www.icgst.com

Performance Analysis of Selfish Node Aware Routing Protocol for Mobile Ad Hoc Networks

T.V.P.Sundararajan , Dr.A.Shanmugam

Bannari Amman Institute of Technology, Anna University Sathyamangalm,INDIA

E-mail : tvpszen@yahoo.co.in, principal@bitsathy.ac.in

Abstract

A mobile ad hoc network consists of nodes that move arbitrarily and form dynamic topologies. The nature of the open structure and scarcely available battery-based energy, node misbehaviors may exist. An individual mobile node may attempt to benefit from other nodes, but refuse to share its own resources. Such nodes are called selfish or misbehaving nodes. These selfish nodes may severely affect the performance of network. In this paper, we propose the selfish aware **AODV+2ACK** model to detect routing misbehavior and to mitigate their adverse effect. The main idea of the **AODV+2ACK** model is to send two-hop acknowledgment packets in the opposite direction of the routing path. In order to reduce additional routing overhead, only a fraction of the received data packets are acknowledged in the **AODV+2ACK** scheme. Thus, we propose to investigate performance of **AODV+2ACK** model even in the presence of selfish nodes and the same has been compared with the ordinary AODV protocol. The simulation study in this paper brings out that the proposed protocol has higher performance than existing AODV and DSR, in terms of throughput, packet delivery ratio and end-end delay.

Keywords *— Ad Hoc Networks, selfish nodes, QoS routing, security*

1. Introduction

A. Ad hoc networks

Ad hoc networks offer methods for self-organizing networks. All nodes act both as participants and routers. Due to node mobility, the routing topology may be subject to constant change. Thus, ad hoc routing poses special requirements to routing protocols. Some well-known routing protocols include DSR [1] and AODV [2].

B. Attack scenarios in MANETs

Any secure networking system should provide the following six properties: Secrecy, authenticity, integrity, availability, non-repudiation, and access control. All attacks on a computer system are a violation of one or more of these security goals. There are a number of well-known attacks [3] on distributed computer systems; these include i) Denial of Service ii) Information theft iii) Intrusion iv) Tampering.

Both the security goals and most of the attacks known from common networks apply to Ad hoc networks, too. Since most network participants are mobile devices, they can easily be stolen (or are lost otherwise).Thus, an attacker can easily gain all data stored (e.g. passwords, cryptographic keys, etc.) on a node. As a consequence, the overall security of an ad hoc network must not depend on a single component.

In mobile networks, radio transmission is the most common means of communication. Eavesdropping on a node is far easier than in wired networks. Since intermediate nodes no longer belong to a trusted infrastructure, but may be eavesdroppers as well, consequent end-to-end encryption is mandatory.

Next, as all nodes in an Ad hoc network cooperate in order to discover the network topology and forward packets, denial of service attacks on the routing function are very easy to mount. Nodes may create stale or wrong routes, creating black holes or routing loops.

Furthermore, in Ad hoc networks exists a strong motivation for non-participation in the routing system [4], [5], [6]. Both the routing system and the forwarding of foreign packets consume a node's battery power, CPU time, and bandwidth, which are restricted in mobile devices. Consequently, selfish nodes may want to save their resources for own use. [7]

There are three main causes for a node not to work according to the common routing protocol:

• Selfish nodes try to save their own resources, as described above.

• Malicious nodes are trying to sabotage other nodes or even the whole network, or compromise security in some way.

• Malfunctioning nodes are simply suffering from a hardware failure or a programming error. Although this is not an attack, they may cause severe irritation in the routing system of an ad hoc network.

Most of the proposed routing protocols were not designed considering security as a goal. Some approaches [8] like *watchdog* and *pathrater* were good starting point for research in detecting misbehaving nodes. But it had many weaknesses.

The *watchdog* technique is based on passive overhearing. Unfortunately, it can only determine whether or not the next-hop node sends out the data packet. The reception status of the next-hop link's receiver is usually unknown to the observer.

Nodes operate in a promiscuous mode wherein the *watchdog* module overhears the medium to check whether the next-hop node faithfully forwards the packet. 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 in the buffer for too long, the *watchdog* module accuses the next hop neighbor of misbehaving. Thus, the *watchdog* enables misbehavior detection at the forwarding level as well as the link level. Based on the watchdog's accusations, the pathrater module rates every path in its cache and subsequently chooses the path that best avoids misbehaving nodes. Due to its reliance on overhearing, however, the *watchdog* technique may fail to detect misbehavior or raise false alarms in the presence of ambiguous collisions, receiver collisions, and limited transmission power.

In order to mitigate the adverse effects of routing misbehavior, the misbehaving nodes need to be detected so that these nodes can be avoided by all well-behaved nodes.

