Distributed Trust Based Co-operative Bait Detection Scheme for Detecting Collaborative Attacks in MANET

E. R. Rakesh Kumar[1], P. Panchanathan[2]
[1]Research Scholar, Dept. of Computer Science & Engineering, Annamalai University, Chidambaram, India
[2]Assistant Professor, Dept. of Computer Science & Engineering, Annamalai University, Chidambaram, India

***Abstract*: In mobile ad hoc networks (MANETs), cooperating nodes with each other is a key requirement for the establishment of communication among nodes. In the presence of wicked nodes, this kind of necessity may lead to severe security related concerns. In case, such nodes may interrupt the routing process. In our context for preventing malicious nodes we have proposed a distributed trust based co-operative bait detection scheme for detecting collaborative attacks in MANET. Here using the trust value which estimated using Bayesian interference the Bait detection process is invoked. For this the source node selects an adjacent node using the random scheduling process. This is the address of this adjacent node is used as bait destination address to bait malicious nodes in order to send a reply RREP message. By this the bait detection is raised. After the detection of malicious node the PDR value is ensured with the Threshold value, from this the again the bait detection process is triggered. From the random schedule table the nodes with less trust value which is considered to be as un-trusted nodes are removed instantly.**

***Keywords*: MANET, Bait, Collaborative Attacks.**

1. Introduction

A. Mobile ad hoc network (MANET)

MANET is a multi-hop wireless network are composed of autonomous nodes that communicate with each other by forming dynamic topology such that nodes can easily join or leave the network at any time without any fixed infrastructure such as access points or base station and maintaining connections in a decentralized manner. The network over radio links are caused due to the self-organization of the mobile nodes. Each device in a MANET is free to move independently in any directions [1]. The infrastructure less property and the easy deployment along with the self-organizing nature makes them useful for many applications like military applications, mobile social networks, emergency deployment, intelligent transportation systems and fast response to disasters [2].

MANET also throws a security challenge due to their features of open medium, dynamically changing topologies, reliance on cooperative algorithms, absence of centralized monitoring points, and lack of clear lines of defense moderate bandwidth, limited battery power, computational power and limited resources. So mobile ad-hoc networks are vulnerable to several different attacks [3].

B. Collaborative Attacks in MANET

The collaborative attacks are defined as two or more types of attacks such as the black hole attacks and the wormhole attacks, which synchronized simultaneously in the network in a collaborative way [4]. It is a synchronized attacks where a system is distributed by more than one attacker simultaneously or involving two or more colluding nodes that can be processed using wired or wireless link and triggered by single or multiple attackers. Collaborative attacks (CA) occur when more than one attacker or running process synchronize their actions to disturb a target network but not necessarily in collaboration where every attack is launched by a specialized expertise. These attacks can be classified into two different categories [5].

Direct Collaborative Attacks: Here, the attacker nodes are already in existence in the original network or a malicious node joins the network or an internal node is compromised in the network. This kind of collaborative attacks can be referred to as direct collaborative attacks. For examples, Black hole and Wormhole attack.

Indirect Collaborative Attacks: The attacks in this category use different non-existent nodes in order to fake other nodes to redirect data packets to malicious node. This kind of collaborative attacks can be referred to as indirect collaborative attacks. For examples, Sybil and Routing table overflow attacks [6].

C. Collaborative attack detection in MANET

Collaborative attacks in ad hoc networks carriage challenges to the detection system. Malicious nodes may collude to conduct more complex and subtle attacks to prevent detection or identification. To detect against collaborative attacks essential that monitoring and detection agents collaborate efficiently. The collaboration should include each existing node in the network.

The main challenges include:

1) Integrating the information from multiple nodes in efficient manner.
2) For developing the attack detection mechanisms that should be robust against noise in the information.
3) For discovering the effective relationship between the range of network from which the information is

integrated and the detection capabilities of the mechanisms;

4) Determining the trade-off between the detection granularity and the dynamics of the networks [7].

D. Problem Identification

In [9], a Co-operative Bait Detection Scheme (CBDS) has been proposed. In which the source select an adjacent node as the bait destination address. But selecting the adjacent node among the neighbors of the source is not described. If the attacker is able to find out that adjacent node, it will try to avoid the Bait REQ requests. The detection process is invoked based on the packet delivery ratio (PDR) metric by the destination. But PDR alone will not be sufficient to detect the misbehaving attacks. Moreover, the detection delay will be increased since the detection process is invoked only when the destination send an alarm.

