Send Orders for Reprints to reprints@benthamscience.net

International Journal of Sensors, Wireless Communications and Control, **2013,** *3,* **000-000**

A Cooperation Enforcement, Malice Detection and Energy Efficient Mechanism for Mobile Ad hoc Networks

Ehsan Ataie[1*], Ali Movaghar[2] and Mostafa Bastam[1*]

[1]*University of Mazandaran, Babolsar, Iran;* [2]*Sharif University of Technology, Tehran, Iran*

Abstract: Mobile ad hoc networks communicate in a self-organized way without depending on any fixed infrastructure. The issue of selfish nodes, which may refuse to cooperate, is a great challenge in such networks and may cause network throughput to drastically reduce. Energy-based selfishness is a category of selfishness in which a selfish node shows non-deterministic and probabilistic selfishness behaviors based on level of its energy. In this paper, we propose a mechanism for coping with this kind of selfishness. This mechanism called CEMDEEM not only detects and isolates energy-based and traditional selfish nodes, but also malicious behaviors like spoofing. We also evaluate performance of Dynamic Source Routing (DSR) protocol fortified by CEMDEEM in the presence of different percentages of selfish nodes. Results show that CEMDEEM noticeably improves network performance with a reasonable additional packet overhead and network delay especially when percentage of selfish nodes is not more than 40.

Keywords: MANET, Malice, Reputation, Routing, Security, Selfishness.

1. INTRODUCTION

A mobile ad hoc network is a self-configuring infrastructure-less network of mobile devices connected by wireless. Ad hoc networks are basically peer-to-peer multi-hop mobile wireless networks where information packets are transmitted in a store-and-forward manner from a source to a destination, via intermediate nodes. They are useful when infrastructure is not available, is impractical or is expensive. In such networks, cooperation takes place at the level of routing, i.e. finding a path for a packet, and forwarding, i.e. relaying packets for other nodes.

However, misbehaving nodes may not follow the cooperation paradigm and cause serious affects on network performance. Nodes misbehave because they are malicious, selfish or malfunctioning. Intentional misbehavior aims at providing an advantage for the misbehaving node [1]. An advantage for a malicious node arises when misbehavior allows it to mount an attack [2]. An example for an advantage gained by misbehavior is power saved when a selfish node does not forward packets for other nodes. *Selfish* nodes use the network but do not cooperate [3].

Different implications of selfishness include degradation of network throughput, loss of packets, partitioning of network, and denial of nodes' services. These harmful effects of selfishness may jeopardize the functioning of the whole network [4][5].

Basic routing protocols in MANETs like DSR [6], AODV [7], TAODV [8], DYMO [9], DSDV [10], ZRP [11] VBR [12] and IZR [13] did not have any mechanism for withstanding security threats and specially selfishness. Subsequent ones such as SRP [14], ARAN [15], SEAD [16], ARIADNE [17], SAODV [18], SAR [19] and SLSP [20] could not still cope with selfishness even though they added some security features to older basic protocols. Analysis and comparison of these protocols and their capabilities are studied in [21-23].

Marti, Guiti, Lai and Baker [24] tried to mitigate selfishness problem by proposing watchdog/pathrater mechanism. Every node has a watchdog process that monitors the direct neighbors by promiscuously listening to their transmission. Buttyan and Hubaux in [25] made use of nugget-based approaches to encourage nodes to perform packet forwarding and routing functions. Michiardi and Molva [26] proposed a mechanism in which each node monitors its neighbor nodes and excludes selfish nodes from routing services. Alarm messages are also broadcasted to inform other nodes of misbehavior observations. In [27] Yang and Meng introduced a token-based mechanism to enforce cooperation among nodes. Buchegger *et al.* in [4, 28, 29] proposed untiselfishness mechanisms based on nodes' reputation. Rodriguez-Mayol and Gozalvez investigated the impact of accurate radio propagation models, channel congestion and operating conditions, on the performance of selfishness preventive techniques, and their capability to detect selfish nodes in [30]. Some existing schemes and their relative advantages and disadvantages are summarized in [31]. Nevertheless, proposed mechanisms have deficiencies like false recognition of selfishness behavior or propagation of alarm messages all over the network; So, withstanding selfishness is still an open subject.

