Pertanika J. Sci. & Technol. 26 (3): 1513 - 1526 (2018)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Cheat-Proof Communication through Cluster Head (C3H) in Mobile Ad Hoc Network

Abu Sufian[1]*, Anuradha Banerjee[2] and Paramartha Dutta[3]

[1]Department of Computer Science, University of Gour Banga, West Bengal, India
[2]Department of Computer Application, Kalyani Government Engineering College, West Bengal, India
[3]Department of Computer and System Sciences, Visva-Bharati University, West Bengal, India

ABSTRACT

The mobile ad hoc network (MANET) is a wireless network based on a group of mobile nodes without any centralised infrastructure. In civilian data communication, all nodes cannot be homogeneous-type and not do a specific data communication. Therefore, node co-operation and cheat-proof are essential characteristics for successfully running MANETs in civilian data communication. Denial of service and malicious behaviour of the node are the main concerns in securing successful communication in MANETs. This scheme proposed a generic solution to preventing malicious behaviour of the node by the cluster head through the single hop node clustering strategy.

Keywords: Ad hoc network, black-hole attack, cheat-proof, malicious node, MANET

INTRODUCTION

Mobile ad-hoc networks (MANETs) is an infrastructure-less, self-organising network in which a set of mobile nodes (capable of receiving and transmitting radio signals) can quickly set up a temporary network (Toh, 2002). This type of network is very useful in emergency situations such as a battle field or a rescue operation after a natural disaster, in commercial applications like vehicular ad hoc networks and in communication in conference halls, among other instances (Helen & Arivazhagan, 2014). There are many underlined protocols that are available to establish this type of network (Abolhasan, Wysocki, & Dutkiewicz, 2004). We can classify all these protocols into three broad categories: (a) proactive or table driven such as DSDV (Perkins & Bhagwat, 1994), WRP (Murthy & Aceves, 1996) and FSR (Pei, Gerla, & Chen, 2000); (b) reactive, such as AODV (Perkins & Royer, 1998), AOMDV

ARTICLE INFO

Article history:
Received: 20 November 2017
Accepted: 28 June 2018

E-mail addresses:
sufian.csa@gmail.com (Abu Sufian)
anuradha79bn@gmail.com (Anuradha Banerjee)
paramartha.dutta@gmail.com (Paramartha Dutta)
*Corresponding Author

ISSN: 0128-7680

(Marina & Das, 2001) and DSR (Johnson, Hu, & Maltz, 2007); and (c) hybrid, where some part of the network is proactive and another is reactive, such as EMR-PL (Banerjee, Sufian, & Duta, 2018) and TORA (Park & Corson, 2001).

In most routing protocol network security and lack of co-operation among nodes are yet to be solved. It is true that MANETs is not effective in civilian data communication because of lack of co-operation and the malicious behaviour of some nodes, while in specific communication such as military data communication, it is effective. Attacks by some malicious nodes from inside the network are a main security issue in MANETs. This malicious behaviour is not an issue of military communication or any other specific communication because all nodes of such type of communication are of the same type and they work specifically in that communication. But in general data communication and networking, mobile nodes are open and the types are different such as communication made through mobile phones, laptops, palmtops, PDAs, among others. Therefore, network security is very much important.

In our scheme of cheat-proof communication through the cluster head (C3H), we used single hop-clustering strategies as a generic mechanism to increase network security through the cluster head (CH) so that MANETs can be successful in civilian data communication. We know that the clustering scheme is more scalable and we can see it in FESC (Banerjee, Dutta, & Sufian, 2018). In single hop-clustering, all the nodes are attached to CH directly and turns the entire network into different partitions called clusters. CHs are connected to each other through gateway nodes and establish a MANETs. The CHs take responsibility for network security and increase co-operation within networks. Network security is a big challenge from the start for MANETs (Deng, Li, & Agrawal, 2002). All the network security and co-operation attacks in MANETs can be classified into two broad categories: selfish attack and malicious attack. A malicious attack is more harmful than a selfish attack in this type of network. This scheme explains step by step the idea behind prevention, such as malicious attacks.

The rest of the paper is organised as follows: In the next section similar works are studied, while the third section explains the clustering strategy. Our proposed solution to the problem of malicious attacks is outlined in the fourth section, followed by a discussion on the simulation results, and in the final section, the conclusion and future scope are given.

