International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Truthful Detection of Selfish Malicious Packet Dropping Nodes in Wireless Ad Hoc Network

R Sharmila[1], K Suresh Joseh[2]
[1]*Department of Computer Science, Pondicherry University, Puducherry, India*
[2]*Department of Computer Science, Pondicherry University, Puducherry, India*

Abstract: ***We present a scalable, effective and practical approach for detection of packet-drop attacks in ad hoc networks. In this attack, a malicious network node chooses to selectively drop packets that are supposed to be forwarded, which results in adverse impact on application good-put and network stability. They are responsible for cooperatively shuttling packets amongst themselves in order to provide the illusion of a network with universal point-to-point connectivity. However, this illusion is shattered—as are implicit assumptions of availability, confidentiality, or integrity—when network routers are subverted to act in a malicious fashion. By manipulating, diverting, or dropping packets arriving at a compromised router, an attacker can trivially mount denial-of-service, surveillance, or man-in-the-middle attacks on end host systems. We have tested our protocol in Emulab and have studied its effectiveness in differentiating attacks from legitimate network behavior.***
Keywords: Malicious network node, Denial-of-Service, Surveillance, Man-in-the-Middle, Point-to-Point nodes

I. INTRODUCTION

The Internet is not a safe place. Unsecured hosts can expect to be compromised within minutes of connecting to the Internet and even well-protected hosts may be crippled with denial-of-service (DoS) attacks. However, while such threats to host systems are widely understood, it is less well appreciated that the network infrastructure itself is subject to constant attack as well. Indeed, through combinations of social engineering and weak passwords, attackers have seized control over thousands of Internet routers. Once a router has been compromised in such a fashion, an attacker may interpose on the traffic stream and manipulate it maliciously to attack others—selectively dropping, modifying, or rerouting packets.

Typically, in an ad hoc network, there are four kinds of possible routing attacks, namely spoofing, fabrication, sinking, and flushing. All routing attack scenarios explained in the literature, such as black holes, worm hole, etc., use single or compounded form of the above four kinds of attacks. Sinking is a malicious behavior of nodes, where nodes do not cooperate in the routing and forwarding operations of the network. Nodes exhibiting sinking behavior maliciously drop data or routing messages. The possible objective of this behavior is to either to selfishly evade from the network responsibilities for resource conservation or to disrupt the network by dropping critical packets. As stated before, cooperativeness of nodes is crucial to the operation of the network. As nodes in the network are autonomous, the neighbor nodes' cooperative behavior needs to be constantly monitored and enforced. Hence, detecting sinking behavior is important for the network integrity.

In wired networks, as the environment is more predictable, detection of sinking behavior is trivial. In wired networks, the service capacity of nodes, channels are known or predictable; this enables the detection of any abnormal behavior of packet dropping. On the other hand, in a wireless, mobile, ad hoc environment, detection of sinking is hindered by the characteristics of the network environment. For example, in wireless networks, dropping of packets can be due to the signal loss or mobility (node moved out of range), etc.

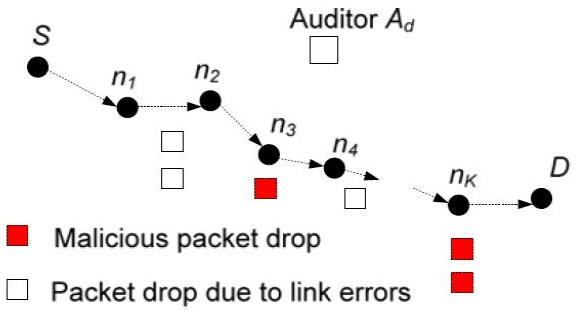


Fig. 1. Network and attack model.

