Stimulating Cooperation in Mobile Ad hoc Networks using Cut Diamond with Diamond method

Sangheethaa Sukumaran[1], Dr. J.Venkatesh[2], Arun Korath[3]

[1] **Research Scholar, Anna University Institute of Technology, Jothipuram Campus, Coimbatore, Tamilnadu, India,**

[2] **Associate. Professor, Department of Management Studies, Anna University Institute of Technology, Jothipuram Campus, Coimbatore, Tamilnadu, India**

[3] **Assistant Professor, Department of CSE, Vedavyasa Institute of Technology, Malappuram, Kerala**

Abstract

In recent years mobile ad-hoc networks have become very popular because of their widespread usage. Cooperation among the nodes in ad-hoc networks is an important issue for communication to be possible. But some nodes do not cooperate in communication and saves their energy. These nodes are called as selfish nodes. In the literature there are many methods which deal with the selfish behavior of the nodes. This paper proposes an approach based on incentive mechanism in a different manner. Here nodes are "made" selfish for some time to encourage nodes to cooperate in the communication. This reduces selfish behavior. Thus this approach is called as cut diamond with diamond.

Keywords: *selfish nodes, ad hoc networks, incentive based*

1. Introduction

Mobile ad-hoc networks are self organizing, self cooperating infrastructure less networks. The emerging mobile ad hoc networking technology seeks to provide users "anytime" and "anywhere" services in a potentially large infrastructure less wireless network, based on the collaboration among individual network nodes. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. In such a dynamic environment routing the packets reliably to the destination becomes a critical issue. All the nodes in an ad-hoc network acts as a router and cooperate among themselves for proper functioning of the network. It is assumed that all the nodes that participate in the network will do forwarding and routing in favor of other nodes. But this assumption does not work in all cases. Sometimes the nodes agree to forward, but fail to do because they want to save their battery power and CPU cycles. They just keep receiving the data destined to them, and drop the data of other nodes without forwarding or routing them, which reduces the throughput of the network. These nodes are called as misbehaving nodes. This paper is organized in to following sections. Section 2 explains related work in this area. Section 3 gives idea about the proposed method. Section 4 is the simulation results and section 5 gives the conclusion and future work.

2. Related Work

There are basically two broad classification of dealing with selfish nodes. One method is to punish the selfish nodes and to isolate them another method is to credit the unselfish node. This paper follows the second approach. There are other methods in the literature which are dealt in brief in the following sections.

2.1 Secure Routing Protocols for Access Control

2.1.1 Watch Dog and Path Rater

Many routing protocols like DSR [1], AODV [2] have been developed for Mobile ad-hoc networks. S.Marti et al [3] addresses the problem of nodes agreeing to forward packets of other nodes but fail to forward. This describes two mechanisms to improve the throughput of the network. One mechanism is the watchdog, which identifies the misbehaving node by monitoring the nearby nodes whether they forward the packets of other nodes in the network. The other mechanism is the path rater that defines the best route by avoiding those misbehaving nodes. Since this approach tries to avoid the misbehaving nodes for routing, there's less chance of dropping packets, thus providing a better throughput even in the presence of high number of misbehaving nodes. But this approach does

not isolate the misbehaving nodes; they still utilize the network services, i.e. the nodes are not punished for misbehaving.

2.1.2 Trust based Secure Routing Protocol

Houssein Hallani and Seyed A. Shahrestani [9] proposed a fuzzy based trust model for nodes. This approach works on AODV routing protocol. Fuzzy logic helps to quantify trust between nodes in ad hoc networks. This paper addresses the following problems. Packets dropped, wrong forwarding, fabrication and replay attacks. This evaluation model is a Mamdani type with four input and one output variables. The elements of a fuzzy set are mapped by membership functions to a value, which defines the degree to which a fuzzy variable is a member of a set. The membership functions μ(P), μ(WF), μ(F), μ(RA), μ(T), map the input variables, packet_dropped, wrong_forwarding, fabrication and replay_ attack, and the output variable, trust_level, into the interval (0,1) respectively. After applying the fuzzy trust evaluation model each node will have a trust level. Each node is assumed to be able to evaluate the trust level of each of its neighbouring nodes based on the information regarding the behaviour history of these nodes. These trust levels are then used to determine the most appropriate route between S and D. But this approach is specific for AODV [10].

