AAS: An Authenticated Acknowledgement-Based Scheme for Preventing Selfish Nodes in Mobile Ad Hoc Networks

M. Gunasekaran[1], P. Sampath[2], B. Gopalakrishnan[2]
[1]Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India
Email: mguna@rediffmail.com
[2]Bannari Amman Institute of Technology, Sathyamangalam, Taml Nadu, India
Email :{ prof_sampth@yahoo.com, bgopal1977@gmail.com }

Abstract - Security is a critical problem when implementing Mobile Ad Hoc Networks (MANETs) and is widely acknowledged. This paper describes the effects of selfish nodes in MANETs. An Ad Hoc Network is a collection of wireless mobile hosts forming a temporary network without the aid of any centralized administration or standard support services. One of the different kinds of misbehavior is node selfishness. A selfish node wants to preserve its own resources while using the services of others and consuming their resources, such misbehaving nodes participate in the route discovery and maintenance phase but refuse to forward data packets, which degrades routing performance. One way of preventing selfishness in MANETs is the detection and exclusion mechanism. In this paper, we focus on authenticated scheme, which preserves communication privacy and mitigates selfish nodes in MANETs. The simulation results are presented here that shows the negative effects which selfish nodes causes in MANETs.

Index Terms: Mobile Ad-hoc Networks (MANETs), selfish nodes, routing, network security, Dynamic Source Routing (DSR).

I. INTRODUCTION

Mobile Ad Hoc Network (MANET) is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes itself, i.e., routing functionality will be incorporated into mobile nodes.

Ad hoc environments introduce two main problems not commonly faced by traditionally fixed network routing protocols. The problems occur due to the lack of fixed infrastructure support and the frequent changes in network topology. MANETs support dynamic communication environments and facilitate large-scale, real-time data processing in complex environments. Ad hoc networks require no fixed infrastructure, such as a base station or access points. Networks can be established inexpensively, as needed.

There exist two types of MANETs namely open and closed [1]. Closed MANETs do not have cooperation problems, since all nodes work towards a common goal and can easily be controlled. Open MANETs contain nodes that share their resources to ensure global connectivity but they may have different goals. The nodes in open MANETs are operated by multiple users, and therefore they need not be forced to cooperate.

Selfish Nodes **[2][3][12]:** Each node relies on the routing information it receives from other nodes in order to determine appropriate routes. Open environment of MANET may lead to selfish nodes (misbehaving nodes). These selfish nodes use the services provided by the network, but they do not contribute to the network. Such nodes have a negative impact on the overall performance of the network. The selfish nodes spend their energy only for their own needs, such as sending packets to destination, but they do not forward packets from other nodes, because they see this process as wasted energy. If the selfish node drops all the packets, then the routing algorithm will eventually find another route around this node. If the selfish node drops only a certain percentage of packets, the other nodes get an impression that it is not done intentionally but only due to the interferences.

We propose the authenticated approach to prevent and mitigate the negative effects of selfish nodes. The basic idea is, when a node forwards a data packet over the next hop, the destination node of the next hop will send back an authenticated acknowledgement packet to indicate that the data packet has been received.

The rest of the paper is organized as follows: In Section II, we summarize the various schemes that have been proposed and studied in the literature to prevent selfish nodes. In Section III, we describe the details of proposed AAS scheme. Section IV presents the performance analysis and simulation results. Finally in Section V, we conclude the paper and discuss the plan for future work.

II. Related Works

A. Incentive-Based Schemes

Some algorithms employ credit based schemes to discourage selfish nodes. These scheme works on the principle that you get credit for every packet you forward, and pay some of the credit to send a message yourself. We need to take into account the malicious nature of non-cooperative nodes, which may tamper with their own credit count. To avoid this, special hardware would be needed for keeping track of the credit.

In Sprite [4], a credit based scheme proposed by Zhong, each node maintains receipts for messages which are received and forwarded. When the nodes get a connection to a credit clearance service, they report those credits, and based on the decision taken by the CCS the nodes need to pay or they may be rewarded with real money. Since this uses an external party for the payment, it may not be useful for all scenarios.