Almost all previous studies assumed selfishness to be absolute meaning that a node either forward packets or drops them. But authors in [32] presented a series of nondetermin-

*Address correspondence to these authors at the University of Mazandaran, Babolsar, Iran; Tel:/Fax: +981125342460; E-mails: ataie@umz.ac.ir; bastam@umz.ac.ir

2210-3279/13 $58.00+.00

istic and probabilistic selfishness behavior. It seems more realistic; because most mobile nodes in a typical MANET application like military or rescue missions are directly or indirectly controlled by human being and as a human behavior, it is more likely that selfishness depends on node's energy level. In this paper, we tried to withstand these energy-based models.

A point in case regarding deficiency of previously mentioned protocols for copping with energy-based selfishness is incorrect accusation and reputation improvement. There are always benign nodes which may be incorrectly accused to be malicious or selfish because of link breakages. Some of the above mechanisms do not handle to exculpate these benign nodes. Some others usually use techniques to allow nodes to improve their reputation. These techniques allow such nodes not to permanently be recognized as selfish nodes. On the other hand, in these models a selfish node may occasionally forward packets. So, it can improve its reputation and never be detected as a selfish node if we use above techniques. In addition, propagation of alarm messages in the network as is done in some previous mechanisms, will deteriorate the energy of sender, intermediate, and destination nodes and thus would increase overall selfishness behaviors.

In this paper, we introduce a mechanism which detects traditional and energy-based selfish nodes and some malicious nodes and also takes into account energy issues that mentioned above. In this way, we also measure network throughput, packet overhead, and network delay when unfortified DSR is used as routing protocol and when DSR is fortified with proposed mechanism and compare the results. We name our mechanism CEMDEEM: Cooperation Enforcement, Malice Detection, and Energy-Efficient protocol.

The remainder of the paper is organized as follows: Section 2 discusses about energy-based selfishness models. In section 3, CEMDEEM mechanism is introduced and its components are explained. Section 4 includes results of simulation according to network throughput, packet overhead, and network delay. Finally, in section 5 we conclude the current study and discuss about future works.

2. ENERGY-BASED SELFISHNESS MODELS OF NODE

Selfishness in mobile ad hoc network has a significant importance, since damages it causes cannot be mitigated by common security mechanisms like cryptography. On the other hand, it is almost probable in such networks that nodes act selfishly when they have limited energy power. This initiative causes energy-bases selfishness models that introduced in [32] as follows:

Suppose E and E_0 are current and initial energy of a node. If we define S_i as probability of selfishness in behavior of node i (i.e. probability that node i drops a packet), then a series of energy-based S_i can be defined as follows:

$$S_i(E,E_0) = 1 - \frac{E}{E_0} \tag{1}$$

$$S_i(E,E_0) = \left(\frac{1 - E/E_0}{1 + E/E_0}\right)^k \tag{2}$$

$$\begin{aligned} &if \;\; E/E_0 \geq 1/k \;\; S_i(E,E_0) = 0 \\ &if \;\; E/E_0 \prec 1/k \;\; S_i(E,E_0) = 1 \end{aligned} \tag{3}$$

which k is a positive integer

In contract, full or absolute selfishness can be defined independent of energy level of node:

$$S_i(E,E_0) = 1 \tag{4}$$

A full or absolute selfish node is one who may forward request packets, but drops data packets and do not reply requests. This assumption is because if such a node does not forward request packets, it will eventually be eliminated from the network.

3. CEMDEEM PROTOCOL

Here, we will illustrate our proposed routing protocol named CEMDEEM, This protocol can be applied to source route protocols in MANETs like DSR. The goal is to enforce nodes to cooperate for routing discovery and packet forwarding functions and also to discover and isolate selfish and malicious nodes of the network. Meanwhile CEMDEEM takes into account energy consideration and tries to gain best performance using less energy consumption.