2.2 Reputation based access control mechanisms

2.2.1 Reputation Based Intrusion Detection System

Animesh KR trivedi et al[12] proposed a reputation based intrusion detection system for Mobile Adhoc Networks (RISM). RISM system runs on every node in network and consists in core of the following modules: The Monitor, holds the responsibility of monitoring activities in the Neighborhood using PACKs (Passive ACKnowledgements) Every node registers all the data packets sent by it to next node and when it receives packets in promiscuous mode, it matches those to the queue of registered packets present in its buffer. After a fixed time interval -termed as the Timing Window, nodes make a log of number of packets for which they haven't received acknowledgment in the form of PACK and communicate it to the reputation manager. Monitor maintains a log of activity of next neighbor for each Window and sends it to Reputation manager. Reputation system receives activity log of next hop neighbor from monitor with number of packets for which it does not receive PACK, called as Missing or Dropped Packets. The number of missing packets is then compared with the Malicious DropThreshold and if it is comparatively lesser, then the reputation manager gives positive performance appraisal else negative. The path manager performs trivial path management functions in collaboration with DSR core. Redemption and Fading are included in design of RISM to allow nodes previously considered malicious to become part of network again as ad-hoc networks run on cooperation and collaboration of peer nodes and no one gets benefited without cooperating with each other. Congestion parameter, Knock test and Timing window are some new concepts that are introduced in this paper.

2.2.2 Reputation Based mechanism to isolate selfish nodes

M. Tamer Refaei et al[13] proposed reputation- based mechanism as a means of building trust among nodes. The mechanism relies on the principle that a node autonomously (i.e., without communicating with other neighboring nodes) evaluates its neighbors based on the completion of the requested service(s). This mechanism based on trust management schemes does not rely on the monitoring of neighbors' transmissions and the exchange of reputation information among nodes. Thus involves less overhead, and this approach does not rely on any routing protocol. This approach provides a distributed reputation evaluation scheme implemented autonomously at every node in an ad hoc network with the objective of identifying and isolating selfish neighbors. Each node maintains a reputation table, where a reputation index is stored for each of the node's immediate neighbors. A node describes a reputation index to each of its neighbors based on successful delivery of packets forwarded through that neighbor. For each successfully delivered packet, each node along the route increases the reputation index of its next-hop neighbor that forwarded the packet. Conversely, packet delivery failures result in a penalty applied to such neighbors by decreasing their reputation index. In other words, when a node transmits a packet to one of its neighbors, it holds the neighbor responsible for the correct delivery of the packet to the final destination. The indication of a success or failure is obtained from feedback received from the destination (*e.g.*, using TCP acknowledgements). The function used to compute the reputation index is a design decision that is influenced by factors including node behavior, node location, as well as others.

To prevent selfish behavior and to provide motivation for nodes to build up their reputation, each node determines whether to forward or drop a packet based on the reputation of the packet's previous hop. Once a node's reputation, as perceived by its neighbors, falls below a pre-determined threshold all packets forwarded through or originating at that node are discarded by those neighbors and the node is isolated. Summarizing, 1. To evaluate neighboring nodes based on the completion of the requested tasks (packet delivery); and 2. To detect the completion of a task based on feedback received from the end host (delivery acknowledgement). Advantages of this approach are, 1. Routing Protocol independence, 2. no need for monitoring the neighboring nodes in a promiscuous mode. 3. Less overhead since nodes does not pass reputation information. But the problem with this approach is that, it uses feedback mechanisms like TCP acknowledgements in connection oriented applications for identifying whether a packet has

reached the destination or not. So this method is not suitable for connectionless applications.