In [3], Buttyan and Hubaux, used the concept of beans (nuggets) as payments for packet forwarding. They proposed two models: packet purse model and packet trade model. In packet purse model, beans are loaded into the packet before it is sent. The sender puts a certain number of beans on the data packet to be sent. Each intermediate node earns beans in return for forwarding the packet. If the packet exhausts, the beans in it drops before reaching its destination. In the packet trade model, each intermediate node buys the packet from the previous node for some nuggets. Thus, each intermediate node earns some beans for providing the forwarding service and the overall cost of sending the packet is borne by the destination.

B. Reputation-based schemes

This is an approach designed to encourage cooperation based on reputation. In this scheme, the nodes that detect misbehavior inform the other nodes in order to exclude the suspicious node from the network. This can be problematic, as nodes may have to choose a route passing the malicious node in order to inform the cooperative nodes. The uncooperative node could simply drop this message.

Marti et al.[2] proposed a scheme, which contains two modules: watchdog and pathrater. The watchdog module overhears the medium to check whether the next-hop node faithfully forwards the packet or not. At the same time, it maintains a buffer of recently sent packets. A data packet is cleared from the buffer when the watchdog overhears the same packet being forwarded by the next hop node over the medium. If a data packet remains too long in the buffer, the watchdog module accuses the next-hop neighbor to be misbehaving. Thus, the watchdog enables misbehavior detection at the forwarding level as well as the link level. Based on watchdog's accusations, the pathrater rates every path in its cache and subsequently chooses the path that best avoids misbehaving nodes. However, the watchdog technique may fail to detect misbehavior in the presence of ambiguous collisions, receiver collisions, limited transmission power, false misbehavior and partial dropping.

The CONFIDANT protocol proposed by Buchegger et al.[5] consists of four important components: the Monitor, the Reputation System, the Path Manager, and the Trust Manager. They perform the vital functions of neighborhood watching, node rating, path rating, and sending and receiving alarm messages, respectively. Each node continuously monitors the behavior of its first-hop neighbors. If a suspicious event is detected, details of the event are passed to the Reputation System. Depending on how significant and how frequent the event is, the Reputation System modifies the rating of the suspected node. Once the rating of a node becomes intolerable, control is passed to the Path Manager, which accordingly controls the route cache. Warning messages are propagated to other nodes in the form of an *Alarm* message sent out by the Trust Manager.

III. System Description

A. Selfish Node Model

We present the selfish node model considered in this paper in the context of the DSR protocol [7].

We focus on the following routing misbehavior: A selfish node does not perform the packet forwarding function for data packets which are unrelated to them. However, it operates normally in the route discovery and the route maintenance phases of the DSR protocol. Since such misbehaving nodes participate in the route discovery phase, they may be included in the routes chosen to forward the data packets from the source. The selfish nodes, however, refuses to forward the data packets from the source and thus leads to confusion.

In guaranteed services such as TCP, the source node may either choose an alternate route from its route cache or initiate a new route discovery process. The alternate route may again contain selfish nodes and, therefore, the data transmission may fail again. The new route discovery phase will return a similar set of routes, including the selfish nodes. Eventually, the source node may conclude that routes are unavailable to deliver the data packets. As a result, the network fails to provide reliable communication for the source node even though such routes are available. In best-effort services such as UDP, the source simply sends out data packets to the next-hop node, which forwards them on. The existence of selfish nodes on the route will cut off the data traffic flow. The source has no knowledge of this at all.

In this paper, we propose the **Authenticated Acknowledgement Scheme (AAS)** to detect such selfish nodes. Routes containing such nodes will be eliminated from consideration. The source node will be able to choose an appropriate route to send its data.

B. The TWOACK Scheme

In the TWOACK scheme [6], TWOACK packets are sent for every data packet received. Acknowledging the received data packets gives the AAS scheme better performance with respect to routing overhead. The AAS scheme has an authentication mechanism to make sure that the acknowledgement packets are genuine. The TWOACK scheme provides no authentication for the acknowledgement packets which has been sent form the receiver but the proposed scheme provides an authentication mechanism which improves the performance of the network and cost as compared to the TWOACK.

C. AAS Scheme

The AAS scheme is a network-layer technique to detect the selfish nodes and to mitigate their effects. It can be implemented as an add-on to existing routing protocols for MANETs, such as DSR. The AAS scheme detects misbehavior through the use of a new type of authenticated acknowledgment scheme termed AAS, which assigns a fixed route of two hops (three nodes) in the opposite direction of the data traffic route.