2. Literature Review

Reshma Lill Mathew and P. Petchimuthu [2] have proposed a collaborative watchdog based on contact dissemination with a log file system. The watchdog has detected a selfish node in the network then spread the information to other nodes when contact occurs. The detection of the contacts among the nodes is performed based on the node's watchdog for the detecting the selfish nodes. Log file system have used for reducing the detection time of the selfish node. After forwarding the packets from the neighbor node to next neighbor node, neighbor node could not overhear the packet dropping of next neighbour node either if transmission collides between source and neighbour node or neighbour node is not within the transmission range of next neighbour node. When this happens it could not provide the security.

Tao Gong and Bharat Bhargava [4] have proposed to defend the ad hoc network under collaborative attacks such as the black hole and the wormhole attacks using new tri-tier cooperative immunization from the inspiration of the human immune system. Tri-tier immunization includes native immune tier to recognize known attacks, adaptive immune tier to learn unknown attacks and parallel immune tier is built with the cloud-computing infrastructure for increasing both the efficiency and robustness of immune computation. The approach provides immunization to isolate the nodes under attacks by the network reconfiguration. Still it provides security reconfiguration is not possible.

Mahdi Nouri et al [8] have proposed a collaborative technique for detecting a wormhole attack in that neighborhood using clustering. Monitor node initiates the detection process by passing messages between the nodes and depending on the messages received determine suspected nodes that sent to the monitor node. The suspected nodes receive at least a minimum number of votes or only one vote are finally detected as malicious nodes by inspecting the votes at monitor node and isolate malicious nodes from a group of nodes in routing process. But, using this technique not possible for detecting wormhole attack in the form of out of band attack. When there is congestion or collision, a node may be dropping packets due to overloaded, and so the algorithm will not work properly. And also if a monitor node continuously monitoring the detection process, it may cause exhausting of battery power because of overhead of being the monitor node.

Jian-Ming Chang et al [9] have proposed a cooperative bait detection scheme (CBDS) by designing a DSR based routing mechanism for detecting and preventing malicious nodes that attempts to launching gray hole/collaborative black hole attacks in MANETs that incorporates the advantages of both proactive and reactive response. Using a reverse tracing technique malicious nodes are detected and prevented from participating in the routing operation. When a significant drop occurs in the packet delivery ratio, an alarm is sent by the destination node back to the source node to trigger the detection mechanism again and the dynamic threshold value can be adjusted according to the network performance. However, if a lower the value is set, some of neighbors of the suspicious node may not be found.

JaydipSen et al [10] have proposed a distributed protocol for detection of packet dropping attack based on cooperative participation of the nodes in a MANET. The protocol works through cooperation of some security components that are present in each node in the networks such as monitor, trust collector, trust manager, trust propagator and whistle blower by using complementary relationship between cryptographic key distribution and intrusion detection activity. The redundancies in routing information make the detection scheme highly robust and secure and using of controlled flooding technique has very low communication overhead. However, after finding the malicious node it does not consider the technique for isolating the malicious node from participating in routing process.

Chang Wu Yu et al [11] have proposed a distributed and cooperative mechanism for detecting potential multiple black hole nodes through collection of some local information. From the information, nodes evaluate that there exists any suspicious node among their one-hop neighbors. After finding the node as a suspicious, a cooperative procedure will be initiated to further check the potential black hole nodes. Then the global reaction is initiated to form a proper notification system to send warnings to the whole network. However, overhearing for collection of local information does not work always properly in situation like collision or weak signal. It leads to incorrect evaluation of the behaviour of the suspicious node.

Weichao Wang et al [12] have developed a new mechanism for audit based detection of collaborative packet drop attacks using hash function based method to generate node behavioral proofs that contain information from both data traffic and forwarding paths. Intermediate node construct a Bloom filter based on the contents of the packets to generate the behavioral proof. It allow the system to successfully locate the routing segment in which packet drop attacks are conducted. However, other nodes cannot find the difference between an audit packet and a common data packet. Security is based on the value of its behavioral proof. So it is not efficient. If there is no malicious node all packets are delivered to destination without any packet dropping at intermediate node. So it does not analyse any scenario for delivery of packet ratio at destination.