LITERATURE REVIEW

Many node co-operation and cheat-proof schemes have been proposed. Marti, Giuli, Lai and Baker (2000) suggested a routing scheme based on DSR to detect misbehaving nodes using a 'watchdog' and providing labels using a 'pathrater'. Nodes are classified using this 'pathrater' so that misbehaving or malicious nodes can be avoided. Levente Buttyan and Jean-Pierre Hubaux proposed a virtual-currency-based scheme called 'Nuglet' (Buttyan & Hubaux, 2001) to increase node co-operation in MANETs. The researchers of this scheme used two purse models; one was the Packet Purse Model (PPM), where Nuglet is debited from the source of the packet and the other was the Packet Trade Model (PTM), where Nuglet is debited from the source or destination of the packet. This scheme also described the purpose of increasing Nuglet for a node and discussed the security of these Nuglet.

Michiardi and Molva (2002) proposed a reputation-based scheme called CORE. This scheme stimulates the selfish node to avoid selfish behaviour such as denial of service attack. Zhong, Chen and Yang (2003) proposed a simple, cheat-proof, credit-based system for mobile ad-hoc networks with selfish nodes called SPRITE. This is an incentive credit- or debit-based system without any tamper-proof hardware. Here nodes could get inceptive by showing receipt of forwarded messages from Credit Clearance Service (CCS).

Kargl et al. proposed the Advanced Detection of Selfish or Malicious Nodes in Ad Hoc Networks (Kargl, Klenk, Schlott, & Weber, 2014). This scheme explained activity-based overhearing, iterative probing and unambiguous probing to detect malicious and selfish nodes in the network. Nasser and Chen (2007) described an intrusion-detection scheme. Here, malicious nodes detected by overhearing the network then responding. This is an enhanced version of 'watchdog' and 'pathrater'.

Kang, Shakshuki and Sheltami (2010) presented a misbehaving node detection scheme at IIWAS in 2010. They used a different Intrusion Detection System (IDS) as watchdog to detect malicious nodes. The IDS, called Enhanced Adaptive ACKnowledgement (EAACK), attempted to overcome difficulties faced by the watchdog. On their part, Enrique Hernandez-Orallo et al. proposed cocoa as a collaborative-contact-based 'watchdog' (Orallo, Olmos, Cano, Calafate, & Manzoni, 2015) to effectively detect selfish nodes speedily. This scheme was said to depend on a 'watchdog', whereas CoCoWa used collaborative work based on the diffusion of local awareness of selfish nodes.

Chang et al. proposed the Cooperative Bait Detection Approach (CBDS) (Chang, Tsou, Woungang, Chao, & Lai, 2015) based on DSR. This scheme exploits both proactive and reactive defence architectures. The authors used the reverse tracing approach to defend a collaborative attack by malicious nodes. Berri et al. (2017) presented a reputation-based node cooperation model at an international conference. According to their scheme, co-operation between nodes can be increased by adding or deducting the reputation of nodes in the network. If a node denied service to a reputed node, the node with the greater reputation would 'lose'. A node could gain more reputation by serving a reputed node and less for serving a non-reputed one.

CLUSTERING SCHEME DETAILS

The main problem of MANETs is the mobility of nodes and for that reason, topology is highly dynamic. Due to this high mobility traditional protocols and the security scheme of fixed networks do not work in ad hoc networks. A clustering strategy can mimic the topology of a traditional network and reduce scalability; not only is this essential for MANETs, it can also reduce other problematic issues in MANETs. Here, we have adopted FESC (Banerjee, Dutta, & Sufian, 2018), a single hop-clustering scheme with some modifications for cheat-proof and co-operation among nodes in MANETs.

There are three types of nodes in this scheme namely, Cluster Head (CH), gateway node and ordinary member. The single hop-clustering scheme is a strategy where all the mobile nodes are attached to some elected CHs directly, making the entire network of many groups

of nodes headed by each CH. CHs are elected temporarily according to high residual energy, bandwidth and low mobility of node compared with other nodes of a cluster. A portion of an instance of our clustering scheme is shown in Figure 1.