2.3 Ticket based approaches for access control

There are many approaches in the literature, which deals with access control in ad-hoc networks. But only few papers [2] [3] [4] deal with packet forwarding and routing misbehaviors.

2.3.1 Centralized and Distributed server

L.Zhou et al [4] and G.Appenzeller et al [5] proposed ticket based approaches. Tickets are provided for the nodes, which are well behaving, and network access is provided only to the nodes with a valid ticket. The ticket is obtained from a centralized authority [4] or from distributed servers [5]. The central server approach has several advantages and disadvantages. The central server approach can work well for a simple, less dynamic network. But for a dynamic network the delay will be more. The distributed approach has no much difference with central authority system except that here there are three or more central servers in the network. In both the approaches when the central server fails, the network functioning becomes vulnerable to attacks.

2.3.2 Localized Approach for Access Control

The localized approach for access control is proposed by Haiyoun Luo et al [6]. This is a ticket-based approach. The localized approach [6] proposes a fully localized design paradigm to provide ubiquitous and robust access control for mobile ad hoc networks. Each well behaving node uses a certified ticket to participate in routing and packet forwarding. Nodes without valid tickets are classified as misbehaving. They will be denied from any network access, even though they move to other locations. Thus, misbehaving nodes are "isolated" and their damage to the mobile ad hoc network is confined to their locality. The access control operation emphasizes multiple node consensus and fully localized instantiation. Since any individual node is subject to misbehaviors, this approach does not rely on any single node. Instead, the nature of cooperative computing in an ad hoc network is leveraged and the approach depends on the collective behaviors of multiple local nodes. Here multiple nodes in a local network neighborhood, typically one or two hops away, collaborate to monitor a node's behavior and determine whether it is well-behaving or misbehaving using certain detection mechanism of their choice. These local monitoring neighbors will renew the expiring ticket of a well-behaving node collectively, while a misbehaving node will be denied from ticket renewal or be revoked of its ticket. In this way, the functionality of a conventional access control authority, which is typically centralized, is fully distributed into each node's locality. Every node contributes to the access control system through its local efforts and all nodes collectively secure the network.

The localized approach does not need any hardware module for security. It does not assume anything about the packet size or type of traffic or the type of data. It not only detects the misbehaving nodes but also isolates them from the network. Average delay for ticket renewal is tolerable, because the node gets its ticket from its locality rather than going to a central server. There's no necessity for the node to rely upon a single node for getting a ticket or for renewal. So this approach is highly robust and scalable.

The localized approach requires that each node should get k tickets from its local neighborhood. It is possible to get k number of tickets in a highly populated network. But it is not possible when the number of nodes in a network is less. Thus the localized approach cannot be used in a sparse network. Moreover the protocol used in localized approach broadcasts the ticket request to all its neighbors, which increases the communication overhead.

The efficiency of the localized approach depends upon the coalition size k. i.e. the number of partial tickets that the node should get to access the network. The parameters viz. average delay, overhead and success ratio, which are used for simulation in [6], vary depending upon the k value. The k value is fixed as 5 in [6] based on the network size. This value does not change when number of nodes in the network increases or decreases. But this value will not work for all the networks. It is applicable only to a large network. For a sparse network, collecting 5 tickets from the neighborhood will cause more delay, because the nodes may not have sufficient number of neighbors in their locality. So in order to reduce the number of tickets a node should receive before successful access of the network, reputation mechanism can be used.

2.3.4 Reputation Based Localized Access Control

This paper [7] proposes a ticket-based approach, which uses reputation mechanism for evaluating the tickets. The nodes can access the network if they have a valid ticket. The tickets are obtained from the neighboring nodes, which have high reputation value. Initially the tickets are issued by a dealer. The tickets have expiration time. Once the expiration time reaches, the nodes have to renew their tickets. For renewing, the nodes will send the broadcast request to all its one-hop neighbors. On receiving the ticket renewal request, the neighbors have to decide whether to send a ticket or not by checking the reputation value of that node. Each node maintains the reputation value, by monitoring their behavior using any monitoring mechanism. When the requesting node receives a reply ticket, it checks the reputation value of the node, which has sent the reply. If the reputation value of the node is greater than a threshold value (this value is chosen based on the network behavior) then the requesting node accepts the ticket, otherwise it rejects the ticket from that node and looks for

other replies. Once it receives a ticket from higher reputation node, the node uses that ticket to prove its behavior and access the network. This makes the ticket obtaining process simpler.