Sukla Banerjee [13] have proposed detection and removal of cooperative black and gray hole attack in MANETs. The total data traffic is divided into small blocks for ensuring an end-to-end checking. Before sending any block source sends a prelude message to the destination to aware the incoming block. Flow of the traffic is monitored by the neighbors of each node. At the end of the transmission destination node sends postlude message containing the no of data packets received. Using this ack source node check whether the data loss is within the tolerable range, if not then the source node start the process of detecting and removing malicious node by collecting the response from the monitoring nodes. However, the ability of this algorithm is based on finding the threshold probability of non-malicious packet drop. If the threshold probability for non-malicious packet drop is low, this algorithm identifies any malicious behaviour. But also it means that increases the false detection rate.

3. Proposed Solution

A. Overview

In this paper, we propose to design a Distributed Trust model co-operative Bait detection scheme detecting collaborative attacks in MANET. The Bait detection process is invoked based on the trust value which is derived using the Bayesian inference. A trust value is estimated for each node from the direct observations. When the trust value of any intermediate node falls below a minimum threshold value, the co-operative Bait detection scheme will be invoked by the source. Moreover, if the trust value of any nodes in the random schedule table becomes low, it will be removed from the table.

The source node selects the adjacent nodes based on the random scheduling method. The source node collects the address details of the intermediate nodes from the routing table forming a random schedule table. This table consists of one hop neighbours, their address and a random time stamp value. The address of the adjacent node is used as bait destination address to bait malicious nodes. The source node selects the adjacent node from the random schedule table having latest time stamp value and invokes the bait detection scheme. It then marks time stamp of the selected node as expired so that next time another node from the table can be selected.

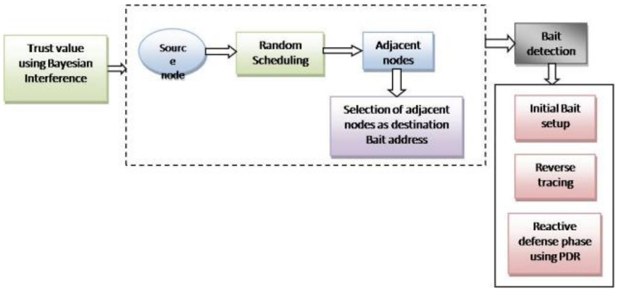


Fig. 1. Block diagram

B. Trust evaluation for Bait detection

The trust value [14] is derived using with the direct observation from an observer node. Bayesian inference deduces the estimation of the unknown probability by using observation. Each observer can listen in the packets forwarded by an observed node and compare them with original packets so that the observer can identify the malicious behaviours of the observed node. From Bayes' theorem the trust model formulation is given by,

$$f(\phi, v \mid u) = \frac{p(u \mid \phi, v) f(\phi, v)}{\int_0^1 p(u \mid \phi, v) f(\phi, v) d\phi'} \quad [14]\ (1)$$

where ϕ - is a random number $0 \le \phi \le 1$, denoted by Φ which is the degree of belief,

u- Number of packets is forwarded correctly,

v- Number of packets is received by a node,

$p(u \mid \phi, v)$- Likelihood function, which follows a binomial distribution

The $p(u \mid \phi, v)$ follows a binomial distribution which is given by,

$$p(u \mid \phi, v) = \binom{v}{u} \phi^u (1-\phi)^{v-u} \quad (2)$$

The prior distribution $f(\phi, v)$ follows the Beta distribution,

$$Beta(\phi; \alpha, \beta) = \frac{\phi^{\alpha-1}(1-\phi)^{\beta-1}}{\int_0^1 \phi^{\alpha-1}(1-\phi)^{\beta-1} d\phi} \quad (3)$$

where $0 \le \phi \le 1$, $\alpha > 0, \beta > 0$

Therefore $f(\phi, v \mid u) \sim Beta(\alpha + u, \beta + v - u)$ (4)

The expectation of Beta distribution is,

$$E[\Phi] = \frac{\alpha}{\alpha + \beta} \quad (5)$$

Owing to the replication of (5) the trust value is calculated iteratively. At the beginning, there is no observation.