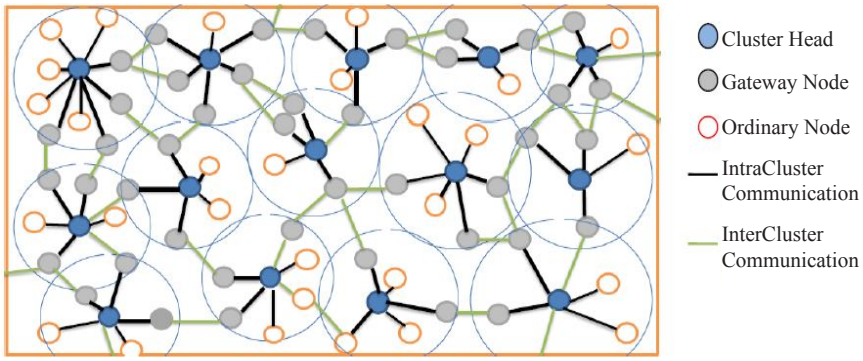

Figure 1. An instance of our clustering

Cluster Formation Strategy

The most important strategy in cluster schemes is the election of a Cluster Head (CH). The stability of the routing directly depends on stability of CHs. In this scheme, we assume CHs are fully supportive and trusted nodes; therefore, it is very important to choose a good candidate for CH. This is a single hop clustering scheme and another important strategy involving adding a node to a cluster, deleting a node from the cluster and merging two clusters to form a new cluster as done in FESC.

Electing Cluster Head (CH)

Four important factors of node were measured and combined with the help of fuzzy logic to get the final metric. This final metric was used to elect the CH and gateway node. The four important factors or metrics were: residual energy, trust value, mobility of a node and connectivity to downlink neighbours.

Residual energy of node. According to the functionality of node, at least 40% of residual energy is required to remain operational.

Let *Ei* be the total residual energy of node *ni*; *Ei* the expanded energy till current time; and *rem_eng(i)* the current residual energy as a fuzzy variable with values between 0 and 1. Therefore, the current residual energy will be represented by Equation 1:

$$res\_eng(i) = 1 - \frac{e_i}{E_i} \quad (1)$$

A value of '*res_eng*' below 0.4 means 'worst result' and one that is close to 1 means 'best result'.

Trust value of node. The initial stage is when a node enters the network; here, 0.5 is assigned as the default trust value, where '*trust_value*' is a fuzzy variable and the values 0 to 1 indicate the trusted level of a node. There are two more supporting variables: '*earn_trust*', which can take any natural number starting from 2, and '*loose_trust*', which is also a natural number ranging from 1 up to '*earn_trust*'. Initial values assigned to '*earn_trust*' and '*loose_trust*' are 2 and 1, respectively. When a node successfully transfers a packet its '*earn_trust*' value increases by 1. If any kind of selfish behaviour is shown its '*loose_trust*' value decreases by 1 and if any malicious behaviour is shown, it loses all its '*trust_value*', and its overall trust values decreases to 0. Trust value will also decrease after successful completion of the data transfer request by one unit. A node can ask the CH to transfer its data packet only if it has a positive trust value. The entire activity is carried out by CH. Equation 2 is used to calculate the current '*trust_value*' of a node.

$$trust\_value = 1 - \frac{loose\_trust}{earn\_trust} \tag{2}$$

More trust means more chances to become a CH as well as more credit to send a data packet.

Mobility of node. In order to stabilise the clustering scheme in MANETs, the CH of each cluster should be less mobile with its downlink neighbours compared with other nodes of the same cluster.

Let transmission power of a signal of node na be *trans_power(a)* and power of this signal when it being received at node n_i be $recv\_power_b(a)$, while the current distance between node n_a and n_b at the time of the *i*-th HELLO message is $dist_i(a,b)$. Therefore, as per Frii's transmission, Equation 3 is:

$$recv\_power_b(a) = K.\ trans\_power(a)/\ dist_i^q(a,b) \tag{3}$$

where, K is constant and q is a factor with the values 2, 3 or 4, depending upon the environment. Re-writing Equation 3, we get:

$$dist_i(a,b) = \sqrt[q]{\frac{K.trans\_power(a)}{recv\_power(a)}} \tag{4}$$

Suppose *t* is the time interval between two consecutive HELLO messages and *n* is the number of HELLO messages observed. Therefore, the effective mobility of node n_a compared to its downlink neighbours is calculated using Equation 5. The average mobility called '*avg_mobility*' of a node n_a with respect to all its downlink neighbours is given in Equation 6.

$$mobility_b(a) = \frac{\sum_{i=2}^{n}(dist_i(a,b) - dist_{i-1}(a,b))}{(n \times t)} \tag{5}$$

$$avg\_mobility(a) = \frac{\sum_{k=1}^{n_d} mobility(a)}{k} \tag{6}$$

Downlink neighbours connectivity. The CH should have more downlink neighbours compared with other member nodes. Here, we assume that the current CH has the standard number of downlink neighbours. This number of downlink neighbours of CH calculated by the fuzzy membership value of that CH and initial standard membership value of this parameter is 0.5. Any node that has more downlink neighbours has a greater chance of becoming a CH.