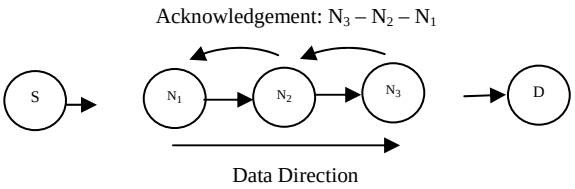

Fig.1. Direction of Data and Acknowledgement

The fig.1 illustrates the operation of the AAS scheme. Suppose that N_1, N_2, and N_3 are three consecutive nodes (triplet) along a route, the route from a source node, S, to a destination node, D, is generated in the route discovery phase of the DSR protocol. When N_1 sends a data packet to N_2 and N_2 forwards it to N_3, it is unclear to N_1 whether N_3 receives the data packet successfully or not. Such an ambiguity exists even when there are no misbehaving nodes. The problem becomes much more severe in open MANETs with potential selfish nodes.

The AAS scheme requires an explicit acknowledgment to be sent by N_3 to notify N_1 of its successful reception of data packets. When node N_3 receives the data packets successfully, it sends out a acknowledgement packet over two hops to N_1 (i.e., the opposite direction of the routing path as shown in Fig.1), with the ID of the corresponding data packets. The triplet [N_1-> N_2-> N_3] is derived from the route of the original data traffic. Such a triplet is used by N_1 to monitor the link N_2->N_3. For convenience of presentation, we term N_1 in the triplet [N_1-> N_2-> N_3] the acknowledgement packet receiver and N_3 the acknowledgement packet sender.

D. Packet Authentication

Since the acknowledgement packets are forwarded by intermediate node without proper protection, a selfish node N_2 can simply fabricate the acknowledgement packets and claim that they were sent by node N_3. Therefore, an authentication technique is needed in order to protect the acknowledgement packets from being forged.

When a node wishes to communicate with another node, a methodology is performed by the sending and receiving nodes, which ensures authentication and integrity. For complete confidentiality, a method authenticates packets that are transmitted serially in a network. A current password is selected for a current packet to be transmitted. The current packet includes current data. A one-way/one-time [13] [14] hash function is applied to the current password to form a current tag. A next password is selected for a next packet that includes next data, and the one-way/one-time hash function is applied to the next data, the next tag, and the current password to obtain a hashed value. The current packet is then transmitted to include the hash value, the current data, the current tag, and a previous password of a previous transmitted packet to authenticate the current data.

E. Algorithm of the AAS Scheme

The triplet N_1-> N_2-> N_3 in Fig.1 has been used to illustrate algorithm and this algorithm run on each of the sender and receiver of the acknowledgement packet.

Acknowledgement Packet Sender (Node N_3)

Step1: Send authenticated acknowledgement packet from N_3 to node N_1
Step2: Initialize the counter of forwarded data packets and acknowledgement packets.
Step3: If the data packet is received then
Step4: Increase the counter of the received packets at node N_3
Step5: If the data packet needs to be acknowledged
Step6: Prepare Message Authentication Code (MAC)
Step7: Prepare acknowledgement packet with ID and MAC
Step8: Send the acknowledgement packet
Step9: Increase the counter of acknowledgement packet at node N3
Step10: Repeat steps from 3 to 9 for all the data packets

Acknowledgement Packet Receiver (Node N_1)

Process 1 **(**receiving authenticated element**)**

Step1: If the authenticated element received by N_1 from acknowledgement packet sender N_3 then
Step2: Record the authenticated element received from N_3

Process 2 (receiving acknowledgement packets)

Step3: Start the observation at randomly selected time
Step4: Initialize the LIST (data structure for holds the data IDs), counter of forwarded data packets, and the counter of missing acknowledgement packets to zero (maintained at the receiving node N_1)
Step5: If the data packet is forwarded then Add the data ID to the LIST, Increase the counter of the forwarded packets, and record the timer
Step6: If the acknowledgement packet is received then check the availability of the data ID for the acknowledgement packet received and check the validity of authenticated element
Step7: Remove the ID from the LIST and clear the timer
Step8: If the data ID of the acknowledgement packet is not received then remove the ID from the LIST and increase the misbehavior counter
Step9: If the observation period expires send misbehavior report
Step10: Repeat the entire steps for all the packets

IV. Performance Evaluation

A. Simulation Methodology

In this section, we present the evaluation of the AAS scheme through the network simulator (ns-2) [15]. We have modified the DSR protocol in ns-2 to simulate the selfish nodes described in Section I.