This protocol is suited for networks in which nodes show energy-based selfishness, though it also shows good performance when selfishness is absolute. In the following sections, CEMDEEM components are introduced and explained.

3.1. Monitoring Component

This component acts like watchdog mechanism in Watchdog/Pathrater [24]. By setting network interface in promiscuous mode, each node can monitor the behavior of its neighbors. This component is responsible for two tasks explained below.

3.1.1. Saving Characteristics of Submitted Packets

Before sending a packet in CEMDEEM, destination is considered. If next node is final destination, there is no need to maintain packet status, since next node is not an intermediate node that should forward the packet. Otherwise, packet characteristics including source address, destination address, next hop and current time is saved in TRSP (Table of Recently Sent Packets) table to make its state considerable in future. Furthermore, the path between current node and source node is saved in order to be used for sending probably alarm messages. Maintaining reverse path is justified only if mechanism configuration allows sending alarm messages in early times.

3.1.2. Network Eavesdropping and Packet Adaptation

When a packet is captured in promiscuous mode of network interface, its destination will be checked. Depending on whether we are the source of the packet or not, CEMDEEM acts differently:

1). If we are the source, packet will be compared with TRSP information. If we actually sent it before, neighbor node did its mission correctly and can be praised.

So, a positive score will be granted to it. But if the packet characteristics did not match any TRSP entry, a spoofing or modification attack have happened.

2). If we are not the source, previous hop in source route will be considered. If we are that hop, we may probably be the hop which have delivered the packet to our neighbor for forwarding. If adaptation of packet specification to TRSP entries proves such assumption, node's reputation will be increased by a positive value. Otherwise, a spoofing or modification attack has happened and delinquent node will be punished by a negative score.

3.2. Reputation Evaluator Component

This component is responsible for evaluating and determining reputation of network nodes. At first, discredit or malice of all nodes is assumed to be zero. In other words, all nodes are seemed to be innocent before any negative observation.

Every malicious behavior of nodes causes a decrease in reputation or increase in discredit metric. We use reputation and discredit in adverse.

Reputation decrement of a node may cause scattering alarm messages against it or deprivation of service. Cooperation for delivering services to other nodes of network can result in reputation increment near other nodes.

3.2.1. Reputation Decrement

Direct and indirect observation of a node misbehavior will result in reduction of node reputation. We have previously discussed about some kind of direct observation concluding reputation reduction. This is when we find ourselves as source or the node before last node in the packet source route, but we cannot find packet specification in TRSP.

Search mechanism of TRSP periodically returns back all packets with expired validation timeout. These are packets that were delivered to neighbor nodes for forwarding, but might be dropped. Penalty of such neighbor nodes can depend on:

- Whether or not dropped packet belong to us
- What is current reputation of neighbor node?
- What is current reputation of packet's destination node?

Punishment can also be done depends on indirect observations. When we receive an alarm message signifying suspicion of a node, alarm handling component will call reputation evaluator and it will update reputation table based on previous records of suspect node and the node initiating alarm message.

3.2.2. Reputation Increment

Direct observation will result in reputation increase. Whenever our neighbor node forwards our packet or other's packet correctly, it will be awarded a positive reputation. Also when our route request replied by a route reply message, all nodes in the path towards destination will gain a positive reputation.

3.2.3. Specifying Trust, Alarm, and Service Thresholds

For the whole network a trust threshold, an alarm threshold, and a service threshold is defined. When discredit of a node exceeds the trust threshold, all path including that node will be eliminated by path evaluation component. When node's discredit exceeds the alarm threshold, alarm handling component will initiate an alarm to inform some other nodes of network. Service threshold is an index for determining whether the node is allowed to take advantage of routing services or not. In our strategy, alarm threshold is less than service threshold and more than trust threshold. This is because we may accuse a node to be misbehaved. Whereas some unforeseen reasons like obscure collision or collision at receiver may cause us to recognize incorrectly. Hastening in deprivation of a node from routing service can put us as suspect node in the lists of our neighbor nodes.