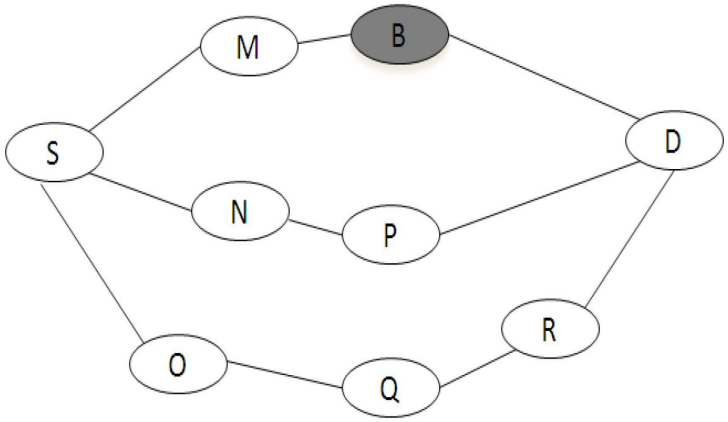
In this paper, we develop a compromised router detection protocol that dynamically infers the precise number of congestive packet losses that will occur. Once the congestion ambiguity is removed, subsequent packet losses can be safely attributed to malicious actions. We believe our protocol is the first to automatically predict congestion in a systematic manner and that it is necessary for making any such network fault detection practical.

II. RELATED WORK

Coverage problems in wireless ad hoc sensor networks were discussed by Chen and Koutsoubos. They proposed communication range reduction as an example measure aimed at energy conservation, which can easily be accommodated by locally adjusting the communication range probabilistically depending on the number of other nodes in that range. Li and Wan developed algorithms to conserve energy for best coverage. Meguerdichian et al discussed the coverage problems by using graph theory models. Huang and Tseng proposed polynomial-time algorithms for coverage problems. A greater diversity of coverage problems including node placement for connected coverage, energy efficient connected coverage, and maintaining connectivity in large WSN were discussed in recent years. Detecting and bypassing the black holes were discussed in, which is very important in sensor placement and positioning. The relationship between the sensing range and communication distance was discussed by Zhang. Grist introduced the novel idea of using homology to infer the sensing coverage using local connectivity.

The problems discussed by the above authors will help detect the boundary of a hole. These papers discuss the detection of a single hole. Very few researchers discussed the detection of multiple holes since they require special and continuous monitoring models to identify multiple holes. The proposed sensor placement and connectivity will be discussed in section 3 which identify multiple disconnected nodes.

Once the sensor nodes are deployed on the field then information transmission will take place. The source node sends the information and the destination node, called base station, receives the information. During the process the information may be lost due to node malfunction. Malfunction may cause partial or total packet dropping. The packet dropping problem was studied by many authors as “suspected nodes and selective forward attacks”. Detecting these nodes is one of the challenging problems. Various models including game theory were proposed for detecting selective forward attacks. In this paper we simulate the forward attack model and detect the nodes that drop packets randomly.



S: Source node
M, N, 0: Intermediate nodes
D: Destination nodes
B: Malicious node

Fig. 2 Data Routing

III. SYSTEM MODEL

Our work proceeds from an informed, yet abstracted, model of how the network is constructed, the capabilities of the attacker, and the complexities of the traffic validation problem. In this section, we briefly describe the assumptions underlying our model.

A. Network models

We consider a network to consist of individual homogeneous routers interconnected via directional point-to point links. This model is an intentional simplification of real networks (e.g., it does not include broadcast channels or independently failing network

interfaces) but is sufficiently general to encompass such details if necessary. Unlike our earlier work, we assume that the bandwidth, the delay of each link, and the queue limit for each interface are all known publicly.

Within a network, we presume that packets are forwarded in a hop-by-hop fashion, based on a local forwarding table. These forwarding tables are updated via a distributed link-state routing protocol such as OSPF or IS-IS. This is critical, as we depend on the routing protocol to provide each node with a global view of the current network topology. Finally, we assume the administrative ability to assign and distribute cryptographic keys to sets of nearby routers. This overall model is consistent with the typical construction of large enterprise IP networks or the internal structure of single ISP backbone networks but is not well suited for networks that are composed of multiple administrative domains using BGP. At this level of abstraction, we can assume a synchronous network model.

We define a path to be a finite sequence hr1; r2; . . . rni of adjacent routers. Operationally, a path defines a sequence of routers a packet can follow. We call the first router of the path the source and the last router its sink; together, these are called terminal routers. A path might consist of only one router, in which case the source and sink are the same. Terminal routers are leaf routers: they are never in the middle of any path.

An x-path segment is a consecutive sequence of x routers that is a subsequence of a path. A path segment is an x-path segment for some value of x > 0. For example, if a network consists of the single path ha; b; c; di, then hc; di and hb; ci are both two-path segments, but ha; ci is not because a and c are not adjacent.