Let the number of downlink neighbours of the current CH be '*ndnb_CH*' and of the node ni be '$ndnb\_n_i$'. At first, the range of number of downlink neighbours needs to be fixed according to the current standard, as given in Equation 7.

$$\text{.....} dnb\_n_i = \begin{cases} ndnb\_n_i, & ndnb\_n_i \quad ndnb\_ \\ ndnb\_ \quad , & \text{otherwise} \end{cases}$$

$$\text{Therefore, } dn = \frac{ndnb\_n_i}{2 \times ndnb\_CH} \tag{7}$$

where, '$ednb\_n_i$' is the effective downlink neighbour and '*dnc'* is the downlink neighbour connectivity; clearly the range of '*dnc*' is from 0 to 1. It is also assumed that '*ndnb_CH*' will never be zero as before it becomes zero, the CH will be changed.

SEVERAL SECURITY ISSUES IN MANETS AND OUR PROPOSED SOLUTION

Besides other challenges such as dynamic topology, energy constrained and lack of bandwidth, MANETs faces two more serious challenges, which are selfish and malicious behaviour of the node. These challenges may come from some node(s) within the network. According to behaviour of the nodes, all nodes of the network can be classified into three categories: normal node, selfish node and malicious node. The normal node works in the expected way and is therefore not a concern. The selfish node works in unexpected ways, whereas the malicious node does more harm to the network. This scheme proposes a solution to security threats that come from malicious node(s) within the network. A malicious node raises the main security challenges in MANETs. Different types of malicious attacks and our proposed solution are discussed below.

Black-Hole Attack

Malicious nodes can participate in communication of other nodes by making a false route reply (RREP) packet mentioning the shortest path to the intended destination. The source node could fall into this trap of the malicious node and start sending data packets through this malicious node. Therefore, malicious nodes will be able to drop those packets or perform other more harmful work such as tampering with the data packets etc.

Figure 2 shows a portion of MANETs, where the malicious node is *m*, the source node, *s* and the destination node is *d*. The source node, *s*, wants to communicate with the destination node, *d*. Therefore, node *s*, will broadcast the route request (RREQ) packet to its neighbours,

including p, and p will also broadcast this RREQ to its neighbour nodes in the same way. In this way, the malicious node, *m*, gets the RREQ packet meant for the destination node, *d*, and node *m* provides a route reply (RREP) packet containing false information saying it has the shortest path to the destination *d*. The source node, *s*, believes the malicious node, *m*, and starts sending data packets to *m*. Now *m* can drop those packets or do more harmful work to them. Similarly, reverse-direction packets can also be captured by malicious nodes.