Whenever a node issues a network access request, its ticket and the reputation value of the node, which gave the ticket, is verified. This ensures that two nodes cannot collaborate with each other and generate false tickets. Moreover other nodes will also monitor the behavior of these nodes. Nodes may try to generate their own tickets for communication. But this will be identified because the tickets are signed and verified using RSA algorithm. So this method is false proof and secure.

2.4 Hardware solutions to access control

2.4.1 Stimulating Cooperation in Self Organizing MANETs

L.Buttayan et al [2] focuses on packet forwarding and they address the problem of stimulating co- operation in self-organizing Mobile Ad-hoc Networks for civilian applications. This approach uses a tamper resistant hardware module called "security module". This security module maintains a nuglet counter. When the node forwards a packet for the benefit of other nodes, the nuglet counter is increased by one, when it sends its own data the counter is decremented by one. Every node has to maintain a +ve counter value in order to send its own data. The nuglet counter is protected from illegitimate manipulations by the tamper resistance of the security module. This approach ensures that the misbehavior is not beneficial and hence it should occur rarely only. But the availability of hardware module is not guaranteed in general.

2.4.2 Sprite

Sprite, was proposed by Zhong et al. in [8]. In Sprite, nodes keep receipts of the received/forwarded messages. When they have a fast connection to a Credit Clearance Service (CCS), they report all these receipts. The CCS then decides the charge and credit for the reporting nodes. In the network architecture of Sprite, the CCS is assumed to be reachable through the use of Internet, limiting the utility of Sprite Incentive based system for Access control

2.*4.3 CORE*

Michiardi and Molva [11] proposed a COllaborative Reputation (CORE) mechanism that also has a watchdog component for monitoring. Here the reputation value is used to make decisions about cooperation or gradual isolation of a node. Reputation values are obtained by regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request. In CORE the reputation value ranges from positive (+) through null (0) to negative (-). The advantage of this method is that having a positive to negative range allows good behavior to be rewarded and bad behavior to be punished. This method gives more importance to the past behavior and hence tolerable to sporadically bad behavior, e.g. battery failure. But the assumption that past behavior to be indicative of the future behavior may make the nodes to build up credit and then start behaving selfishly.

2. *4.4 CONFIDANT*

CONFIDANT[1] collects evidence from direct experiences and recommendations. Trust relationships are established between nodes based on collected evidence trust decisions are made based on this relationships. There are four interdependent modules: (a) monitor, (b) reputation system, (c) path manager, and (d) trust manager. Monitor collects evidence by monitoring the transmission of a neighbor after forwarding a packet to the neighbor. It then reports to the reputation system only if the collected evidence represents a malicious behavior. Reputation system changes the rating for a node if the evidence collected for a node's malicious behavior exceeds the pre-defined threshold value. Then, path manager makes a decision to delete the malicious node from the path. Also path manager assists the node in making decision such as whether to forward a received packet by checking the upstream node's identity (previous-hop) in the blacklist. Trust manager is responsible for forwarding and receiving recommendations to and from trustworthy nodes. Here recommendations are known as ALARM messages and trustworthy nodes are referred as friends. The ALARM messages received from friends are evaluated for trustworthiness before being sent to the reputation system. Trust manager assists in making trust decisions for the following, whether to: (a) provide and accept routing information, (b) accept a node as a part of route, and (c) take part in a route originated by some other node. CONFIDANT proves to show better network performance in presence of malicious nodes compared to DSR protocol.

3. Cut diamond with diamond

This section of the paper discusses the cut diamond with diamond approach in detail. It also gives the assumptions made in this paper.

