Collaborative Alert in a Reputation System to Alleviate Colluding Packet Droppers in Mobile Ad Hoc Networks

K. Gopalakrishnan and V. Rhymend Uthariaraj

Ramanujan Computing Centre, College of Engineering Guindy,
Anna University Chennai, Chennai – 600 025, Tamil Nadu, India
mrkrishauc@yahoo.in, rhymend@annauniv.edu

Abstract. The nature of the ad hoc network seems to have a promising future in real world and vast researches are going on to make the network more secure in an open wireless environment. The misbehaving nodes in ad hoc network results in degradation of overall network throughput and creates difficulty in finding route between nodes. Thus the collaborative nature of ad hoc network seems to be endangered due to the presence of misbehaving nodes and even get worse when the misbehaving node colludes to misbehave. This paper addresses a colluding packet dropping misbehavior and proposes a collaborative alert mechanism in a reputation system to alleviate it. The simulation result shows that the proposed system increases overall network throughput, reduces the malicious drop and false detection when compared to existing system and defense less scenario.

Keywords: Routing Security, Collaborative Alert, Reputation System, Colluding Packet Droppers, Ad Hoc Networks.

1 Introduction

Mobile Ad Hoc Networks (MANETs) is a collection of mobile nodes that communicates with each other by not depending on the preexisting infrastructure and centralized administration. A routing protocol is used to discover correct and efficient route between a pair of nodes so that messages may be delivered in a timely manner. The lack of preexisting infrastructure makes each node in the network to function as routers which discover and maintain routes to other nodes in the network. Ad hoc network maximizes the total network throughput by using all available nodes for routing and forwarding. The more nodes that participate in the routing process results in greater the aggregate bandwidth, shorter routing paths and minimizes the network partition. The lack of centralized administration and the nature of ad hoc networks pose a threat to the routing process [11]. A node agrees to forward packets on behalf of other nodes during the route discovery phase but failed to do so due to malicious or non-malicious behavior [13]. The non-malicious packet dropping exists due to network congestion, mobility and node malfunction. When the malicious node

N. Meghanathan et al. (Eds.): CCSIT 2011, Part I, CCIS 131, pp. 135–146, 2011.

colludes to mischief then it further increases the complexity in discovering routes and also results in frequent network partitioning and performance degradation. This paper addresses the colluding packet dropping misbehavior and proposes a novel solution to alleviate it. The rest of the paper is organized as follows. The section 2 describes about the related works. The proposed work is described in section 3. In section 4 and 5 the simulation study and the results are discussed. Finally, section 6 concludes the work and insights about the future work.

2 Related Works

Marti et al. [5] proposed a scheme which contains two components namely *Watchdog and Pathrater* in each and every node to detect and mitigate the routing misbehavior in mobile ad hoc networks. The nodes operate in promiscuous mode where in the watchdog maintain a buffer of recently sent packets and compare each overheard packet with the packet in the buffer. If there is a match then the packet in the buffer is removed else if the packet remained in the buffer for longer than a certain time out then the watchdog increments a failure tally for the node responsible for forwarding the packet. If the tally exceeds certain threshold limit then the node is identified as misbehaving. The pathrater combines the knowledge of misbehaving nodes with link reliability data to pick a route that is most likely to be reliable. This approach does not punish misbehaving nodes that do not cooperate and also relieves them of the burden of forwarding packets for other nodes. Buchegger et al. [7] proposed a protocol called *CONFIDANT (Co-operation Of Nodes: Fairness In Dynamic Ad-hoc NeTwork)* based on selective altruism and utilitarianism to detect and isolate misbehaving nodes. The protocol adds a trust manager and a reputation system to the watchdog and pathrater scheme. The trust manager evaluates the events reported by the watchdog and the reputation system maintains a blacklist of nodes at each node and shares with all other nodes that are in their friends list. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. Michiardi et al. [8] proposed a mechanism called *CORE (COllaborative REputation mechanism)* to enforce node cooperation in mobile ad hoc network. CORE stimulates node cooperation based on collaborative monitoring and a reputation mechanism. The reputation metric is computed based on data monitored by the local entity and some information provided by the other nodes involved in each operation. The nodes with good reputation can utilize the network resources where as the node with bad reputation are gradually excluded from the network. Bansal et al. [10] proposed a reputation mechanism termed as *OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks)* based on direct observation experienced by a node from its neighbors. The routing decisions are based on direct observation of neighboring nodes behavior and it completely disallows the exchange of second hand reputation. Hu et al. [12] proposed a scheme called *LARS - A Locally Aware Reputation System* for Mobile Ad Hoc Networks for which the reputation of nodes is derived by using direct observation. When a selfish node is identified then its k-hop neighbors become aware of the selfishness, where k is a parameter which is adaptive to the security requirement of the network. In order to avoid false accusation and the associated trust issues, conviction of the selfish node is valid only if m different neighbors accuse, where m − 1 is an upper bound on the number of malicious nodes in the neighborhood. The success of this scheme relies on the critical selection of value for m.