The distribution $f(\phi, v)$ is $Beta(\phi;1,1)$ at the beginning,

The trust evaluation is given by

$$E_n[\Phi] = \frac{\alpha_n}{\alpha_n + \beta_n} \quad (6)$$

where $\alpha_n = \alpha_{n-1} + u_{n-1}$, $\beta_n = \beta_{n-1} + v_{n-1} - u_{n-1}$

$\alpha_0 = \beta_0 = 1$, $n \in Z+$

Therefore, the trust value T^S from the direct observation from the trust evaluation using Bayesian inference is given by,

$$T^S = E_n[\Phi] \quad (7)$$

The trust value is calculated and it is checked with the minimum threshold value. As the trust value of any intermediate node falls below a minimum threshold value $T^S < TH_m$, the co-operative Bait detection scheme will be invoked by the source.

C. Selection of Adjacent nodes

Next to the trust value calculation the source node selects the adjacent node in the sense that the address of this node is used as bait destination address to bait malicious nodes to send a reply RREP message. The adjacent node is selected from the random schedule table having latest time stamp value and invokes the bait detection scheme. The random scheduling is based upon the routing table. The routing table consists of the distance of one hop neighbours, their address and a random time stamp value.

The timestamp can be calculated from the delay time as follows,

$$\partial = (t_4 - t_1) - (t_3 - t_2) \quad [15] \quad (8)$$

t_1 - timestamp of the request packet transmission
t_2 - timestamp of the request packet reception
t_3 - timestamp of the response packet transmission
t_4 - timestamp of the response packet reception

	Distance	Address	Random time stamp value
One hop neighbours			

Fig. 2. Random scheduling table

D. Bait detection

After the selection of the adjacent node by the source node, malicious nodes are thereby detected and prevented from participating in the routing operation, using a reverse tracing technique [9].

The Co-operative Bait Detection Scheme (CBDS) comprises three steps: 1) the initial bait step; 2) the initial reverse tracing step; and 3) the shifted to reactive defence step.

1) Initial Bait Step

Here the source node selects an adjacent node n_r within its one-hop neighborhood nodes and cooperates with this node by taking its address as the destination address of the bait RREQ′. The bait phase is activated whenever the bait RREQ′ is sent earlier for seeking the initial routing path. The bait analysis procedure is as follows.

a) If n_r node had not launched a black hole attack, then after the source node had sent out the RREQ′, the other nodes has sent the RREP indicates that the malicious node is present in the reply routing. So in order to detect the route a reverse tracing program is initiated.
b) If only the n_r has sent the RREP for the RREQ′ from the source node, there was no other malicious node in the network except the n_r.
c) If both n_r and the other nodes in the network have sent the RREP shows that the malicious node is present in the route reply.
d) If the n_r does not send the RREP intentionally, then n_r would be directly directed into the blackhole list by the source node.

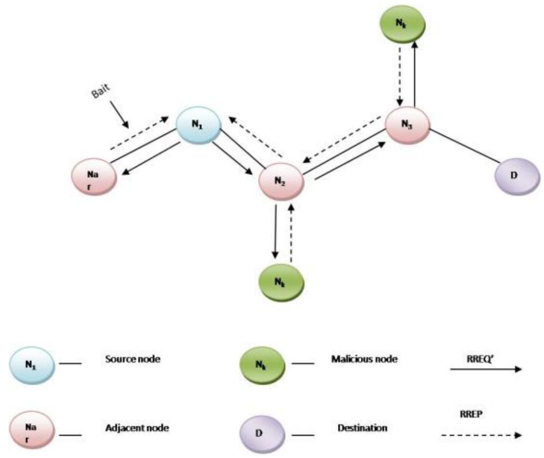


Fig. 3. Random selection of a cooperative bait address

2) Reverse Tracing Step

Using the reverse tracing setup the malicious nodes are detected through the route reply RREP for the RREQ′ message. If a malicious node has received the RREQ′, it will reply with a false RREP [9].