3.1 Network Model

There are two types of nodes in an ad hoc network, well behaving and misbehaving. Well behaving nodes are those which cooperate in the network and utilize the services of the network without causing any problems to other nodes. Misbehaving nodes can be either selfish or malicious. Malicious nodes are those who disturb the network functionalities by *not cooperating* in the communication (e.g. by dropping others' packets, or redirecting routing packets etc). Selfish nodes

are nodes which just want to save their energy and thus they don't forward packets of other nodes. They don't intentionally disturb the network as malicious node. They want just to save their energy and CPU cycles. This paper considers a network with selfish nodes.

3.2 The Approach

This paper follows a different credit based mechanism to mitigate the effect of selfish nodes in the network. Credit based mechanism means the nodes are encouraged to behave selfless. But this paper follows a method in which the nodes are allowed to be selfish for a predetermined amount of time. This method is called as cut diamond with diamond because; here selfishness is mitigated by allowing nodes to be selfish. i.e. any node can be in selfish mode (not cooperating in communication) for a predetermined amount of time (say "t/2")by cooperating in the communication for a specific amount of time (say "t"). A node can behave selfish after participating in communication. For example, consider the following network.

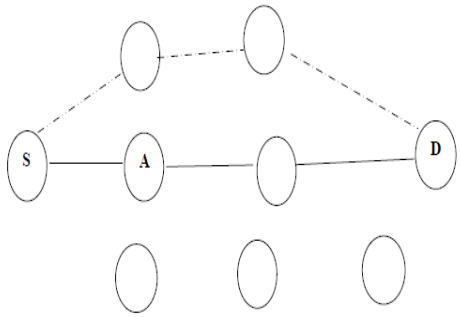

Fig 3.1 Example Networks Scenario showing two possible routes taken by source nodes S and Destination D.

Let A be an arbitrary node which was cooperating in forwarding others' data say between Source S and Destination D. If the communication was existing for some t amount of time, then the node A can be selfish (it need not cooperate in communication unless it has its own data for transmission) for t/2 amount of time. After t/2 amount of time, the node has to participate in the network. This scheme can be implemented by enforcing a method where the source node will select another route after time period t, by excluding the nodes that were helping it for communication (in the past time period t). This method is also called as virtual mobility when a source node selects another path as if it had moved to a new location. This scheme can be implemented over the DSR protocol.

DSR is Dynamic Source Routing protocol which has two phases route discovery and route maintenance. Route discovery process is initiated when a node wants to transmit data, but it doesn't have any newer route in its cache. Route maintenance is initiated when the existing route fails. In the existing DSR protocol, some changes can be incorporated to handle selfish nodes.

In DSR the destination receives more than one copy of route requests and it replies to the first coming route request (or it can sometimes look for shortest path before replying). Whatever may be, the destination should wait for some time interval to receive more than one route request from the same source and it should store them in a table, instead of discarding as in normal DSR. Route reply can be given to either the shortest path or to the first route request received. Now once the communication path is established, the destination should start receiving data packets from the source node. This path can be used for t amount of time. This t can be decided based on the number of nodes in the network, dynamicity of the network topology. Value of t can also be altered by periodically calculating the throughput of the network or it can be set as a global parameter in the network after some number of trials. Now after t time period the source node S will select another route from the cache memory which excludes nodes that were present in the route that was used by source for the past t time period. Node A can be selfish for t/2 amount of time. Either the source or destination node can advertise about A's participation in the communication to all other nodes and they can also inform that A will not participate in communication for t/2 amount of time and it should not be mistaken as a selfish node. This message can be sent as a broadcast to all nodes in the network.

But there are two problems to be addressed now. 1. The advertisement should be genuine which means one malicious node should not give false advertisement in favor of some other malicious node. (resulting in collaborative misbehavior) 2. The overhead incurred at source and destination in forming and sending the advertisement as a broadcast (a selfish node may deny to do it). The first problem can be overcome in network with selfish nodes. No node is malicious they are just selfish. The overhead at source and destination may be compromised because everybody expects a proper network functioning.

