various scenarios, our algorithm detect the wormholes. In this algorithm, a central node collects the information from all the nodes in the network and analyzes whether there exists a wormhole link. The algorithm leverages the order of the nodes to receive the innovative packet, and utilizes machine learning techniques to distinguish the wormhole cases.

In DAWN, during regular data transmissions, each node records the abnormal arrival of innovative packets and shares this information with its neighbors. This algorithm is efficient and practical without strong assumptions. Furthermore, we theoretically prove that DAWN guarantees a good lower bound of successful detection rate.

6. CONCLUSIONS

An Intrusion Detection System aiming at securing the DAWN protocol has been developed using specification-based technique. The IDS performance in detecting misuse of the DAWN protocol has been discussed. In all the cases, the attack was detected as a violation to one of the DAWN protocol specifications.

From the results obtained, it can be concluded that our IDS can effectively detect Sequence Number Attack, Packet Dropping Attack and Resource Depletion Attack with Incremental Deployment. The method has been shown to have low overheads and high detection rate. The prototype has also given some insight into the problems that arise when trying to run real applications on an Ad Hoc network

In this paper, analyzed the wormhole attack on DAWN protocol with respect to different performance parameters such as throughput ,packet delivery ratio and this conclude the effect of wormhole attack is more on DAWN protocol as compared to others .In future work implement some security algorithm on these protocols to avoid the wormhole attack. This proposed method can be used to find the secured routes and prevent the wormhole nodes in the MANET.

Simulation results validate the ability of our protocol to successfully detect both local and distributed attacks against the DAWN routing protocol. The algorithm also imposes a very small overhead on the nodes, which is an important factor for the resource-constrained nodes.

ACKNOWLEDGEMENT

The author Principal, and Head, Department of Electronics and Communication Engineering, Chandy College of Engineering, Tuticorin for their support.

REFERENCES

[1] C. Karlof and D. Wagner,"Secure Routing inWireless Sensor Networks: Attacks and Counter measures", in Elsevier's Adhoc Network Journal Special Issueon Sensor Network Application and Protocols, vol.1, 2-3,pp.293-315,September2003

[2] Rouba El Kaissi, Ayman Kayssi, Ali Chehab and Zaher Dawy, "DAWWSEN:A Defense Mechanism against Wormhole Attacks In Wireless Sensor Networks", in The Second Internationa lConference on Innovations InInformationTechnology, pp. 1-10, 2005.

[3] Neha Dubey Krishna Kumar International Journal of Computer Applications (0975 – 8887) Volume 114 – No. 14, March 2015.

[4] Y.C. Hu, A. Perrigand D. B. Johnson, "Packet Leashes : A Defense against Wormhole Attacks in Wireless Networks", in 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),pp.1976-1986, 2003.

[5] Thaier Hayajneh, Prashant Krishnamurthy and David Tipper, "SECUND: A Protocol for Secure Neighborhood Creation in Wireless Adhoc Networks",in 5th International Conference on Collaborative Computing : Networking, Applications and Worksharing, vol.1,.2-3,pp.1-10,2009.

[6] Van Tran, Le Xuan Hung, Young-KooLee, Sungyoung Leeand Heejo Lee, "TTM: An Efficient Mechanism to Detect Wormhole Attacks in Wireless Ad-hoc Networks", in 4th IEEE conference on Consumer Communications an Networking Conference, pp. 593-598,2007

[7] L. Huand D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks", in Network and Distributed System Security Symposium(NDSS),SanDiego,February2004

[8] I. Khalil, S. Bagchi and N.B. Shroff, "LITEWORP: A Light weight Countermeasure for the Wormhole Attack in Multihop Wireless Network", in International Conference on Dependable Systems and Networks(DSN), pp.1-22,2005

[9] Chen, Wei Lou, Xice Sunand Zhi Wang, "SLAW: Secure Localization Against Wormhole Attacks Using Conflicting Sets", in Technical Report, The Hong Kong Polytechnic University, pp.111, 2010

[10] Nait Farid Nait-Abdesselam, Brahim Bensaou and Tarik Taleb, "Detecting and Avoiding Wormhole Attacks in Wireless Ad Hoc Networks", in Proceeding of Wireless Communications and Networking Conference, pp. 3117-3122, 2007.

[11] An Approach to Detect Wormhole Attack in AODV based MANET

[12] Joshi Dhara Buch1 and Devesh Jinwala2 Prevention of wormhole attack in wireless sensor network.International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.5, Sep 2011.

[13] Stamouli. (2003). *Real-time intrusion detection for ad hoc networks*. Master's Thesis, University of Dublin, Dublin.

[14] P. Ashmy and S. Devendran *'Distributed Detection Algorithm for Wormholes in Wireless Network'*, Proceedings of International Conference on Technical Innovation in Electrical, Electronics, Computer & Communications (ICTIEECC 2016), PET Engineering College, Vallioor, pp.983-992 March, 2016.