In this paper, the AODV+**2ACK** routing protocol has been implemented and the performance were analyzed based on the following performance metrics.

• Average end to end delay
• Throughput
• Packet delivery ratio.
• Routing overhead.

The rest of the paper is organized as follows. In Section 2, We first present related works in the area. In Section 3 we deals with AODV+2ACK routing misbehavior model , 2 acknowledgment scheme and authentication scheme used. In section 4, Performance Analysis metrics, the Glomosim simulator and its simulation environment with assumptions. In section 5, we present our simulation results that compare AODV scheme with AODV+2ACK scheme. We conclude the work in section 6.

2. Related Work

The security problem and the misbehavior problem of wireless networks including MANETs have been studied by many researchers. Various techniques have been proposed to prevent selfishness in MANETs.

Marti et al.[8] proposed a *watchdog* and *pathrater* scheme to improve the throughout of an ad hoc network in the presence of misbehaving node. *Watchdog* keeps track of misbehaving nodes. *Pathrater* avoids routing through those misbehaving nodes.

Yang et al.[9] extended AODV with a self –organized security approach. A token is utilized for authentication within the network, which is issued with a decentralized scheme. Only with valid token, can a node participate in route discovery and data packet delivery. Their protocol does not assume the existence of centralized trusted server and is suitable for ad hoc network situations.

Peng Ning et al. [10] gave systematic analysis of insider attacks against mobile ad hoc routing protocols. It evaluates AODV as a case study. Analysis results are classified into according to type of insider attack including route disruption, route invasion, node isolation and resource consumption. Although attacks mentioned in this paper are evaluated against AODV routing protocol, most of the other routing protocols are susceptible to similar attacks. Collaborative Voting System (CVS) is an effort to overcome limitations of watchdog mechanism while detecting Byzantine faults. CVS approach has few constraints like computational overhead which will consume more energy and communication overhead which will increase network traffic.

A credit-based scheme, termed Sprite, was proposed by Zhong et al. [11]. In Sprite, nodes keep receipts of the received/forwarded messages. In this scheme, when a node receives a message, the node keeps a *receipt* of the message. Later, when the node has a fast connection to a Credit Clearance Service (CCS), it reports to the CCS the messages that it has received/forwarded by uploading its receipts. The CCS then determines the charge and credit to each node involved in the transmission of a message, depending on the reported receipts of a message. In the network architecture of Sprite, the CCS is assumed to be reachable through the use of the Internet, limiting the utility of Sprite.

The design of this system need to address two main issues. First, since there is no tamper-proof hardware at any node and the charge and credit are based on the reports of the selfish nodes, a selfish node (or even a group of colluding node) may attempt to cheat the system to maximize its expected welfare. Second, a node should receive enough credit for forwarding a message for another node, so that it can send its own messages with the received credit, unless the resource of the node itself is extremely low.

The main problem with credit-based scheme is that they usually require some kind of tamper-resistant hardware and/or extra protection for the virtual currency or the payment system. We focus on reputation-based techniques in this paper instead.

The *CONFIDANT* protocol proposed by Buchegger and Le Boudec in [12] is another example of reputation-based schemes. The protocol is based on selective altruism and utilitarianism, thus making misbehavior unattractive. *CONFIDANT* 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.

The Monitor component in the *CONFIDANT* scheme observes the next hop neighbor's behavior using the overhearing technique. This causes the scheme to suffer from the same problems as the *watchdog* scheme.

The 2ACK scheme proposed by K. Balakrishnan et al.[13] does not rely on end-to-end acknowledgment. Such an acknowledgment scheme may not exist in some traffic flows (such as UDP). Instead, the 2ACK scheme tries to detect misbehaving links as the links are being used. Such a proactive detection approach results in quicker detection and identification of misbehaving links. Note that it may be beneficial to include end-to-end acknowledgments in the 2ACK scheme. In such a combined scheme, the 2ACK transmission and the monitoring processes are turned on only when routing performance degrades. It will further reduce the routing overhead of the 2ACK scheme.

Kejun Liu et al.[14] propose the 2ACK scheme to mitigate the adverse effects of misbehaving nodes. The basic idea of the 2ACK scheme is that, when a node forwards a data packet successfully over the next hop, the destination node of the next-hop link will send back a special two-hop acknowledgment called 2ACK to indicate that the data packet has been received successfully. Such a 2ACK transmission takes place for only a fraction of data packets, but not all. Such a "selective" acknowledgment1 is intended to reduce the additional routing overhead caused by the 2ACK scheme. Judgment on node behavior is made after observing its behavior for a certain period of time.