3.3. Path Evaluator Component

This component in conjunction with reputation evaluator component assesses the amount of trust to each path in the cache. Below, this responsibility is explained in more details.

3.3.1. Determining Path Credit

When a node wants to send a packet or decides to salvage a packet, it would first search paths in its routing table. DSR is based on choosing shortest path between paths found in routing table. But in CEMDEEM, selection of a path is a function of path length and path credit. Here, we have defined credit of a path as the lowest credit of nodes forming the path. Other credit computation algorithms can easily be applied to this component.

3.3.2. Eliminating Low Credit Paths

When discredit of a node exceeds the trust threshold, all paths containing that node will be eliminated from routing table.

3.3.3. Preventing Insertion of Low Credit Paths

In DSR, nodes learn new paths by listening to network traffic in promiscuous mode. CEMDEEM does not allow low credit paths which are found by this approach to be inserted in routing table.

3.4. Alarm Handling Component

In a MANET using CEMDEEM as its routing protocol, nodes inform each other of malicious or selfish ones by sending alarm messages. As a result, receiver of such alarms can modify their reputation table based on some formulas.

Alarm handling component is responsible for generating alarm messages against misbehaved nodes and managing alarms received from other trusted nodes in the network. Operation of this component can be illustrated in two major parts.

3.4.1. Generating and Sending Alarm Messages

When discredit of a neighbor node exceeds the alarm threshold, an alarm message will be generated and be sent to the source node whose packet had been dropped by our sus-

pect neighbor. Source route of this message is reverse of a prefix of the source route in dropped packet.

3.4.2. Receiving and Handling Alarm Messages

For performance reasons, alarm message will be considered and applied by three sets of nodes:

1). neighbor nodes: these nodes can listen the message in promiscuous mode. They have better evaluation of our reputation as alarm initiator and thus can decide more precisely against alarm validity and credibility. In addition, since suspect node is in our neighborhood it may also be a neighbor of our neighbors. So, recognizing a misbehaved node by its neighbors could be a great help to them in order to not establish their communication paths through that node.

2). intermediate nodes: nodes connecting us to victim node whose packets were dropped, should forward the alarm message towards destination node. Whereas energy consumption in intermediate nodes is indispensable for receiving and forwarding the message, it is absolutely economical to process the alarm for recognizing selfish node.

Since suspect node is on the path between destination and victim source of dropped packet, path learning mechanism done by intermediate nodes which is designed in basic DSR could be very dangerous. So, for the sake of overall network performance, these nodes should know misbehaved node and do not save such paths.

3). source node of dropped packet: this node is the main victim of selfish or malicious node. If the source still sends more packets through old dangerous path, receiving enough number of alarm messages will cause it to find some new probably safer path and thereby prevent wasting its energy and packets.

Alarm messages indeed provide indirect observation for CEMDEEM mechanism. Neighbors of alarm initiator, intermediate nodes, and victim node can update their reputation table according to current credit of both suspect and initiator node. After updating the table, each of these entities will act as we explained below:

- Neighbor nodes which receive alarm in promiscuous mode will free the message
- Intermediate nodes will forward message to the next node according to source route of alarm packet
- Victim node will free the message

4. SIMULATION RESULTS

The criteria for measuring and analysis of network performance are *network throughput* i.e. proportion of received packets at destination nodes to sent packets, *network delay* i.e. average time between packet send and packet receipt for all packets that are correctly received, and *packet overhead* i.e. the average number of control packets sent for each data packet.

Nodes communicate using Constant Bit Rate sources that are randomly bound to a subset of all the nodes forming the network. There are six sources and nine destinations forming ten CBR connections. Among them four are source of two connections each and each of two others are source of one CBR connection. Eight destination nodes receive just one connection and there is one node which is destination of two connections. For each connection, source node starts to send packets at a random time and continue packet transmission until end of simulation time. Packet size is set to 512 bits while the source throughput is one packet per second.