Many approaches in the literature provide methods to select a best path by avoiding selfish nodes in the route. But this causes over burden to the nodes which are not selfish. Every best route will select a route with good nodes only, thus causing over burden to those nodes which are not selfish. Our approach does not over burden the unselfish node instead it encourages all nodes to participate in the communication.

4. Simulation results

Ns2 [18] is used for simulation. Ns2 is a discrete event simulator, which is widely used for simulation of both wired and wireless networks. The modified DSR (our cut diamond with diamond approach) is compared with Dynamic Source Routing protocol. The parameters used for performance analysis are 1.Network Throughput. 2. Fairness. 3. Overhead. Network throughput is measured as the ratio of number of packets sent by the source to the number of packets received at the destination. It is measured for different network scenario by varying the number of nodes. The average mobility of the nodes is set as 15m/s for the scenarios and for creating the scenario random waypoint model is used. Number of nodes in the network is varied from 20-100.

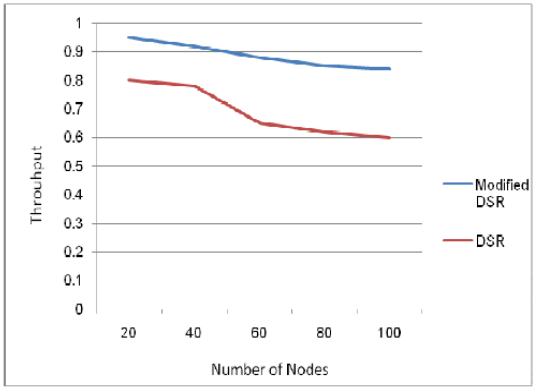

Fig 4.1 Throuhput of the network by varying number of nodes

Figure 4.1 shows that the throughput in a network with modified DSR is better compared to a normal DSR. This is because modified DSR encourages the nodes to be cooperative. So the effect of selfish nodes is reduced to a greater extent. Whereas with the normal DSR, there are selfish nodes in the network which drop packets of other nodes and reduces throughput. Fairness and profit is calculated as the ratio of amount of help given to the neighbors to the amount of help gained from the network. The unselfish nodes expect that their data should reach the destination without been dropped in between. The cut diamond with diamond approach encourages the nodes to be cooperative. This reduces the number of nodes turning selfish. So many nodes in the network receive a fair service. This is also clear from Fig 4.1. Overhead is the total number of bytes sent by the nodes in the scenario. Modified DSR uses a control packet to advertise that a node will not participate in the communication for a time period t/2 because it helped this node for a time period t. But this overhead is at acceptable level only.

Fig 4.2 shows that the overhead in modified DSR is more compared to normal DSR for different traffic scenarios. Effect of t (i.e the time for which a node is cooperative) under various scenario is shown in Fig 4.3. As mentioned earlier, t can be varied dynamically based on network throughput or set as a global parameter.