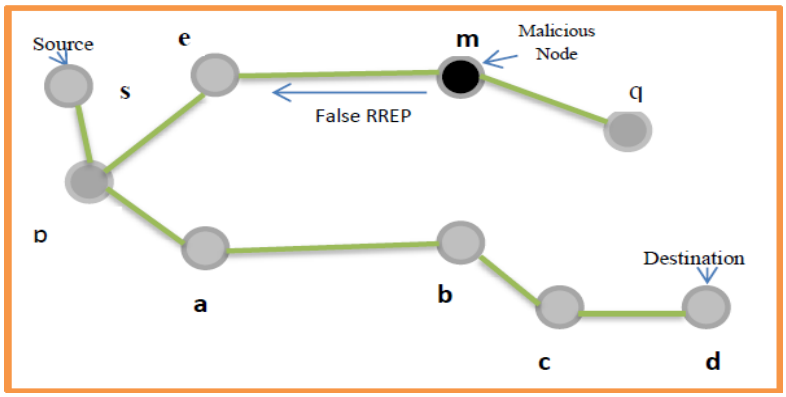

Figure 2. Black-hole attack by node *m*

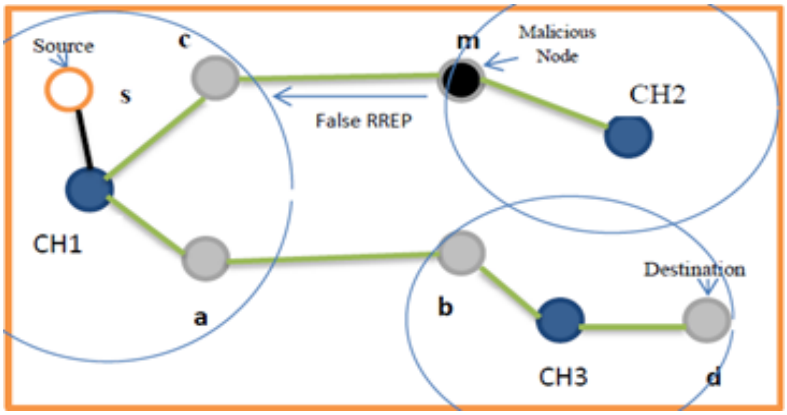

Figure 3. Solution to the black-hole attack

This black-hole attack can be prevented through the use of the CH. The CH can exert 'punishment' on malicious nodes and separate them from the network. As discussed earlier, in the clustering scheme, communication is performed through the CH, gateway node, source node and destination node. In this scheme, the black-hole attack can arise from the gateway node only and between two neighbouring CHs; at the most, two gateway nodes can participate in a communication path. The gateway node is one hop away from the CH. Therefore, activity can be monitored by the CH easily. If any gateway nodes carry out a black-hole attack, that is, when its first false-information reply enters the communication path but the node starts dropping packets, it can be easily caught red-handed from consultation with the neighbouring CH. The malicious node can then be separated and taken out of the cluster as well as the network itself.

In the example mentioned in the Figure 2, malicious node *m* can be bound by two successive CHs, CH1 and CH2, as shown in Figure 3. Between these two CHs gateway nodes, *c* and *m*, can only assist communication between these two cluster heads, CH1 and CH2. Here, node *m* is directly monitored by the cluster head, CH1; therefore, node *m* has no chance to carry out the black-hole attack as all communication is controlled only by CHs in our scheme. Therefore, here, this particular required communication can be done by CH1 and CH3.

Wormhole Attack

Here, two successive malicious nodes collude and make a wormhole between them. Whenever a route request arrives, colluding nodes hide their node information, so the source node does not perceive the presence of the two colluding nodes. The source node estimates the path length, which is less than two hops away from the actual path length. Therefore, the probability of selection of this path is very high. If this path is selected, then the source node will start sending the data packet through this path, and the two malicious nodes can drop the packets or do more harmful work such as tampering.