For the communication pattern, we implement CBR transmissions between pairs of nodes. The source and destination for each pair are randomly chosen such that there is no limit on the number of sources or destinations that a node can host. The AAS scheme is analyzed under varying traffic conditions by running simulations for networks. We measure the following evaluation metrics for different percentage of selfish nodes in the network. Packet Delivery Ratio: defined as the ratio of the number of packets received at the destination node to the number of packets sent by the source node. Routing Overhead: defined as the ratio of the amount of routing-related transmissions in bytes to the amount of data transmissions in bytes in a network.

B. Simulation Environment

The network simulator (ns-2) helps us to evaluate the communication aspects of our method, such as route discovery and average route load in ad hoc wireless network. Simulation with the following parameters has been done to study the effects of the node selfishness, monitoring technique and proposed approach on the performance of MANETs.

TABLE: 1
Simulation Parameters

Parameter	Value
Node distribution	[700 x 700], [1000 x1000]
Node Mobility	[0, 10], [10, 20] m/s
Data rate (traffic)	2 x 4kb
Pause Time	10 sec, 60 sec
Simulation time	180s
Nodes	Tested on 40, 60 nodes
Misbehaving nodes	10, 20, 30

Scalability: We examine the scalability of the system in terms of number of nodes, fraction of misbehaving nodes, mobility, and distribution area. We have chosen different values like 0- 40 number of nodes, 0 to 40% of misbehaving nodes, mobility from 0m/s to 20m/s and for distribution area from 700 sq meters to 1000 sq meters.
Availability: We change the different parameters to see whether the design provides security service to mobile hosts.
Robustness: There are some provisions in the proposed design to ensure robustness of the system against wrong observations and wrong accusations, i.e., maliciously excluding cooperative nodes by spreading the rumor that they misbehave.

C. Simulation Results

In this section we analyze the results obtained from simulation experiments carried out in ns-2 to study the impact of selfish nodes on the network and to evaluate the network with different approaches under different conditions.

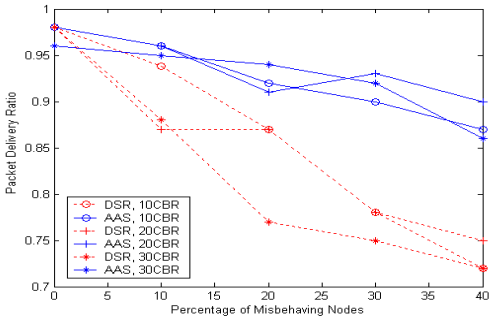

Fig. 2 Packet Delivery Ratio of AAS and DSR

Fig.2 shows the comparison of network performance in terms of network throughput when we use different mobility of the mobile nodes in the presence of selfish nodes. Initially nodes are uniformly distributed and node mobility are emulated according to the random way point model. We run simulations with the assumption of selfish nodes as 0, 10, 20, and 30 with pause time to 10ms with random source and destination pairs through the simulations. And also compares the packet delivery ratio of the original DSR scheme, and the proposed AAS scheme. The

percentage of selfish nodes in the network varied from 0 to 40%. The packet delivery ratio decreases as more as nodes in the network are selfish. This is due to the problem of missing routes and the overhead of searching for alternative routes. When compared with the original DSR scheme, the proposed AAS scheme maintains a relatively high packet delivery ratio (Throughput).

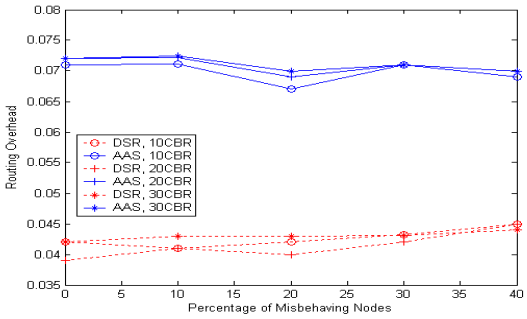

Fig. 3 Routing Overhead of AAS and DSR

Fig.3 indicates that the overhead transmission increases if we increase the pause time from 10sec to 60sec. And for mobility if we increase the speed of nodes movement, overhead transmission decreases.