Initially an address P-list and a route information K_k list is created,

$$P = \{n_1 \ldots n_k \ldots n_m \ldots n_r\} \quad [9] \quad (9)$$

$$K_k = \{n_1 \ldots n_k\} \quad (10)$$

So when a malicious node n_m, replies with a false RREP, this address P-list is recorded in the RREP. If the node n_k receives the RREP, it will separate the P-list by the destination address n_1 of the RREP in the IP field and get the address list $K_k = \{n_1 \ldots n_k\}$, where K_k represents the route information from source node n_1 to destination node n_k. After that, node n_k determines the differences between the address P-list and K_k list.

$$K'_k = P - K_k \quad (11)$$

$$K'_k = \{n_{k+1} \ldots n_m \ldots n_r\} \quad (12)$$

where K'_k – route information to the destination node

K'_k is stored in the RREP's "Reserve field" and then they are reverted to the source node. The source node receives the RREP and the K'_k list of the nodes which received the RREP. In order to ensure that K'_k does not come from the malicious node, the n_k node after receiving the RREP compares

- A. the source address in the IP fields of the RREP;
- B. the next hop of n_k in the $P = \{n_1, \ldots n_k, \ldots n_m, \ldots n_r\}$;
- C. one hop of n_k;

If A is not the same with B and C, then the received K'_k performs a forward back. Otherwise, n_k have to just forward back the K'_k that was produced by it.

The trusted set T is given by,

$$T = P - S \quad (13)$$

where S is the Dubious path information

$$S = K'_1 \cap K'_2 \ldots K'_k \quad (14)$$

Representation of malicious node in the route

The below figure shows the operation for the detection of the tracing set up

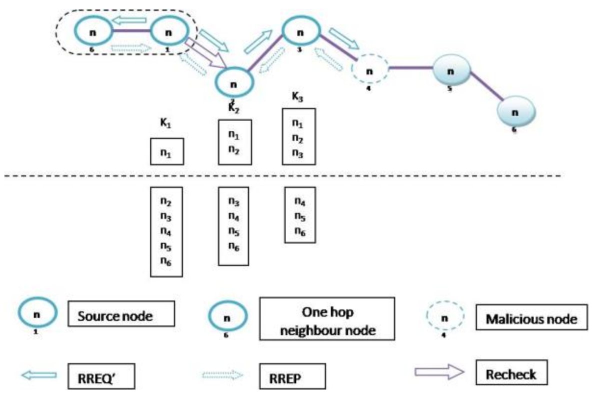


Fig. 4. Reverse tracing program for the CBDS approach

Let us consider a case such that a single malicious node n_4 is present in the route.

1) Initially the source node n_1 pretends to send a packet to the destination node n_6. While sending the RREQ′, node n4 replies with a false RREP with the address list P= {n_1, n_2, n_3, n_4, n_5, n_6}.
2) Node n_5 is a random node which is filled by n_4. If n_3 had received the RREP by n_4, it first separates the P-list with the destination address n_1 of RREP in the IP field. From this it gets the address K_3= {n_1, n_2, n_3}.
3) Next to this n_3 conducts the set difference and acquires K'_3= {n_4, n_5, n_6}, then replies with K'_3 and RREP to the source node n_1. Similarly, n_2 and n_1 performs the same process after receiving the RREP such that K'_2= {n_3, n_4, n_5, n_6} and K'_1= {n_2, n_3, n_4, n_5, n_6}
4) The dubious path information of the malicious node S= {n_4, n_5, n_6}
5) After this the source node calculates the trust value set T= P-S, from this we can obtain T= {n_1, n_2, n_3}
6) Next the source node will send the test packets to this path and the recheck message to n_2, requesting it to enter the promiscuous mode and listening to n_3. As the result of the listening phase, it could be found that n_3 might divert the packets to the malicious node n_4; hence, n_2 would revert the listening result to the source node n_1, which would record n_4 in a blackhole list.
7) If nodes n_4 and n_5 were cooperative malicious nodes the trusted set is T= {n_1, n_2, n_3}, n_2 is requested to listen to which node n_3 might send the packets. Either n_5 or n_4 would be detected, and their cooperation is stopped. Hence, the remaining nodes would be baited and detected.