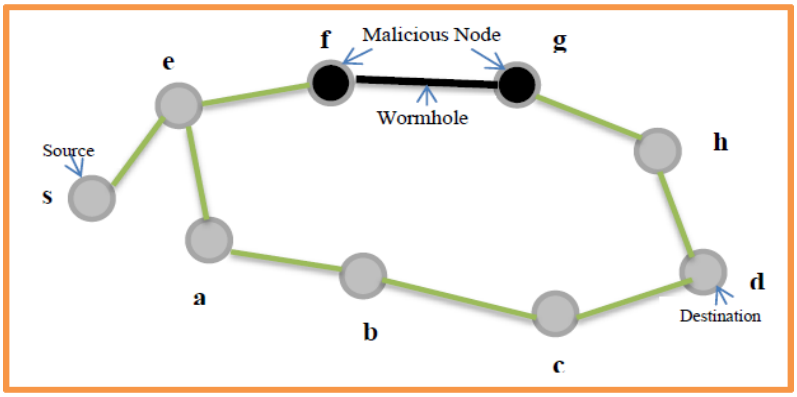

Figure 4. Wormhole attack by nodes, *f* and *g*

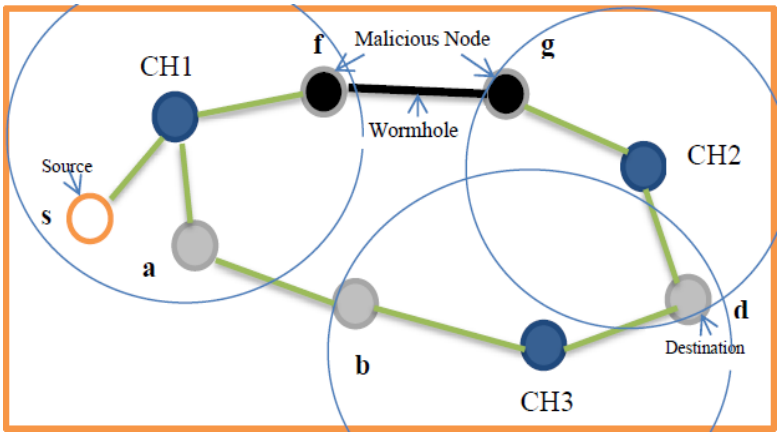

Figure 5. Solution to the wormhole attack

In Figure 4, in the portion of the network shown, source node, *s*, wants to communicate with destination node, *d*. Source nodes broadcast their RREQ packet to their neighbours, including *e*. When the RREQ is broadcast, node *f* will receive it, and it passes through the wormhole to node *g*. Node *g* sends it to node *h* without laying any marks on it. Therefore, node *h* understands that this route request packet comes from node *e* directly, but this was not the case; in this way, a virtual shortest path *s-e-h-d* is established but the actual path is *s-e-f-g-h-d*. Now, source node *s* selects this virtual path *s-e-h-d* instead of the real shortest path, *s-a-b-c-d*. When source node *s* starts sending data packets through this shortest path, the malicious node is able to capture those data packets. Reverse-direction data packets also can be captured in the same way.

As mentioned earlier, at the most, two gateway nodes could be between two successive CHs. In a wormhole attack, two successive nodes collude to make a wormhole between them; therefore, a wormhole attack is not feasible in a single hop clustering scheme because two gateway nodes are directly monitored by the CH. This is the advantage of the single hop clustering scheme. Even activity initiated by gateway nodes is easily caught by two successive CHs. In the example illustrated in Figure 4, malicious nodes *f* and *g* collude to make a wormhole between the two nodes. However, this portion of the networks seen through our scheme would look like what is seen in Figure 5. Here, node *f* is directly controlled by the cluster head, CH1, and *g* is directly controlled by CH2. The nodes, *f* and *g*, cannot collude to make a wormhole without knowing their respective cluster heads. Therefore, wormhole attacks are not feasible in our single hop clustering scheme.

Spoofing (Impersonation Attacks)

Some malicious nodes hide their addresses and use the address of another node during communication and does harm to the network. In such instances, a normal node gets falsely blamed, creating a loose trust level for such type of malicious nodes in the network. This type of attack is called spoofing.

In this scheme, every node is just a single hop away from its respective CH and sends a reply to HELLO messages from time to time to the respective CH. If any node tries to carry out spoofing attacks by hiding its address, the CH will check the address or identity through the link the node uses to connect to its CH, and if CH finds that the node is hiding its address, the CH will catch the malicious node in its cluster red-handed.

Slander Attack

This is quite similar to spoofing. Here, the malicious node attempts to reduce the overall trust of another node, but the malicious node does not hide its address. Instead, it colludes with other malicious nodes to send false information about a normal node to reduce their trust level within the networks.

A cluster member node might collude with other cluster member nodes and give false information to the CH to reduce the trust value of the target node. However, this is not possible for this scheme as explained earlier as every node is directly attached to its respective CH; therefore, increase or decrease of trust value of a node is directly done by the respective CH

without any certification of other member nodes. So, a slander attack can be resisted through the CH.

Routing Table Overflow Attack

This type of attack occurs basically on proactive routing, where the routing table is maintained by each node and even routes are not required. A malicious node sends false information by claiming it has many routes to many nodes, but those nodes do not actually exist. In this way, malicious nodes try to overflow the routing tables of other nodes so that other nodes cannot add more real route information into the routing table.

Some member nodes may unnecessarily send false information to its CH to overflow the routing table of that CH; therefore, the required routing information can be dropped by the CH. However, here the CH is the only node in a cluster that maintains the route and stores the routing table and, as already mentioned, this scheme assumes that the CH node is trust worthy. Therefore, the question of this kind of attack does not arise in our routing scheme.

Grey-Hole Attack

Here, malicious nodes flow along the same principle of behind a black-hole attack, but the malicious node drops selective packets such as data packets, and allows passage to control packets such as the RREQ packet. Therefore, another node falsely perceives the malicious node as a normal node.

A grey-hole attack is very difficult to catch because the malicious node passes the controlling packet, triggering the same resistance to the grey-hole attack as to a black-hole attack. This can come only from a gateway node. This scheme assures data packet delivery only after getting the acknowledgment message of the respective data packet from neighbouring CH. Therefore, this scheme can raise adequate resistance to grey-hole attacks.

SIMULATION

The simulation environment is given in Table 1. Performance analysis of the algorithms was done using the network simulation (NS-2) version 2.33. C3H was compared with CCS and EAACK, which are two state-of-the art approaches of detection of selfish and malicious nodes. Simulation metrics are a percentage of correct detection, malicious nodes, network throughput (percentage of data packets that can reach their respective destinations) and end-to-end delay per session.

In C3H, each CH computes the trust value of its members based on their previous activities and this trust value is considered along with residual energy and relative velocity. Therefore, if a node ceases to forward one particular message, the CH can easily investigate the chances of its complete exhaustion and breakage of links.