3 Collaborative Alert in a Reputation System

The proposed *Collaborative Alert in a Reputation System (CARS)* consists of three main components a monitor to detect the packet dropping misbehavior, reputation system to maintain the trust value for the neighborhood nodes and a path manager to maintain the routes without containing packet droppers in it as shown in Fig. 1.

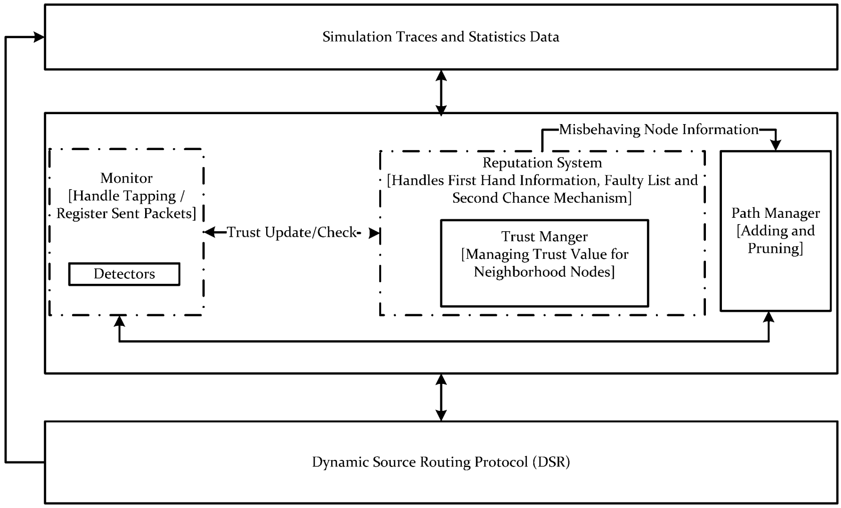

Simulation Traces and Statistics Data
Misbehaving Node Information
Monitor [Handle Tapping / Register Sent Packets]
Detectors
Trust Update/Check
Reputation System [Handles First Hand Information, Faulty List and Second Chance Mechanism]
Trust Manger [Managing Trust Value for Neighborhood Nodes]
Path Manager [Adding and Pruning]
Dynamic Source Routing Protocol (DSR)

Fig. 1. Functional Block Diagram of CARS

These components are added as an add-on into the existing routing functionality of *Dynamic Source Routing (DSR)* protocol. This enables each node in the network to execute this add-on functionality along with the usual routing protocol operations. Whenever a node overhears a packet from the neighboring node for the first time then the neighboring node information is stored in the *Neighbor Connectivity List (NCL)* along with the timestamp at which the packet is overheard and its trust value is initialized into 0. The timestamp and the trust value are updated for the subsequent packet overhearing from the neighboring node. The monitor component is responsible for tapping and registering the sent packets. It has an internal component called detectors which are used to identify the different kinds of packet dropping misbehavior as described in sub section

4.1. The reputation system handles the first hand information received from the monitor and maintains the trust value for the neighborhood nodes accordingly with the help of its internal component called trust manager. Once a nodes reputation value reaches the *Negative Threshold* limit then it will be added into the faulty list and any traffic to and from the misbehaving node will be rejected. As soon as a node is added into the faulty list a second chance timer will be initiated for that node.