Elimination of un-trusted nodes:

Here the trust values are estimated for the nodes in the random scheduling table. In this case if the trust value is less for any nodes in the random schedule table, they are removed immediately. It can be expressed as,

$$N_1 < T^S \quad (15)$$

where N_1 is the node having less trust value

Overall algorithm

1. At the start the trust value is calculated from the observer node and it is checked with the minimum threshold value.
2. If any intermediate node falls below a minimum threshold value the bait detection scheme is invoked by the source node.
3. The source node for this selects the adjacent node using random scheduling process, with the latest time stamp value from the Random scheduling table.
4. The bait detection is started initially such that the bait phase is activated whenever the bait RREQ′ is sent earlier for seeking the initial routing path.
5. From this initial bait setup the malicious node is detected from the route reply.
6. For this a reverse tracing setup is initiated. Here the trust set and a dubious information set is acquired from the nodes sending RREP for RREQ′. From this cooperative malicious nodes were found and their cooperation is stopped and the bait detection is continued for the remaining nodes.
7. Next the trust value is estimated from the random scheduling table, and the node with less trust value is removed immediately.

E. Simulation Model and Parameters

The Network Simulator (NS2) [16], is used to simulate the proposed architecture. In the simulation, 50 mobile nodes move in a 1000-meter x 1000-meter region for 50 seconds of simulation time. All nodes have the same transmission range of 250 meters. The simulated traffic is Constant Bit Rate (CBR).

The simulation settings and parameters are summarized in table.

Table 1
Simulation Model and Parameters

No. of Nodes	100 and 200
Area Size	1000 X 1000
Mac	IEEE 802.11
Transmission Range	250m
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	512
Rate	150Kb
Attackers	5,10,15,20 and 25

F. Performance Metrics

The proposed Distributed Trust Based Co-operative Bait Detection Scheme for Detecting Collaborative Attacks (DTCBDS) is compared with the BDS technique. The performance is evaluated mainly, according to the following metrics.

- *Packet Delivery Ratio:* It is the ratio between the number of packets received and the number of packets sent.
- *Packet Drop:* It refers the average number of packets dropped during the transmission
- *Delay:* It is the amount of time taken by the nodes to transmit the data packets.

4. Results

Scen-1(100 Nodes)
1) Based on Attackers

In our first experiment we vary the number of attackers as 5, 10, 15, 20 and 25 for 100 nodes scenario.

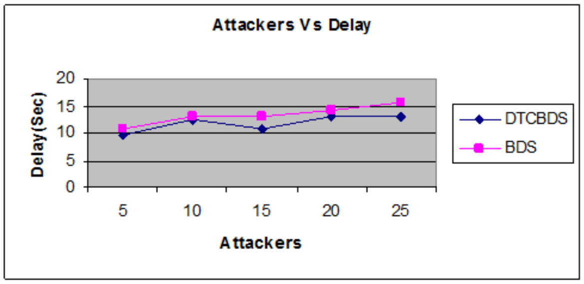
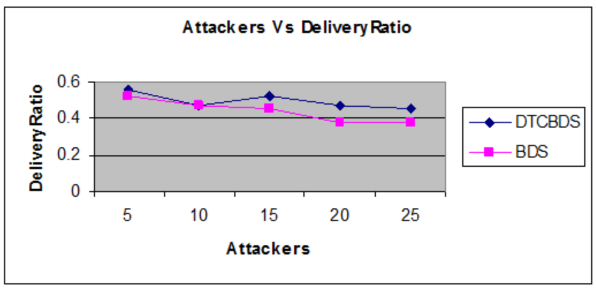


Fig. 5. Attackers vs. Delay



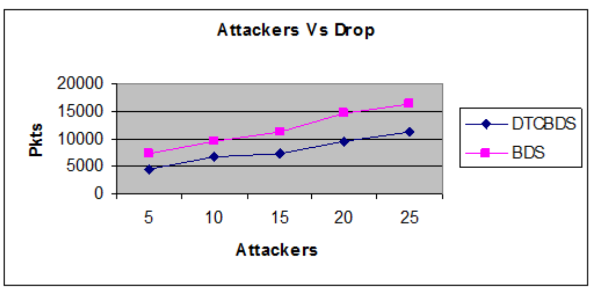


Fig. 7. Attackers vs. Drop

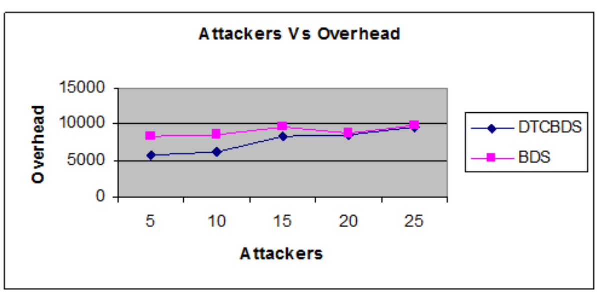


Fig. 8. Attackers vs. Overhead

Figure 5 shows the delay of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the delay of our proposed DTCBDS approach has 12% of less than BDS approach.