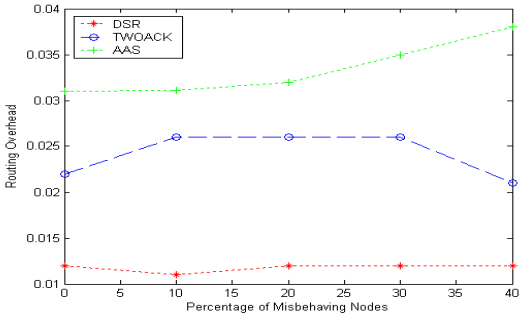

Fig. 4 Routing Overhead of AAS, TWOACK, and DSR

Fig. 4 compares the routing overhead of the AAS, TWOACK and DSR schemes for different percentages of selfish nodes. It can be observed that the routing overhead of the AAS is relatively higher than the TWOACK scheme and the original DSR scheme. This is due to the increase of data traffic being delivered successfully in the AAS scheme.

V. Conclusions and Future Work

In this paper we have described the possible extension to DSR protocol for preventing the effects of selfish nodes in MANETs. The scheme provides a solution to security support in MANETs and mitigates the selfish nodes. We have also evaluated the impact of selfish nodes in the network in terms of throughput, routing overhead and end-to-end delay. Simulation based analyses of this technique show improvement in throughput by detection and exclusion of the selfish nodes, but it increases the end-to-end delay as well as overhead transmission.

Our model motivates mainly the service availability in each network environment and this is crucial in supporting ubiquitous service for mobile users.

Future Work

The reputation value can be calculated more accurately through the accurate estimation procedure. Network performance can also be improved using dedicated nodes as independent network operators. In this paper we have discussed only about excluding the selfish nodes to increase the throughput. In future we can also identify the weaken links in the network and exclude them to improve the network performance further.

References

[1] H. Miranda and L. Rodrigues, “Preventing Selfishness in Open Mobile Ad Hoc Networks,” Proc. Seventh CaberNet Radicals Workshop, Oct. 2002.
[2] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” Proc. MobiCom, Aug. 2000.
[3] L. Buttyan and J.P. Hubaux, “Enforcing Service Availability in Mobile Ad-Hoc WANs,” Proc. MobiHoc, Aug. 2000.
[4] S. Zhong, J. Chen, and Y.R. Yang, “Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks,” Proc. INFOCOM, Mar.-Apr. 2003.
[5] S. Buchegger and J.-Y. Le Boudec, “Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks,” Proc. MobiHoc, June 2002.
[6] K. Balakrishnan, J. Deng, and P.K. Varshney, “TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks,” Proc. IEEE Wireless Comm. and Networking Conf. (WCNC ’05), Mar. 2005.
[7] D. Johnson, D. Maltz, Y.C. Hu, and J. Jetcheva, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR),” Internet draft, Feb. 2002.
[8] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam. Authentication Scheme for Ad Hoc and Sensor Wireless Networks. In *Proc. of Tenth International World Wide Web Conference (WWW10)*, May 2001.
[9] Y. Hu, A. Perrig, and D.B. Johnson, “Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks,” Proc. MobiCom, Sept. 2002.
[10] L. Zhou and Z.J. Haas, “Securing Ad Hoc Networks,” IEEE Network Magazine, vol. 13, no. 6, 1999.
[11] B. Awerbuch, D. Holmer, C.-N. Rotaru, and H. Rubens, “An On-Demand Secure Routing Protocol Resilient to Byzantine Failures,” Proc. ACM Workshop Wireless Security (WiSe), Sept. 2002.
[12] D. Eastlake and P. Jones, “RFC 3174—US Secure Hash Algorithm1 (SHA1),” technical report, Motorola and Cisco Systems, Sept. 2001.
[13] Y. Hu, D.B. Johnson, and A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks”, Ad Hoc Networks, vol.1, no. 11, pp 175-192, 2003.
[14] L. Lamport, “Password Authentication with Insecure Communication”, Comm. ACM, vol. 1, no. 11, pp. 770-772, Nov. 1981.
[15] “The Network Simulator-2,” http://www.isi.edu, 2005.