In all movement scenarios, a node randomly chooses a point as its next destination and move towards the point at a constant speed.

Movement and communication patterns have been generated using the tools provided by the CMU extensions to ns-2.

Here, we have compared three situations: performance of unfortified DSR when nodes' selfishness obey equation (1), performance of CEMDEEM when selfishness probability follows equation (1), and performance of CEMDEEM when selfish nodes behave based on function (4). Equation (1) is selected for simulation as a typical nondeterministic and probabilistic form of selfishness. Detailed values of simulation parameters are shown in table 1. Table 2 contains the value of protocol parameters used for simulation scenarios.

Table 1. Simulation Parameters.

Variable	**Value**
Size of Environment	1000^{m} x 1000^{m}
Number of Nodes	30
Speed	Uniform [$2^{m/s}$, $5^{m/s}$]
Pause Time	Uniform [0^{s}, 120^{s}]
Initial Energy	50^{J}
Simulation Time	1000^{s}

Table 2. CEMDEEM Parameters.

Variable	**Value**
Tolerable	15
Trust Threshold	0.7
Alarm Threshold	0.8
Service Threshold	0.9

Following, we will describe each parameter:

- Tolerable: maximum amount of discredit of a node. Initial value of nodes' discredit is zero.
- Trust, Alarm, and Service Thresholds: proportion of the Threshold constant that when a node's discredit exceeds them, it will be acted as describe in section 3.2.

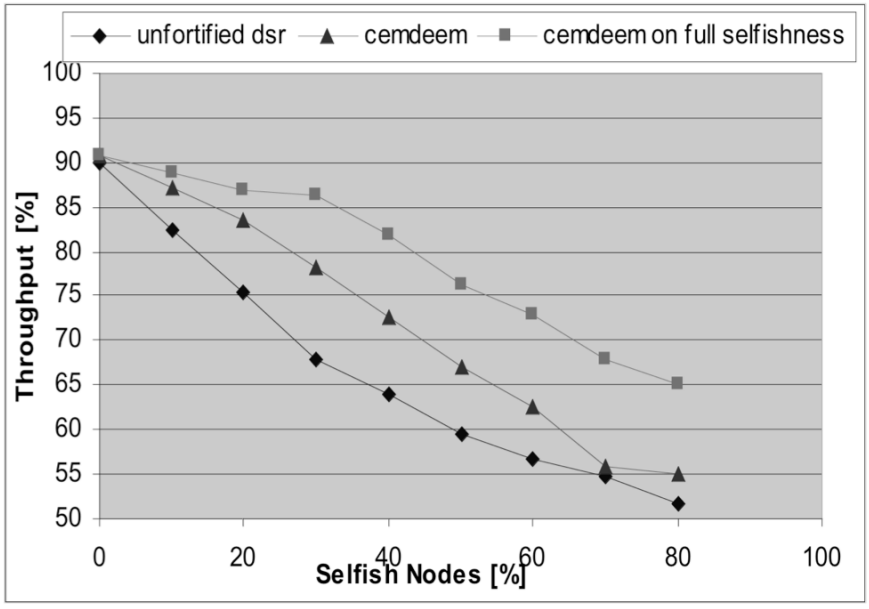

Fig. (1). Throughput Comparison.