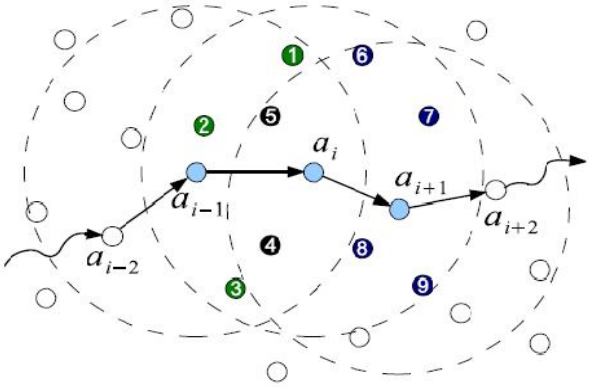


Fig. 3 (a) Observe node (network view)

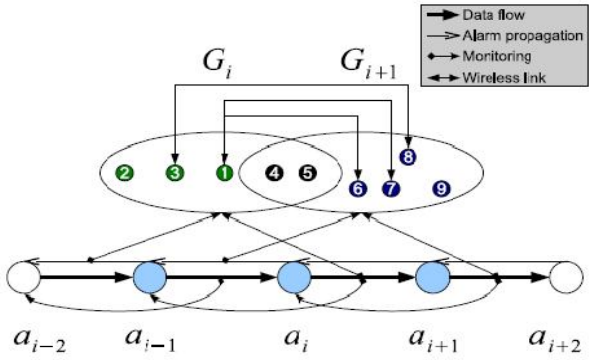


Fig. 3 (b) Observe node (logic view)

B. Threat Models

As explained in Section 1, this paper focuses solely on data plane attacks (control plane attacks can be addressed by other protocols with appropriate threat models). Moreover, for simplicity, we examine only attacks that involve packet dropping. However, our approach is easily extended to address other attacks such as packet modification or reordering similar to our previous work. Finally, as in, the protocol we develop validates traffic whose source and sink routers are uncompromised.

A router can be traffic faulty by maliciously dropping packets and protocol faulty by not following the rules of the detection protocol. We say that a compromised router r is traffic faulty with respect to a path segment during if contains r and, during the period of time, r maliciously drops or misroutes packets that flow through. A router can drop packets without being faulty, as long as the packets are dropped because the corresponding output interface is congested. A compromised router r can also behave in an arbitrarily malicious way in terms of executing the protocol we present, in which case we indicate r as protocol faulty. A protocol faulty router can send control messages with arbitrarily faulty information, or it can simply not send some or all of them. A faulty router is one that is traffic faulty, protocol faulty, or both. Attackers can compromise one or more routers in a network. However, for simplicity, we assume in this paper that adjacent routers cannot be faulty. Our work is easily extended to the case of k adjacent faulty routers.

IV. PROBLEM STATEMENT

Under the system and adversary models defined above, we address the problem of identifying the nodes on PSD that drop packets maliciously. We require the detection to be performed by a public auditor that does not have knowledge of the secrets held by the nodes on PSD. When a malicious node is identified, the auditor should be able to construct a publicly verifiable proof of the misbehavior of that node. The construction of such a proof should be privacy preserving, i.e., it does not reveal the original

information that is transmitted on PSD. In addition, the detection mechanism should incur low communication and storage overheads, so that it can be applied to a wide variety of wireless networks.

V. PROTOCOL X

Protocol X detects traffic faulty routers by validating the queue of each output interface for each router. Given the buffer size and the rate at which traffic enters and exits a queue, the behavior of the queue is deterministic. If the actual behavior deviates from the predicted behavior, then a failure has occurred. We present the failure detection protocol in terms of the solutions of the distinct subproblems: traffic validation, distributed detection, and response.

A. Traffic Validation

The first problem we address is traffic validation: what information is collected about traffic and how it is used to determine that a router has been compromised.

B. Single Packet Loss Test

If a packet with fingerprint *fp* and size *ps* is dropped at time *ts* when the predicted queue length is $q_{pred}(ts)$, then we raise an alarm with a confidence value c_{single}, which is the probability of the packet being dropped maliciously.