Figure 6 shows the delivery ratio of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the delivery ratio of our proposed DTCBDS approach has 11% of higher than BDS approach.

Figure 7 shows the drop of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the drop of our proposed DTCBDS approach has 33% of less than BDS approach.

Figure 8 shows the overhead of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the overhead of our proposed DTCBDS approach has 15% of less than BDS approach.

Scen-2(200 Nodes)
1) Based on Attackers

In our second experiment we vary the number of attackers as 5, 10, 15, 20 and 25 for 100 nodes scenario.

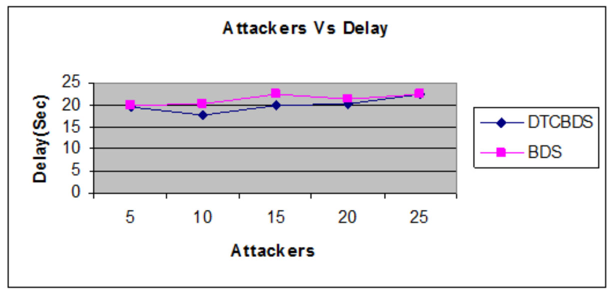


Fig. 9. Attackers vs. Delay

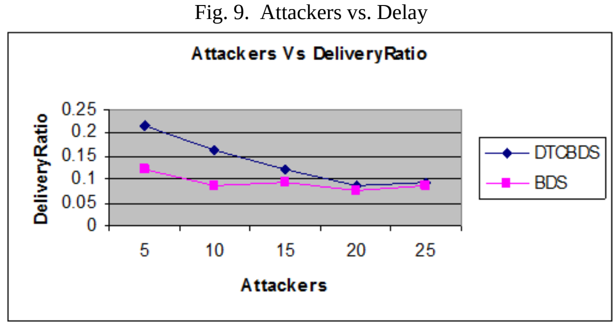


Fig. 10. Attackers vs. Delivery Ratio

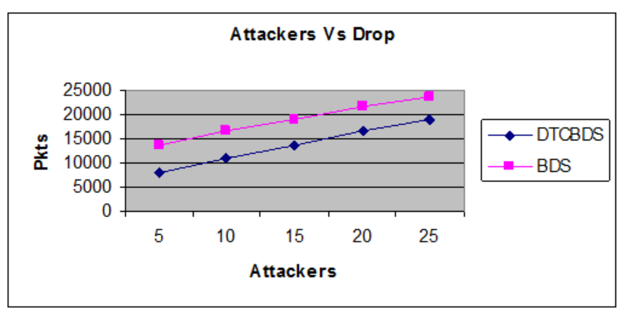


Fig. 11. Attackers vs. Drop

Figure 9 shows the delay of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the delay of our proposed DTCBDS approach has 6% of less than BDS approach.

Figure 10 shows the delivery ratio of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the delivery ratio of our proposed DTCBDS approach has 26% of higher than BDS approach.

Figure 11 shows the drop of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the drop of our proposed DTCBDS approach has 29% of less than BDS approach.

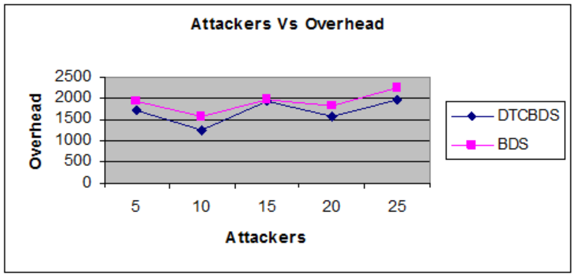


Fig. 12. Attackers vs. Overhead

Figure 12 shows the overhead of DTCBDS and BDS techniques for different number of attacker scenario. We can conclude that the overhead of our proposed DTCBDS approach has 12% of less than BDS approach.