The misbehaving node information is also communicated to the path manger in order to prune the routes which have the misbehaving link in it and also an explicit route error packet will be sent to the source of the packet to inform about the misbehaving link. Once the source or an intermediate node receives an explicit route error packet then it checks whether it is originated by the source of the misbehaving link or from the neighborhood of the misbehaving node. The route which contains the misbehaving link is pruned if the packet is originated by the source of the misbehaving link else routes containing the destination of the misbehaving link will be pruned from both primary and secondary route cache. When the second chance timer of the misbehaving node reaches $100s$ then the node is removed from the faulty list and reintroduced into the network by considering it to be useful again after reducing its trust value by half. The reason for not resetting the trust value of the reintroduced node to 0 is that the node might still continue to misbehave. If it continues to misbehave then it will be detected soon.

3.1 Packet Monitoring and Trust Evaluation of CARS

The procedure for packet monitoring and trust evaluation of the proposed system is shown in Fig. 2. Whenever a node receives a packet it checks the trust value of the transmitting node. If it is not equal to *Negative Threshold* then it further checks whether it is a destination or not else drops the packet.

It accepts the packet if it is a destination node else it checks whether it's a forwarding or an overhearing node. If it is a forwarding node it checks whether the receiving hop trust value is not equal to *Negative Threshold*, if so it forwards the packet to the next hop else drops the packet. Further it checks whether the forwarded packet is a *RREQ* or the receiving hop is a destination. If so the procedure ends else it waits to overhear the same packet forwarded by the receiving hop. The trust value of the receiving hop is incremented by 1 if it overhears the packet else it waits until the packet timeout period to overhear the *RERR* packet. If it does not overhear the *RERR* packet then the trust value of the receiving hop is decremented by 2 else the procedure ends. On the other hand, if it is an overhearing node then it checks whether the receiving hop is active in its *NCL* and its trust value is not equal to *Negative Threshold.* If so it checks the relation between transmitting and receiving hop else the procedure ends. The procedure ends if the relation of receiving hop is a destination else it waits to overhear the same packet from the receiving hop. If it overhears the packet then the trust value of the receiving hop is incremented by 1 else it waits for the *RERR* packet. The procedure ends if it overhears the *RERR*

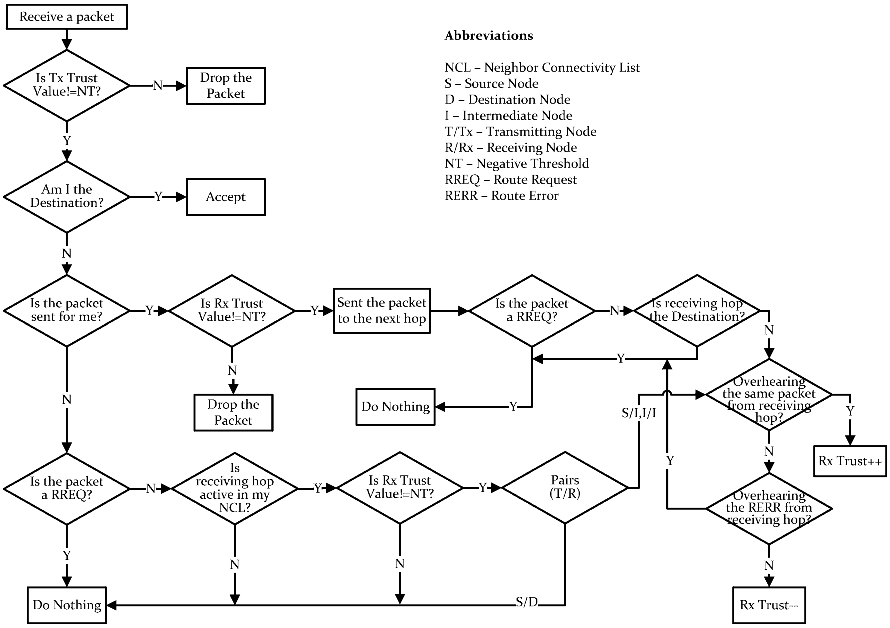


Fig. 2. Packet Monitoring and Trust Evaluation of CARS

packet within packet timeout period else the trust value of the receiving hop is decremented by 2. The neighboring node is considered to be active if any kind of packet is already overheard from it within 3000*ms* at the time of checking. The *RREQ* packet is not monitored because it can be dropped due to network operations [9].