The mean μ and σ standard deviation σ of X can be determined by monitoring during a learning period. We do not expect μ and σ to change much over time, because they are in turn determined by values that themselves do not change much over time. Hence, the learning period need not be done very often.

A malicious router is detected if the confidence value c_{single} is at least as large as a target significance level s_{single}.3

C. *Combined Packet Losses Test*

The second test is useful when more than one packet is dropped during a round and the first test does not detect a malicious router. It is based on the well-known Z-test[4]. Let L be the set of $n > 1$ packets dropped during the last time interval. For the packets in L, let *ps* be the mean of the packet sizes, q_{pred} be the mean of $q_{pred}(ts)$ (the predicted queue length), and q_{act} be the mean of $q_{act}(ts)$ (the actual queue length) over the times the packets were dropped. We test the following hypothesis: "The packets are lost due to malicious attack": $\mu > q_{limit} - q_{pred} - ps$.

For the standard normal distribution Z, the probability of Prob(Z < z1) gives the confidence value $c_{combined}$ for the hypothesis. A malicious router is detected if $c_{combined}$ is at least as large as a target significance level s^{level} combined.

One can question using a Z-test in this way because the set of dropped packets are not a simple random sample. But, this test is used when there are packets being dropped and the first test determined that they were consistent with congestion loss. Hence, the router is under load during the short period the measurement was taken and most of the points, both for dropped packets and for nondropped packets, should have a nearly full Q. In this Section we show that the Z-test does in fact detect a router that is malicious in a calculated manner.

VI. PROPOSED SYSTEM

The main aim is to construct an algorithm which is accurate for selective packet dropping attack detection made by malicious node. This method also provides a correct and openly verifiable method that is a proof to support the detection result. Entropy method is used to detect malicious behavior to improve detection accuracy. In Fig 1 indicates source, destination and three intermediate nodes where packet is transmitted and malicious node drops selective packets along with link error due to which some packets are dropped. link errors packet dropping takes place because of the low connectivity between the nodes i.e. Due to the harsh channel condition, a malicious node can cover-up its attack by selectively dropping a small number of highly important packets. Gray hole attack is used by malicious node to drop some packets in the network.

We can determine the nodes on the routing path from the network model that causes the packet losses. This is done by the auditor who is not aware of any secrets above the node. When a malicious node is particularly identified, auditor provides a publicly verifiable proof method. Each intermediate nodes sends an acknowledgment to the source after receiving the key during key transmission phase. Each intermediate node provides a bitmap showing the status of each packet whether the packet is lost or received in a sequence of successive packet transmissions to the auditor.