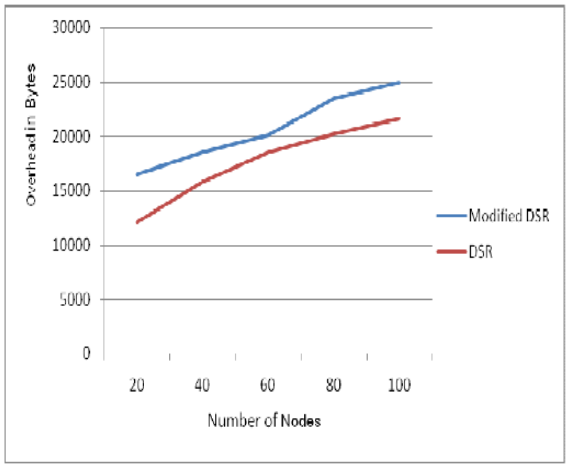

Fig 4.2 Overhead in bytes

In Fig 4.3 t is varied from 10s to 50s. The distance between source and destination is kept as 200m for simulation. This t depends on how long a communication happens between two nodes and how long an intermediate node is helping them for forwarding. For larger values of t the network throughput decreases in a sparse network. So for a long time the nodes are made passive. This is because the amount of time that a node can be selfish is given as t/2(average case). For a better performance amount of time that a node can behave selfish can be calculated using some other formula. For simplicity this paper uses t/2. When the number of nodes is less, the throughput is less compared to a dense network where number of nodes is more. This is because in a sparse network, if some nodes are helping for communication and goes for selfish state then communication in the network is disturbed. Anyway from the Fig 4.1 it is clearer that cut diamond with diamond approach out performs normal DSR.

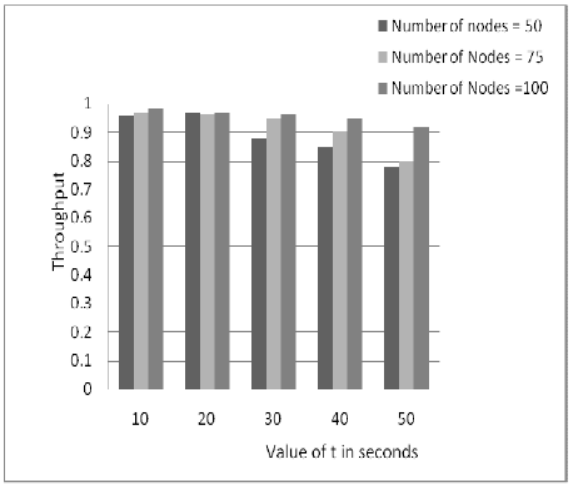

Fig. 4.3 Throuput for various values of t.

5. Conclusion and future work

This paper has addressed some of the existing approaches in the literature for access control in ad hoc networks. This paper also proposed a new method called cut diamond with diamond based on DSR protocol. The method dealt in the paper has a clear advantage of reducing the burden on well behaving nodes and increasing the chance of nodes cooperating for communication. This paper has suggested a mechanism for improving the cooperation of mobile nodes in an ad hoc network. But we have assumed that the network has no malicious nodes. In future this approach will be extended to work in a network with malicious nodes also.

References

[1] Buchegger, Sonja ; Le Boudec, Jean-Yves, " Performance Analysis of CONFIDANT Protocol: Cooperation of Nodes - Fairness in Dynamic Ad-Hoc Networks," in Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC). IEEE, June 2002.

[2]L.Buttayan and J.P.Hubaux, "Stimulating Cooperation in Self-organizing Mobile Ad-Hoc Networks," in ACM/Kulwer Mobile Networks and Applications, vol. 8, no.5, pp. 579-592, Oct 2003.

[3]S.Marti, T.Giuli, K.Lai and M.Baker, "Mitigating Routing Misbehavior in Mobile Ad-Hoc Networks," in Proc. ACM MOBICOM, pp. 255-265, 2000.

[4]L.Zhou, E.B.Schnder and R.Can Renese "COCA: a Secure Distributed on Line Certificate Authority," ACM Tran, Computer Sys, vol. 2, no. 4, pp. 329-368, Nov. 2002.

[5]G.Appenzeller, M.Roussopoulous, and M.Baker, "User-friendly access control for public network ports," in Proc. IEEE INFOCOM, pp. 699- 707, 1999.

[6]Haiyoun Luo, P.Zerfos, Songwu Lu, L.Zhang "URSA-Ubiquitous and Robust Access Control for Mobile Ad hoc Networks," IEEE/ACM Transactions on Networking, vol. 12, no. 6, pg. 1049-1063, Dec. 2004.

[7] Sangheethaa Sukumaran, Elijah Blessing, " Reputation Based Localised access control for mobile ad-hoc networks", in Lecture notes in Computer Science, Volume 4104/2006, ISSN 0302-9743.

[8] S.Zhong, J.Chen, and Y.R.Yang, " Sprite: A Simple, Cheat Proof,Credit based System for Mobile Ad Hoc Networks", in roceedings of INFOCOM, Apr. 2003.