3.2 Colluding Packet Dropping Misbehavior

As shown in Fig. 3, the solid circle represents a node and a solid line between them shows that the nodes are within the communication range of each other. Assume that the source node S communicates with the destination node D via the intermediate nodes $I_1 \rightarrow I_2 \rightarrow I_3$ and the nodes I_2 and I_3 colludes to misbehave. If I_3 drops the packet then the previous hop I_2 will not monitor and report to the source of the packet about this spiteful behavior because it colludes to mischief with node I_3.

In this scenario the neighboring nodes N_1 and N_2 are within the transmission range of I_3 so it can identify spiteful behavior and report to the source of the packet about this misbehaving link. Once the trust value of the misbehaving node I_3 reaches the *Negative Threshold* in N_1 or N_2 then node I_3 was added into their faulty list and also an explicit route error packet has been generated and sent to the source of the packet to inform about this misbehaving link $I_2 \rightarrow I_3$. The source route of an explicit route error packet should not contain node I_2 in it because it colludes to misbehave with I_3. Once the source or an intermediate node receives

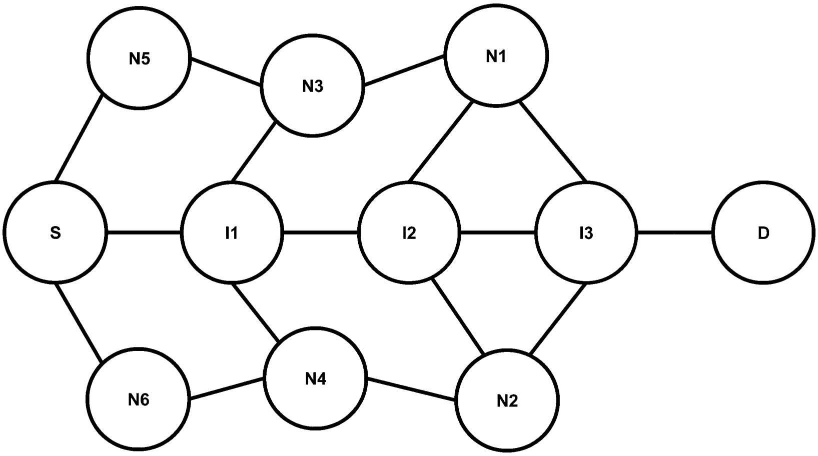


Fig. 3. Colluding Packet Dropping Misbehavior

an explicit route error packet then it will prune the routes from both the primary and secondary cache which have the misbehaving node I_3 in it.

4 Simulation Study

The proposed system was implemented in *ns-2.34* as an add-on to the *DSR* routing protocol. The *DSR* is an on demand source routing protocol which supports for promiscuous listening in order to overhear the neighboring nodes transmission and updates its trust value accordingly. Further, the source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use [2]. In simulation two different kinds of mobility models were used to mimic the real world movement of the mobile nodes and evaluated the performance of the proposed system in each of them separately. The first one is a *Random Waypoint (RWP)* mobility model based on *Entity (E)* mobility model in which the mobile node movements are independent of each other and the other one is a *Reference Point Group Mobility (RPGM)* Model based on *Group (G)* mobility model in which the mobile nodes move as a group [4]. The Random Waypoint mobility model is based on *CMU Monarch v2* implementation and Reference Point Group Mobility model is based on the implementation of [6].

There exists multiple group of mobile nodes and each group work towards different goal but there exists a communication between groups as described in [1], [3] so the group mobility model utilizes both inter and intra group *CBR* traffic patterns to evaluate the proposed system under the group mobility scenario. Each node is assigned an initial value of *Energy (E)* by using an uniform distribution function in the interval $(E_i–3J, E_i + 3J)$ where the energy is expressed

Table 1. Simulation Parameters

Parameter	Value
Simulation Area	900 m x 900 m
Simulation Time	900 s
Number of Nodes	50
Number of Groups	5
Nodes Per Group	10
Reference Point Separation	100
Node Separation from Reference Point	50
Propagation Model	Two-Ray Ground Reflection
Antenna	Omni Directional Antenna
RXThresh_	3.65262e-10
Packet Timeout	0.5 s
Transmission Range	250 m
Transmission Power	0.281838W
Reception Power	0.281838W
Traffic Type	CBR (UDP)
Maximum Connections	15
Payload Size	512 bytes
Seed Value	1-20
Negative Threshold	-40
Positive Threshold	40

in *Joules (J)* and the initial energy $E_i = 500J$. The consequence of this choice is that every node will run out of energy at different times in the simulation, which adds a degree of randomness to the simulation for evaluating the performance of the proposed scheme. The simulation parameters that were used in the simulation are shown in Table 1.