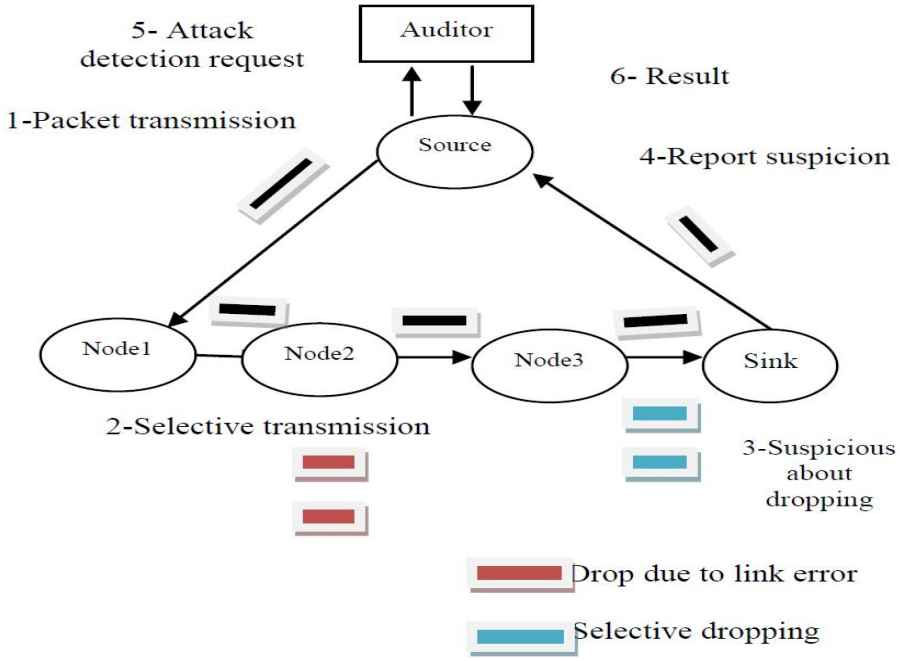


Fig. 4 Packet transmission from source to destination

VII. PROPOSED ALGORITHM

The proposed detection mechanism keeps a log of the individual one-way packet latencies as well as the short-term and long-term average latencies of packets preceding a particular packet (say i). If Di is the latency of the ith packet then the short-term average delay (savgi(m)) and the long-term average delay (lavgi(n)) with window sizes of m and n respectively can be defined as

Algorithm 1 Packet Drop Attack Detection Algorithm

1: initialize xn = 0
2: function CLASSIFY(i)
3: slopeI (i) = (savgi − savgi−k)/k
4: slopeF (i) = (savgi+l − savgi+p)/(l − p)
5: if savgi − lavgi ≤ (0.005lavgi) then
6: cause = attack;
7: else if slopeI (i) − slopeF (i) < 0 then
8: cause = attack;
9: else
10: cause = congestion;
11: end if
12: if cause == attack then
13: ALERT()
14: end if
15: end function
16:
17: function ALERT()
18: if xn = 0 then
19: start timer for value t;
20: end if
21: update xn = xn + 1;

22: if timer has expired then
23: if xn > η then
24: generate alarm for “Packet Drop Attack”;
25: end if
26: xn = 0;
27: end if
28: end function
29:{
30: loop
31: for each packet arrival i do
32: calculate delay Di;
33: consider n previous arrivals;
34: calculate lavgi = _n
j=1(Di−j/n);
35: consider m previous arrivals;
36: calculate savgi = _m
j=1(Di−j/m);
37: if out-of-sequence packet then
38: {/* packet loss detected */}
39: wait for l new arrivals;
40: CLASSIFY(i)
41: end if
42: end for
43: end loop when session is terminate

VIII.EXPERIMENTAL RESULT

In this section, Parameters that influence packet delivery are being plotted to get visual results. These parameters includes-

1) *Packet Drop:* This parameter shows measure of dropped packets. The reason of packet dropping may be any. Packet dropping may happen due to adversary nodes, network congestion etc.
2) *Total Packets:* This parameter depicts the total number of packets that are sent by the source node toward base station. They may successfully reach at base station or may be dropped by some malicious node.
3) *Packet Delivery Ratio:* This parameter is the ratio of successfully transmitted packets to the total number of packets. This parameter depicts successfully transmitted packets.