Table 1
Simulation parameters

Requirements	Specification
Topology area	500 m × 500 m
Traffic type	Constant bit rate(CBR)
Packet size	512 bytes
HELLO packet interval for original versions of protocols	10 ms
Node mobility	10-30 m/s
Signal frequency	2.4 GHz
Channel capacity	2 Mbps
Transmission power	300-600 mW
Receiving power	50-300 mW
Mobility model	Random waypoint
Radio range	50-100 m
Initial energy of nodes	5-10 j
Pause time	1 s
Number of nodes	20, 40, 60, 80, 100

Unlike CCS and EAACK, C3H considers residual energy of nodes and relative velocity between a CH and its members. If residual energy is very high and relativity is low, but the node does not respond to the message forwarding request of its CH, it is accused of malicious activity and its trust value reduces. If this trust value reduces below a pre-defined limit, the node is blacklisted network-wide. As seen in Figure 6, correct detection of malicious activity is higher in case of C3H.

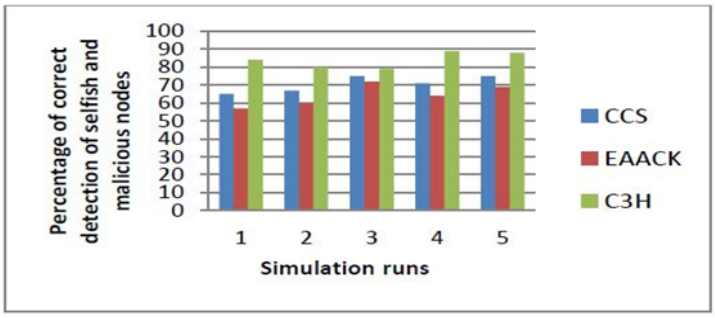

Figure 6. Percentage of correct detection of malicious activities in different simulation runs

The reason for this is that the competitors mentioned do not consider factors like energy and velocity and therefore, sometimes punish non-malicious nodes; this is not the right behaviour. In this way, we lose links to certain good nodes and also, packets generated by them are not forwarded to their respective destinations; no nodes cooperate with them. So, network throughput in C3H is much higher than CCS and EAACk, as seen in Figure 7.

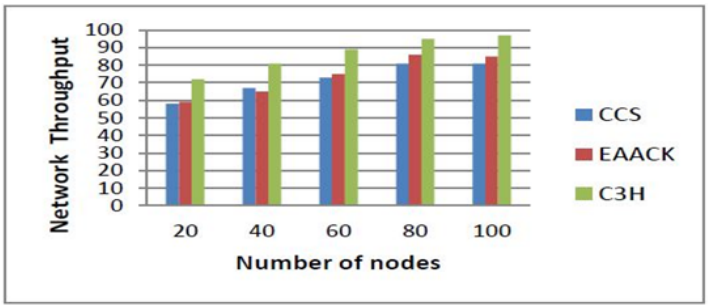

Figure 7. Network throughput vs number of nodes

Figure 8 shows end-to-end delay, which is much less in C3H due to availability of a number of good links.

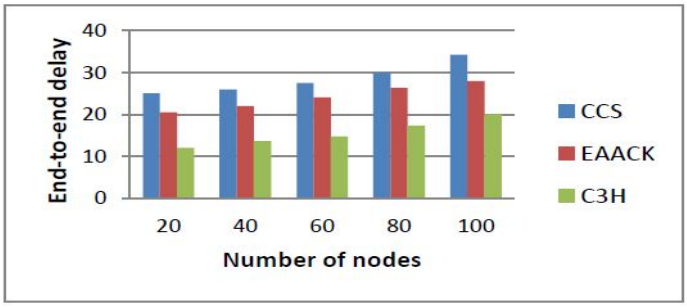

Figure 8. End-to-end delay per session vs number of nodes

CONCLUSION AND FUTURE SCOPE

In this scheme, cluster heads (CHs) are the most vital node as this scheme assumes the CH to be the most trustworthy node. Therefore, choosing the best candidate for cluster head is extremely important, and this scheme does the same using four parameters described earlier under the subsection, 'Electing Cluster Head (CH)'. Through these CHs, malicious attacks can be avoided and prevented. The CH takes packet transfer requests and gives processing priority based on trust values of that node. This clustering strategy can also be used to increase node co-operation, which is essential to successful data transmission in civilian data communication using MANETs.