4.1 Modeling the Misbehavior

The proposed system was simulated by introducing three different kinds of packet dropping misbehavior as listed below

1. Packet Dropping Type 1 - These nodes participate in the *DSR* Route Discovery and Route Maintenance phases, but refuse to forward data packets on behalf of other nodes (which are usually much larger than the routing control packets)
2. Packet Dropping Type 2 - These nodes participate in neither the Route Discovery phase, nor forwarding data packets. They only use their energy for their own packet transmission
3. Packet Dropping Type 3 - These nodes behave differently based on their energy levels. When the energy lies between full energy E and a threshold T_1, the node behaves properly. On the other hand if an energy level lies between T_1 and another lower threshold T_2 then it behaves like a node of

Packet Dropping Type 1. Finally, for an energy level lower than T_2, it behaves like a node of Packet Dropping Type 2. The relationship between T_1, T_2 and E is $T_2 < T_1 < E$

The node behavior has been added as a node definition type in the *ns2* node model. The syntax that is used to define the node configuration has been enhanced with a new optional feature that allows selecting the packet dropping model among three possible choices. It is also necessary to modify the *DSR* routing protocol implemented in *ns2* because the networking functions (routing and packet forwarding) are overridden by the routing protocol selected in the node configuration. The modified version of the *DSR* routing protocol checks the current node configuration and depending on the packet drop model used for that node, it decides whether to execute the networking functions or not.

4.2 Performance Metrics

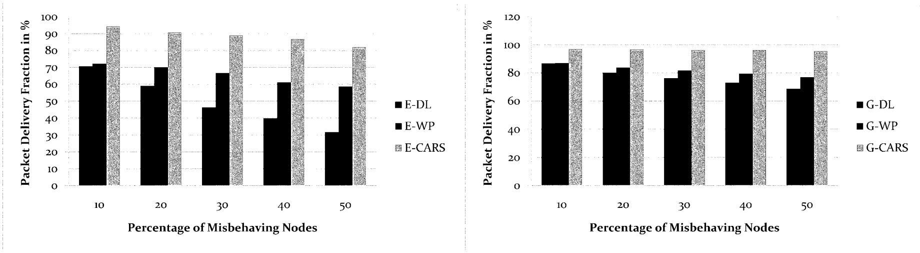
The performance of the proposed system has been measured by using the following parameters

1. Packet Delivery Fraction (%) - The packet delivery fraction is measured in terms of the ratio of the data packets delivered to the destinations to those generated by the *Constant Bit Rate (CBR)* sources
2. Normalized Routing Load (Packets) - The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission
3. Average End-End Delay (Seconds) - This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the *MAC*, propagation and transfer times
4. Average Energy Dissipation (Joules) - The average amount of network energy dissipated over the simulation period
5. Malicious Drop (Packets) - The total number of packets dropped by the different kind of packet droppers
6. False Detection (%) - The percentage of nodes detected falsely as a misbehaving node over the simulation period

The measurements of the network performance were made using a script that parses and analyzes the trace file output generated from the simulation. The trace file provides information about a set of defined events that occurred in the simulation such as medium access control layer events, routing layer events and agent level events.

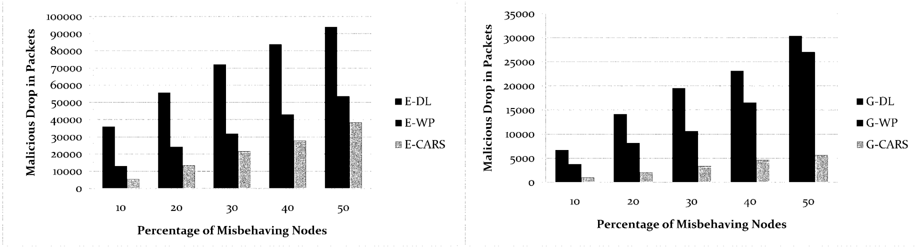
5 Results and Discussions

The simulation results of the proposed system were compared with *Defense Less (DL)* scenario and the existing scheme *Watchdog Pathrater (WP)* [5]. This paper calculates a 95% confidence interval for the unknown mean and plots the