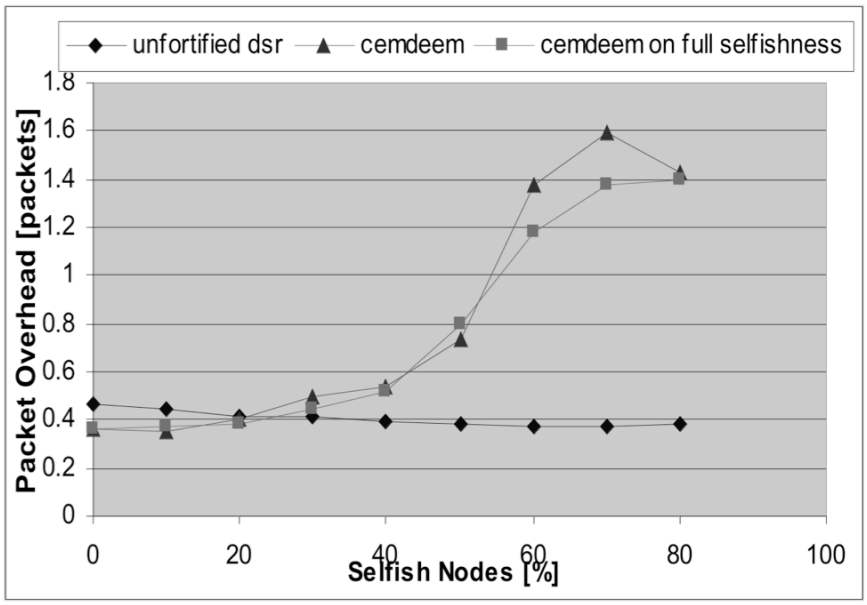

Fig. (2). Packet Overhead Comparison.

Fig. (**1**) shows throughput results in the presence of different percentages of selfish nodes. Best operation of CEMDEEM in contrast with unfortified DSR can be achieved when about 20-40 percent of nodes act selfishly. There, CEMDEEM can improve network throughput about 8-11 percent.

Meanwhile, comparing CEMDEEM to unfortified DSR, throughput is enhanced about 19 percent when selfish nodes absolutely drop data packets. This magnitude of throughput improvement demonstrates CEMDEEM efficiency in comparison with the maximum of 17 percent improvement of network throughput when watchdog/pathrater is applied. In other words, the protocol shows a noticeable performance not only in the presence of energy-based selfishness, but also when selfishness is absolute.

Increase of percentage of selfish nodes gradually affects performance of CEMDEEM in such a way that CEMDEEM comparatively acts like unfortified DSR when percentage of energy-based selfish nodes reaches 70. Our investigation shows that CEMDEEM can still detect these nodes. But in high percentages, most of the actual paths include selfish nodes. This causes a huge number of route discovery messages to be sent across the network and increases packet

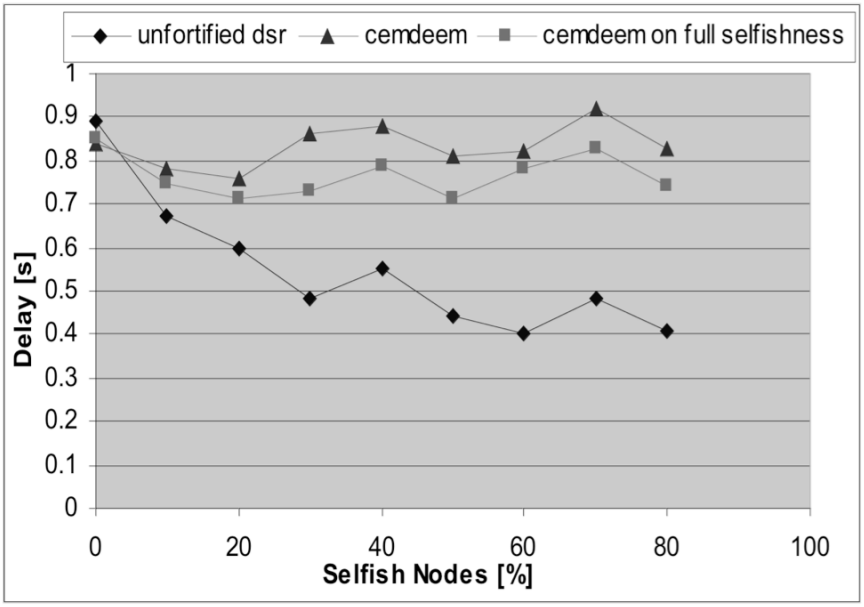

Fig. (3). Delay Comparison.

overhead and network delay as can be seen in (Figs. **2** and **3**).