[9] Houssein Hallani and Seyed A. Shahrestani," Mitigation of the Effects of Selfish and malicious Nodes in Ad-hoc Networks" in WSEAS TRANSACTIONS on COMPUTERS Issue 2, Volume 8, PP.No206- 221, ISSN: 1109-2750,February 2009

[10] C.E. Perkins and E.M.Royer, " Ad hoc On-Demand Distance Vector Routing", in Proceedings of the 2nd IEEE workshop on Mobile Computing Systems and Applications., New Oreleans, LA, Feb.1999.

[11] Pietro Michiardi and Refik Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia, 2002.

[12] Animesh Kr Trivedi1, Rishi Kapoor1, Rajan Arora1, Sudip Sanyal1 and Sugata Sanyal , " RISM - Reputation Based Intrusion Detection System for Mobile Adhoc Networks" Available from link profile.iiita.ac.in/aktrivedi_b03/rism.pdf

[13] M. Tamer Refaei, Vivek Srivastava, Luiz DaSilva, Mohamed Eltoweissy, " A Reputation-based Mechanism for Isolating Selfish Nodes in Ad Hoc Networks",in Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05) , 2005

[14] Kejun Liu, Jing Deng, Pramod K. Varshney, and Kashyap Balakrishnan "An Acknowledgment based Approach for the Detection of Routing Misbehavior in MANETs", September 2006,

[15] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, vol. 353, pp. 153-181, 1996.

[16] T.V.P. Sundararajan, Dr.A.Shanmugam, " Performance analysis of selfish node aware routing protocol for Mobile Ad-Hoc Networks" in ICGST-ICNIR Journal, volume 9, Issue 1, July 2009.

[17] William Kozma Jr. and Loukas Lazos , "Reactive Identification of Misbehavior in Ad Hoc Networks Based on Random Audits", in proceedings of PP. Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. pp.612-614, June 2008.

[18]http://www.isi.edu/nsnam/ns/ns-documentation.html.

Sangheethaa Sukumaran is doing part time research in Information Technology at Anna University of Technology, Coimbatore. She has completed B.E in Information Technology from Sri Krishna College of Engineering and Technology Coimbatore with University rank in the year 2003. She has completed M.E in Network and Internet Engineering from Karunya University with University Rank in the year 2006. She has worked in Coimbatore Institute of Technology, Coimbatore for about 2.5 years and in SSN College of Engineering – Chennai for about a year. Currently she is working as Asst. Professor in Computer Science and Engineering Department of Vedavyasa Institute of Technology, Calicut, Kerala from the year 2008. Her teaching experience spans to 6 years. Her research area includes ad hoc networks, security in mobile communications etc. She has published 4 papers in International journals. 2 papers in IEEE Digital library and presented papers in 6 international conferences.

Dr. Venkatesh Jaganathan is Associate Professor in the School of Management Studies, Anna University of Technology Coimbatore, Tamilnadu, India. He has done Ph. D in Management and Ph.D in International Business. He has more than 13 years of teaching and research experience. He is associated as reviewer board member and reviewer in many reputed National and International Journals. He has won the "Best research paper award" from All India Management Association (AIMA), New Delhi in 2007. He is a member in International Advisory Council, at University of Atlanta, USA and Life member in Indian Society for Technical Education, New Delhi. He is also Fellow Member of Indian School of Labour Education and an Accredited Management Teacher (AMT) awarded by AIMA. He has published 63 papers in international journals, 58 in National journals. He authored 2 books. His research area includes Systems and International Business.

Arun Korath is working as Asst. Professor in the Department of Computer Science in Vedavyasa Institute of Technology, Kerala. He has done his Bachelor of Engineering from Bharathiar University, Master of Engineering from Karnataka University, Master of Business Management from Kannur University. He has more than 6 years of teaching and research experience. He has published 6 papers in various reputed international journals. He is doing part time research in management studies from Anna University

.