(a) Entity Mobility Scenario (b) Group Mobility Scenario

Fig. 4. Packet Delivery Fraction in %

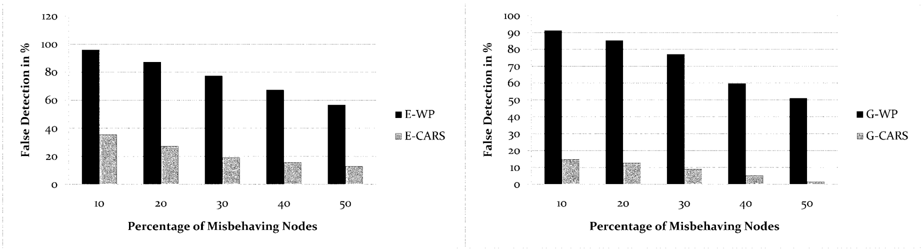


(a) Entity Mobility Scenario (b) Group Mobility Scenario

Fig. 5. Malicious Drop in Packets

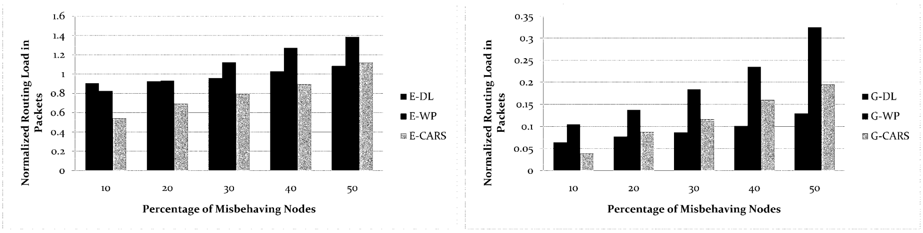
confidence intervals on the graphs.The packet delivery fraction of *CARS* was increased by 24-51% and 11-27% when compared to *DL* scenario, 21-26% and 11-19% when compared to WP scheme under both entity and group mobility scenario respectively as shown in a,b of Fig. 4.

As shown in a,b of Fig. 5, the malicious drop of *CARS* has been decreased by 60-85% and 82-86% when compared to *DL* scenario, 29-58% and 68-79% when compared to *WP* scheme with respect to both entity and group mobility scenario respectively. The false detection of *CARS* was decreased from 44-61% and 50-77% when compared to *WP* scheme under both entity and group mobility scenario respectively as shown in a,b of Fig. 6. As shown in a,b of Fig. 7, the normalized routing load of *CARS* has been decreased by 20-35% and 32-63% when compared to the *WP* scheme under entity and group mobility scenario respectively. Since the average energy dissipation is directly proportional to the overall network throughput. As shown in a,b of Fig. 8, the average energy dissipation of *CARS* was increased by 6-9% and 5-6% when compared to *WP* scheme under entity and group mobility scenario respectively. The average end-end delay has been decreased by 3-46% and 14-70% when compared to *WP* scheme under entity and group mobility scenario respectively as shown in a,b of Fig. 9. The result shows that the proposed system performs better when compared to



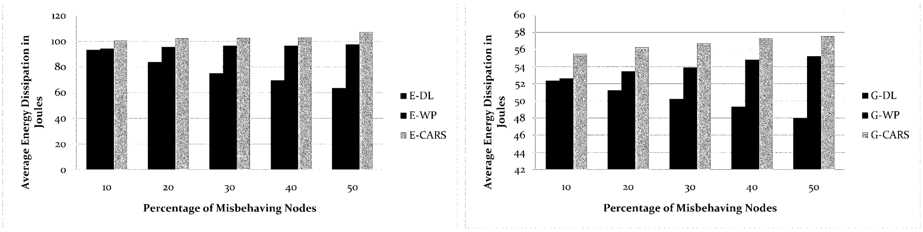
(a) Entity Mobility Scenario (b) Group Mobility Scenario