Fig. (**2**) shows packet overhead in the presence of different percentages of selfish nodes. Network delay diagram is shown in Fig. (**3**).

As it can be seen, network delay gradually decreases when unfortified DSR is applied. Since in high percentages, most of the paths include selfish nodes, most of the packets will never arrive at destination. So, only those packets with source and destination in the neighborhood or close distance are delivered. Thus according to the definition, network delay will decrease.

5. CONCLUSION

Probabilistic and nondeterministic models of selfishness in MANETs are dependent on node's instantaneous energy level. Since selfishness usually arises from node's interest in its survivability, these models seem to be reasonable and probable. In such models a selfish node can improve its reputation by occasionally forwarding packets and may never be recognized as a selfish node. So, detecting a true selfish node is not a trivial problem. In addition, propagation of alarm messages in the network, as is done in some other mechanisms, will causes average energy of nodes to decrease and thus increases selfishness behavior.

In this research, we introduce CEMDEEM protocol for coping with energy-based selfishness. This mechanism helps to discover and isolate selfish and malicious nodes of the network. Meanwhile CEMDEEM takes into account energy consideration and tries to gain best performance using less energy consumption.

Simulation results show that CEMDEEM noticeably enhances network performance and meanwhile stays packet overhead and network delay reasonable especially when percentage of selfish nodes is under 40.

Though CEMDEEM can cope with some malice behavior, our next plan is to fortify this protocol in such a way that it could resist more malicious attacks. We also plan to enhance the protocol to distinguish between selfishness and malice using two different numerical criteria. Another research is also conducted to take into account different weights for direct and indirect observations of misbehavior.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Zhou, L., Haas, Z.J.: Securing Ad Hoc Networks. IEEE Network Magazine, vol. 13, no. 6, pp. 24-30 (1999).

[2] Deng, H., Li, W., Agrawal D.P.: Routing Security in Wireless Ad Hoc Networks. IEEE Communication Magazine, vol. 40, no. 10, pp. 70-75 (2002).

[3] Michiardi, P., Molva, R.: Ad hoc networks security. ST Journal of System Research (2003)

[4] Michiardi, P., Molva, R.: Ad hoc networks security. Mobile Ad Hoc Networking, vol. 1, pp. 275--297 (2004).

[5] Buchegger, S.: Coping with Misbehavior in Mobile Ad Hoc Networks. PhD thesis, EPFL (2004).

[6] Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Mobile Computing, Chapter 5, pp. 153—181, Kluwer Academic Publishers (1996).

[7] Perkins, C.: Ad hoc On Demand Distance Vector (AODV) Routing. Internet draft, http://tools.ietf.org/html/draft-ietf-manet-aodv-13 (2003).

[8] Iftikhar M. and Zomaya A.: Improving Performance of Mobile Ad Hoc Networks Using Efficient Tactical on Demand Distance Vector (TAODV) Routing Algorithm, International Journal of Innova-

tive Computing, Information and Control, vol.8, no.6, pp.4375-4389 (2012).

[9] Chakeres I. and Perkins C.: Dynamic MANET On-demand Routing Protocol (DYMO), Internet Draft, http://tools.ietf.org/html/draft-ietf-manet-dymo-26 (2013).

[10] Perkins, C., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: SIGCOMM 94 conference on Communications Architectures (1994)

[11] Hass, Z.J., Pearlman, M.R.: The Zone Routing Protocol (ZRP) for Ad Hoc Networks. , Internet draft, http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-04.txt (2002).

[12] Liang, B., Haas, Z.J.: Hybrid Routing in Ad Hoc Networks with Dynamic Virtual Backbone. IEEE Transactions on Wireless Communications, vol. 5, no. 6, pp.1392-1405 (2006).