REFERENCES

Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks, 2*(1), 1–22.

Banerjee, A., Dutta, P., & Sufian, A. (2018). Fuzzy-controlled energy-efficient single hop clustering scheme with (FESC) in ad hoc networks. *International Journal of Information Technology, 10*(3), 213–327.

Banerjee, A., Sufian, A., & Duta, P. (2018). EMR-PL: Energy-efficient multipath routing based on link life prediction in ad hoc networks. *Journal of Information and Optimization Sciences, 39*(1), 285–301. doi: 10.1080/02522667.2017.1374733.

Berri, S., Varma, V., Lasaulce, S., Radjef, M. S., & Daafouz, J. (2017, May). Studying node cooperation in reputation-based packet forwarding within mobile ad hoc networks. In *International Symposium on Ubiquitous Networking* (pp. 3–13). Springer, Cham.

Buttyan, L., & Hubaux, J. P. (2001). *Nuglets: A virtual currency to stimulate cooperation in self-organized mobile ad hoc networks.* Technical Report No. DSC/2001/001, Swiss Federal Institute of Technology, Lausanne.

Chang, J. M., Tsou, P. C., Woungang, I., Chao, H. C., & Lai, C. F. (2015). Defending against collaborative attacks by malicious nodes in MANETs: A cooperative bait detection approach. *IEEE Systems Journal, 9*(1), 65-75.

Deng, H., Li, W., & Agrawal, D. P. (2002). Routing security in wireless ad hoc networks. *IEEE Communications Magazine, 40*(10), 70-75.

Helen, D., & Arivazhagan, D. (2014). Applications, advantages and challenges of ad hoc networks. *Journal of Academia and Industrial Research (JAIR), 2*(8), 453–457.

Hernandez-Orallo, E., Olmos, M. D. S., Cano, J. C., Calafate, C. T., & Manzoni, P. (2015). CoCoWa: A collaborative contact-based watchdog for detecting selfish nodes. *IEEE Transactions on Mobile Computing, 14*(6), 1162–1176. doi:10.1109/TMC.2014.2343627.

Johnson, D., Hu, Y. C., & Maltz, D. (2007). The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. *IETF RFC 4728*.

Kang, N., Shakshuki, E. M., & Sheltami, T. R. (2010). Detecting misbehaving nodes in MANET. In *Proceeding iiWAS '10 Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services* (pp. 216–222). ACM. doi: 10.1145/1967486.1967522

Kargl, F., Klenk, A., Schlott, S., & Weber, M. (2014). Advanced detection of selfish or malicious nodes in ad hoc networks. In *European Workshop on Security in Ad-hoc and Sensor Networks* (pp. 152-165). Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-30496-8_13

Marina, M. K., & Das, S. R. (2001). On-demand multi path distance vector routing in ad hoc networks. In *Proceedings of the Ninth International Conference on Network Protocols, IEEE Computer Society* (pp. 14–23). Washington, DC, USA.

Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile Ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* (pp. 255-265). ACM.

Michiardi, P., & Molva, R. (2002). Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Advanced Communications and Multimedia Security* (pp. 107-121). Springer, Boston, MA. doi:10.1007/978-0-387-35612-9_23

Murthy, S., & Aceves, J. J. G. L. (1996). An efficient routing protocol for wireless networks. *Mobile Networks and Applications, 1*(2), 183–197.

Nasser, N., & Chen, Y. (2007). Enhanced intrusion detection system for discovering malicious nodes in mobile ad hoc networks. In *Communications, 2007. ICC'07. IEEE International Conference on* (pp. 1154-1159). IEEE. doi: 10.1109/ICC.2007.196.

Park, V., & Corson, S. (2001). *Temporary-ordered routing algorithm (TORA)*. Internet Draft, draft-ietf-manettora-spec-04.txt.

Pei, G., Gerla, M., & Chen, T. W. (2000). Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on* (Vol. 1, pp. 70-74). IEEE. doi: 10.1109/ICC.2000.853066.

Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM Computer Communication Review* (Vol. 24, No. 4, pp. 234-244). ACM.

Perkins, C. E., & Royer, E. M. (1998). Ad-hoc n-demand distance vector routing. *draft-ietf-manet-aodv-02.txt.*

Toh, C. K. (2002). *Ad hoc mobile wireless networks: Protocols and systems.* New Jersey: Prentice Hall PTR.

Zhong, S., Chen, J., & Yang, Y. R. (2003). Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies* (Vol. 3, pp. 1987-1997). IEEE. doi: 0-7803-7753-2/03