Fig. 6. False Detection in %



(a) Entity Mobility Scenario (b) Group Mobility Scenario

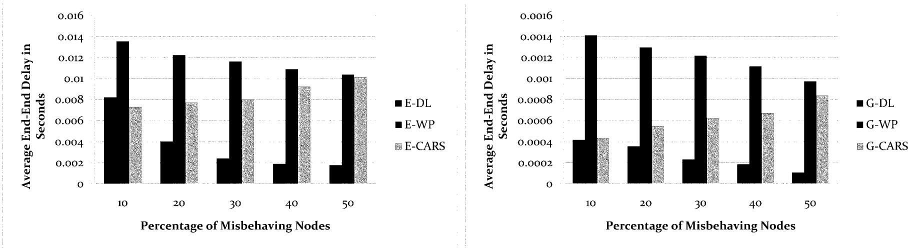
Fig. 7. Normalized Routing Load in Packets



(a) Entity Mobility Model (b) Group Mobility Model

Fig. 8. Average Energy Dissipation in Joules

the Watchdog Pathrater scheme. The timely generation of an explicit route error generation reduces the false detection and in turn increases the overall network throughput. The strong malicious traffic rejection results in reduction in control overhead and end-end delay of per packet delivered to the destination.



(a) Entity Mobility Scenario

(b) Group Mobility Scenario

Fig. 9. Average End-End Delay in Seconds

6 Conclusion and Future Work

The simulation results of the proposed system shows that the overall network throughput has been greatly improved and the percentage of malicious packet drop also reduced due to efficient detection and isolation of packet droppers. Further, the timely generation of an explicit route error packet to inform the source of the packet about the misbehaving link combined with the forward traffic rejection reduces the percentage of false detection when compared to the existing scheme. The proposed system was immune to colluding packet dropping misbehavior because of an explicit route error generated by the neighboring nodes and also other kind of overhearing technique drawbacks due to the efficient monitoring and detection of neighboring nodes. In future work, more kind of misbehaving nodes will be considered and also the faulty list will be shared with the rest of the nodes in the network in order to mitigate the misbehaving nodes without incurring additional control overhead.

Acknowledgment. The first author would like to acknowledge the support received from *CSIR-HRDG (Council of Scientific and Industrial Research-Human Resource Development Group)*, India, through Senior Research Fellowship.

References

1. Hong, X., Gerla, M., Pei, G., Chiang, C.: A Group Mobility Model for Ad Hoc Wireless Networks. In: 2nd ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems, pp. 53–60. ACM, Seattle (1999)
2. Johnson, D.B., Maltz, D.A., Broch, J.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet Draft, The Internet Engineering Task Force (1999)
3. Liang, B., Haas, Z.: Predictive Distance-Based Mobility Management for PCS Networks. In: 18th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1377–1384. IEEE Computer Society, New York (1999)

4. Hong, X., Gerla, M., Pei, G., Chiang, C.: A Wireless Hierarchical Routing Protocol with Group Mobility. In: IEEE Wireless Communications and Networking Conference, pp. 1536–1540. IEEE, New Orleans (1999)
5. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: 6th International Conference on Mobile Computing and Networking, pp. 255–265. ACM, Boston (2000)
6. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. J. Wireless Communication and Mobile Computing 2, 483–502 (2002)
7. Buchegger, S., Le Boudec, J.Y.: Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks). In: IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, pp. 226–236. ACM, Lausanne (2002)
8. Michiardi, P., Molva, R.: CORE: A COllaborative REputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks. In: 6th Joint Working Conference on Communications and Multimedia Security, vol. 228, pp. 107–121. Kluwer, Portoroz (2002)
9. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The Broadcast Storm Problem in a Mobile Ad Hoc Network. J. Wireless Networks 8, 153–167 (2002)
10. Bansal, S., Baker, M.: Observation-based Cooperation Enforcement in Ad hoc Networks. Technical Report, Stanford University (2003)
11. Yau, P., Mitchell, C.J.: Security Vulnerabilities in Ad Hoc Networks. In: The Seventh International Symposium on Communication Theory and Applications, pp. 99–104. HW Communications Ltd, Ambleside (2003)
12. Hu, J., Burmester, M.: LARS A Locally Aware Reputation System for Mobile Ad Hoc Networks. In: 44th Annual Southeast Regional Conference, pp. 119-123. ACM, Melbourne (2006)
13. Gopalakrishnan, K., Rhymend Uthariaraj, V.: Scenario based Evaluation of the Impact of Misbehaving Nodes in Mobile Ad Hoc Networks. In: 1st IEEE International Conference on Advanced Computing, pp. 45–50. IEEE Computer Society, Chennai (2009)