5. Conclusion

In this paper, we have proposed a distributed trust based co-operative bait detection scheme for detecting collaborative attacks in MANET. Here using the trust value which estimated using Bayesian interference the Bait detection process is invoked. For this the source node selects an adjacent node using the random scheduling process. This is the address of this adjacent node is used as bait destination address to bait malicious nodes in order to send a reply RREP message. By this the bait detection is raised. From the random schedule table, the nodes with less trust value which is considered to be as un-trusted nodes are removed instantly.

References

[1] Meenakshi Patel and Sanjay Sharma, "Detection of Malicious Attack in MANET A Behavioral Approach", IEEE 3rd International Advance Computing Conference (IACC), pp. 388-393, 2012.
[2] Reshma Lill Mathew, P. Petchimuthu, "Detecting Selfish Nodes in MANETs Using Collaborative Watchdogs", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, Issue 3, March 2013.
[3] Umesh Kumar Singh, Kailash Phuleria, Shailja Sharma and D.N. Goswami, "An analysis of Security Attacks found in Mobile Ad-hoc Network", International Journal of Scientific & Engineering Research, vol. 5, Issue 5, May 2014.
[4] Tao Gong and Bharat Bhargava, "Immunizing mobile ad hoc networks against collaborative attacks using cooperative immune model", Security and Communication Networks, vol. 6, Issue 1, pp. 58–68, January 2013.
[5] Ajay Dureja and Vandna Dahiya, "Performance Evaluation of Collaborative Attacks in Manet", Journal of Computer Science and Information Technology, vol. 3, Issue. 7, pp. 457-465, July 2014.
[6] Cong Hoan Vu and Adeyinka Soneye, "An Analysis of Collaborative Attacks on Mobile Ad hoc networks", LAP LAMBERT Academic Publishing, June 2009.
[7] Bharat Bhargava, Ruy de Oliveira, Yu Zhang and Nwokedi C. Idika, "Addressing Collaborative Attacks and Defense in Ad Hoc Wireless Networks", 29th IEEE International Conference on Distributed Computing Systems Workshops, pp. 447-450, 2009.
[8] Mahdi Nouri, Somayeh Abazari Aghdam and Sajjad Abazari Aghdam, "Collaborative Techniques for Detecting Wormhole Attack in MANETs", International Conference onResearch and Innovation in Information Systems (ICRIIS), pp. 1-6, 2011.
[9] Jian-Ming Chang, Po-Chun Tsou, Isaac Woungang, Han-Chieh Chao, and Chin-Feng Lai, "Defending Against Collaborative Attacks by Malicious Nodes in MANETs: A Cooperative Bait Detection Approach", IEEE systems journal, vol. PP, Issue: 99, pp. 1-11, 2014.
[10] JaydipSen, M. Girish Chandra, P. Balamuralidhar, Harihara and S.G., Harish Reddy, "A Distributed Protocol for Detection of Packet Dropping Attack in Mobile Ad Hoc Networks", IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, pp. 75-80, 2007.
[11] Chang Wu Yu, Tung-Kuang Wu, ReiHeng Cheng and Shun Chao Chang, "A Distributed and Cooperative Black Hole Node Detection and Elimination Mechanism for Ad Hoc Networks", Emerging Technologies in Knowledge Discovery and Data Mining Lecture Notes in Computer Science, vol. 4819, pp. 538-549, 2007.
[12] Weichao Wang, Bharat Bhargava and Mark Linderman, "Defending against Collaborative Packet Drop Attacks on MANETs", 2nd International Workshop on Dependable Network Computing and Mobile Systems (DNCMS), vol. 27, 2009.
[13] Sukla Banerjee, "Detection / Removal of Cooperative Black and Gray Hole Attack in Mobile Ad-Hoc Networks", In proceedings of the world congress on engineering and computer science (WCECS), pp. 22-24, 2008.
[14] Zhexiong Wei, Helen Tang, F. Richard Yu, Maoyu Wang, and Peter Mason, "Security Enhancements for Mobile Ad Hoc Networks with Trust Management Using Uncertain Reasoning", IEEE Transactions on Vehicular Technology, 2014.
[15] https://www.eecis.udel.edu/~mills/time.html.
[16] Network Simulator: http:///www.isi.edu/nsnam/ns