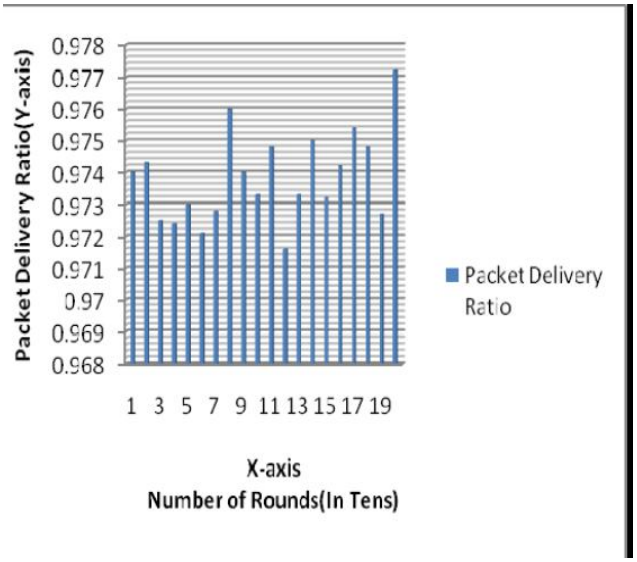


Fig. 5. Number of Rounds Vs Packet Delivery Ratio

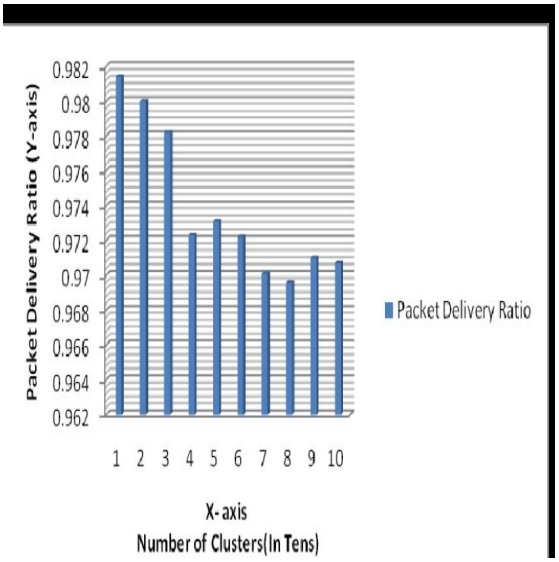


Fig. 6 Number of Clusters Vs Packet Delivery Ratio

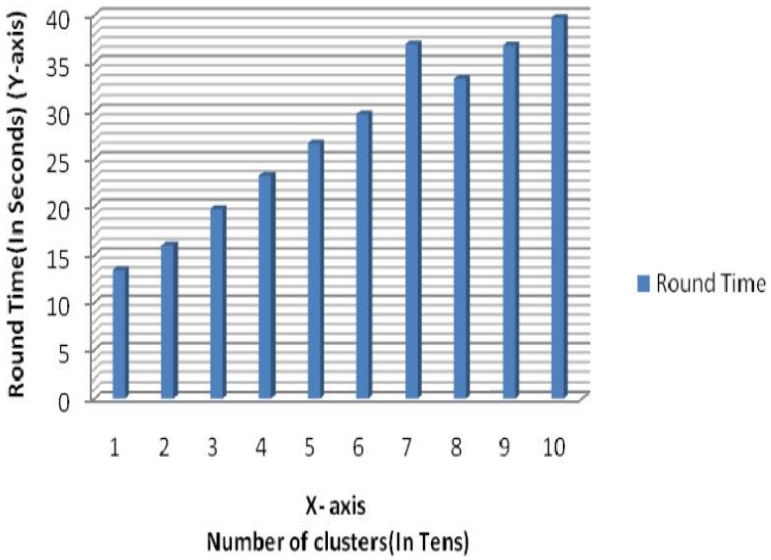


Fig. 7 Number of Clusters Vs Round Time

"Figure 5" and "Figure 6" depicts visual results regarding successful packet delivery. In Figure 5,the number of sensor nodes used for simulation are 20, while in Figure 6 and Figure 7 number of nodes vary according to clusters for comparison purposes. For comparative results, number of clusters has been varied from 1 to 10 and each cluster carries 5 nodes. For our detection strategy, packet delivery ratio is independent of number of 668 rounds, but it is dependent on number of clusters or network size.

IX. CONCLUSIONS

Detection of selfish nodes is crucial in WMNs since these nodes don't forward packets for other nodes and degrade the performance of the networks. In this paper, we showed that compared with conventional detection algorithms that utilize only the distribution of the number of lost packets, exploiting the correlation between lost packets significantly improves the accuracy in detecting malicious packet drops. Such improvement is especially visible when the number of maliciously dropped packets is comparable with those caused by link errors. To correctly calculate the correlation between lost packets, it is critical to acquire truthful packet-loss information at individual nodes. We developed an HLA-based public auditing architecture that ensures truthful packet-loss reporting by individual nodes. This architecture is collusion proof, requires relatively high computational capacity at the source node, but incurs low communication and storage overheads over the route. To reduce the computation overhead of the baseline construction, a packet-block-based mechanism was also proposed, which allows one to trade detection accuracy for lower computation complexity. The implementation and optimization of the proposed mechanism under various particular protocols will be considered in our future studies.