[13] Samar, P., Pearlman, M.R., Haas, Z.J.: Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks. ACM/IEEE Transactions on Networking, vol.12, no.4, pp.595-608 (2004).

[14] Papadimitratos, P., Haas, Z.J.: Secure Routing for Mobile Ad Hoc Networks. In: ACM Mobile Computing and Communications Review (MC2R), vol. 1, no. 2 (2002).

[15] Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: A Secure Routing Protocol for Ad Hoc Networks. In: 10th IEEE International Conference on Network Protocols (2002).

[16] Hu, Y.C., Johnson, D.B, Perrig, A.: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In: 4th IEEE Workshop on Mobile Computing Systems and Applications (2002).

[17] Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks. In: 8th ACM International Conference on Mobile Computing and Networking (2002).

[18] Zapata,M.G.: Secure ad-hoc on-demand distance vector (SAODV) routing, IETF MANET, internet draft, http://people.ac.upc.edu/guerrero/papers/draft-guerrero-manet-saodv-06.txt (2006).

[19] Yi,S., Naldurg ,P., Kravets ,R.: Security aware ad-hoc routing for wireless networks, In: Proc. of the 2nd ACM international Symposium on Mobile Ad hoc networking and Computing (Mobi - Hoc'01), pp.299-302 (2001).

[20] Papadimitratos, P., and Haas,Z.: Secure link state routing for mobile ad-hoc networks, In: Proc. Of Symposium on Applications and the internet Workshops (SAINT'03), pp. 379-383 (2003).

[21] Abusalah, L., Khokhar, A., Guizani, M.: A Survey of Secure Mobile Ad Hoc Routing Protocols. IEEE communications surveys & tutorials, vol. 10, no. 4 (2008).

[22] Hu, Y., Perrig, A.: A Survey of Secure Wireless Ad Hoc Routing, IEEE security & privacy, vol. 2, no. 3, pp. 28-39 (2004).

[23] Ertaul, L., Ibrahim, D.: Evaluation of Secure Routing Protocols in Mobile Ad Hoc Networks (MANETs). In: Security and Management, pp. 363--369 (2009).

[24] Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: 6th Annual International Conference on Mobile Computing and Networking, ACM Press, pp. 255-265 (2000).

[25] Buttyan, L., Hubaux, J.P.: Nuglets: a Virtual Currency to Stimulate Cooperation in Self-organized Ad Hoc Networks. Technical Report, Swiss Federal Institute of Technology, Lausanne (2001).

[26] Michiardi, P., Molva, R.: Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks. In: IFIP - Communication and Multimedia Security Conference (2002).

[27] Yang, H., Meng, X., Lu, S.: Self-Organized Network-Layer Security in Mobile Ad Hoc Networks. In: Proceedings of ACM MOBICOM Wireless Security Workshop, Atlanta, pp. 11-20 (2002).

[28] Buchegger, S., Boudec, L.: Self-policing mobile ad hoc networks by reputation systems. IEEE Communications Magazine, vol. 43, no. 7, pp. 101-107 (2005).

[29] Buchegger, S., Mundinger, J., Boudec, J.Y.: Reputation Systems for Self-Organized Networks. IEEE Technology and Society Magazine, vol. 27, no. 1, pp. 41-47 (2008).

[30] Mayol, A.R., Gozalvez, J.: On the Implementation Feasibility of Reputation Techniques for Cooperative Mobile Ad-hoc Networks. In: European Wireless Conference EW2010, pp. 616--623 (2010)

[31] Yoo, Y., Agrawal, D.P.: Why does it pay to be selfish in a MANET?. IEEE Wireless Communications Magazine, vol. 13, no. 6, pp. 87--97 (2006).

[32] Ataie, E., Movaghar, A.: Performance Evaluation of Mobile Ad Hoc Networks in the Presence of Energy-based Selfishness. In: 3rd IEEE International Conference on Broadband Communications, Networks, and Systems (BroadNets), USA (2006).

Received: May 29, 2013 Revised: September 17, 2013 Accepted: September 23, 2013