REFERENCE

[1] H. C. Papadopoulos, S. Member, and C. W. Sundberg, "Shared Time-Division Duplexing (STDD): Impact of Runlengths of Dropped Packets and Fast-Speech Activity Detection," vol. 47, no. 3, pp. 856–870, 1998.
[2] F. Anjum and R. Talpade, "LiPaD: lightweight packet drop detection for ad hoc networks," Veh. Technol. Conf. 2004. VTC2004-Fall. 2004 IEEE 60th, vol. 2, no. C, p. 1233–1237 Vol. 2, 2004.
[3] A. T. Mizrak, Y. C. Cheng, K. Marzullo, and S. Savage, "Detecting and isolating malicious routers," IEEE Trans. Dependable Secur. Comput., vol. 3, no. 3, pp. 230–244, 2006.
[4] A. T. Mzrak, S. Savage, and K. Marzullo, "Detecting Malicious Packet Losses," IEEE Trans. Parallel Distrib. Syst., vol. 20, no. 2, pp. 191–206, 2009.
[5] J. F. Charles Joseph, B. S. Lee, A. Das, and B. C. Seet, "Cross-layer detection of sinking behavior in wireless Ad Hoc networks using SVM and FDA," IEEE Trans. Dependable Secur. Comput., vol. 8, no. 2, pp. 233–245, 2011.
[6] Y. Reddy, J. Durand, and S. Kafle, "Detection of packet dropping in wireless sensor networks," ITNG2010 - 7th Int. Conf. Inf. Technol. New Gener., pp. 879–884, 2010.
[7] F. Na, "Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks : Proposals and Challenges," vol. 13, no. 4, pp. 658–672, 2011.
[8] J. Sen, "An Efficient Algorithm for Detection of Selfish Packet Dropping Nodes in Wireless Mesh Networks," Comput. Inf. Syst. Ind. Manag. Appl. CISIM 2010 Int. Conf., vol. 3, no. i, p. 8, 2010.

[9] X. Li, R. Lu, X. Liang, and X. S. Shen, "Side Channel Monitoring : Packet Drop Attack Detection in Wireless Ad Hoc Networks," Commun. Soc., 2011.
[10] L. Sánchez-Casado, G. Maciá-Fernández, and P. García-Teodoro, "An efficient cross-layer approach for malicious packet dropping detection in MANETs," Proc. 11th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. - 11th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC-2012, pp. 231–238, 2012.
[11] S. Sultana, G. Ghinita, E. Bertino, and M. Shehab, "A lightweight secure scheme for detecting provenance forgery and packet dropattacks in wireless sensor networks," IEEE Trans. Dependable Secur. Comput., vol. 12, no. 3, pp. 256–269, 2015.
[12] A. Chaudhary, V. N. Tiwari, and A. Kumar, "Design an anomaly based fuzzy intrusion detection system for packet dropping attack in mobile ad hoc networks," Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014, pp. 256–261, 2014.
[13] T. Shu and M. Krunz, "Privacy-preserving and truthful detection of packet dropping attacks in wireless ad hoc networks," IEEE Trans. Mob. Comput., vol. 14, no. 4, pp. 813–828, 2015.
[14] S. Uyyala and D. Naik, "Anomaly based intrusion detection of packet dropping attacks in mobile ad-hoc networks," 2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014, vol. 2, pp. 1137–1140, 2014.
[15] S. Pal, B. Sikdar, and J. Chow, "Real-time detection of packet drop attacks on synchrophasor data," 2014 IEEE Int. Conf. Smart Grid Commun. SmartGridComm 2014, pp. 896–901, 2015.
[16] A. Kaur and S. S. Sran, "Detection of packet-dropping attack in recoverable concealed data aggregation protocol for homogeneous wireless sensor networks," Int. Conf. Adv. Comput. Commun. Technol. ACCT, vol. 2015–April, pp. 666–670, 2015.
[17] S. Vhora, R. Patel, and N. Patel, "Rank Base Data Routing (RBDR) Scheme using AOMDV : A Proposed Scheme for Packet Drop Attack Detection and Prevention in MANET."
[18] P. Vadivazhagu and P. D. Selvam, "Network lifetime enhancement method for sink relocation and packet drop detection in wireless sensor networks," 2015 Int. Conf. Commun. Signal Process. ICCSP 2015, pp. 500–504, 2015.
[19] M. V. Mohan, "HLA SIGNATURE AND ENTROPY BASED METHOD TO DETECT MALICIOUS PACKET," 2017.
[20] J. Sen, M. G. Chandra, P. Balamuralidhar, S. G. Harihara, and H. Reddy, "A distributed protocol for detection of packet dropping attack in mobile ad hoc networks," Proceeding - 2007 IEEE Int. Conf. Telecommun. Malaysia Int. Conf. Commun. ICT-MICC 2007, no. May, pp. 75–80, 2007.