3. Routing Model

In this section, we describe the problems caused by routing misbehavior. But first, we summarize background of AODV, DSR, assumptions and notations used throughout this paper.

A. Review of MANET Routing Protocols

Routing protocols for MANETs have been classified according to the strategies of discovering and maintaining routes into three classes: proactive, reactive, and hybrid. Of course, each routing protocol reacts differently to node mobility and density. A routing protocol for MANETs is usually evaluated in terms of performance metrics that are end to-end delay, overhead, throughput and data delivery ratio. This section outlines the main features of each class. Also, a brief summary about the protocols that have been used in simulations is given.

Proactive Routing Protocols: Proactive routing protocols acquire routing information periodically and store then in one or more routing tables.

Reactive Routing Protocols: Reactive routing protocols discover or maintain a route as needed. This reduces overhead that is created by proactive protocols. Flooding strategy is used to discover a route.

Ad hoc on demand Distance Vector: AODV is a hop-by-hop routing protocol, which introduces a more dynamic strategy to discover and repair route when compared to DSR. Destination sequence numbers are used to avoid the problem of infinite loops. AODV maintains only active routes to reduce overheads and control traffic. This protocol is applicable for different levels of node density, mobility and loads. It is suitable for scenarios with moderate mobility and density networks.

Dynamic Source routing: DSR is a reactive source routing protocol . It discovers routes on demands using route discovery and maintenance strategy. Multiple routes are applied to achieve load balancing and to increase robustness. DSR can operate well with high mobility nodes because it can recover from routes failure quickly. It can support up to one hundred node which means it can work well over medium network density.

B. Assumptions

This section outlines our assumptions regarding the properties of the physical and network layers. Throughout this paper, we assume bidirectional communication. Such symmetry of links is needed for the transmission of the designed 2ACK packets. Our scheme works with on demand routing, such as AODV [15], [16]. We further assume that there is no collusion among misbehaving nodes. We argue that misbehavior caused by selfishness is usually limited to individual nodes in MANETs.

C. Notations

We use the following notations throughout the paper:

- X * Y: the size of network area.
- N: the total number of nodes in the network.
- R: the transmission range of each node. We assume that the transmission of all nodes is omni-directional and the transmission range is homogeneous. We assume R = 250 meters in our simulations.
- Vm: the maximum speed of a mobile node.
- h: the average number of hops from the source node to the destination node.
- l : the expected progress of one-hop transmission.
- d: the expected distance between the source node and the destination node.
- ρ_m : the fraction of nodes that are misbehaving. This is also the probability of a node being a misbehaving node. The misbehaving nodes are selected among all network nodes randomly. In our simulations, ρ_m ranges from 0 to 0.4.
- R_{mis}: the threshold to determine the allowable ratio of the total number of 2ACK packets missed to the total number of data packets sent.


- R_{ack} : the acknowledgment ratio, the fraction of data packets that are acknowledged with 2ACK packets (maintained at the 2ACK sender).
- τ: the value of *timeout*, beyond which time a data packet will be considered to be unacknowledged.
- *Tobs*: the observation period prior to declaring node misbehavior.
- *Cmis*: the counter of missing 2ACK packets (maintained at the observing node).
- *Cpkts*: the counter of forwarded data packets (maintained at the observing node).

D. Routing Misbehavior Model

We present the routing misbehavior model considered in this paper in the context of the AODV protocol. Due to AODV's popularity, we use it as the basic routing protocol to illustrate our proposed add-on scheme. The details of AODV can be found in section 3.*A*. The implementation of our scheme as an add-on to other routing schemes will be discussed in Section 6.

We focus on the following routing misbehavior: A selfish node does not perform the packet forwarding function for data packets unrelated to itself. However, it operates normally in the Route Discovery and the Route Maintenance phases of the AODV 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 misbehaving nodes, however, refuse to forward the data packets from the source. This leads to the source being confused. 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 misbehaving nodes and, therefore, the data transmission may fail again. The new Route Discovery phase will return a similar set of routes, including the misbehaving 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 a misbehaving node 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 2ACK technique to detect such misbehaving 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. In this work, we use TCP to demonstrate the adverse effect of routing misbehavior and the performance of our proposed scheme. The attackers misbehaving nodes) are assumed to be capable of performing the following tasks:

- dropping any data packet,
- masquerading as the node that is the receiver of its
- next-hop link,
- sending out fabricated 2ACK packets,
- sending out fabricated key generated by the 2ACK packet senders, and
- claiming falsely that its neighbor or next-hop links are misbehaving.

E. The 2ACK Scheme

The watchdog detection has a very low overhead. Unfortunately, the watchdog technique suffers from several problems such as ambiguous collisions, receiver collisions, and limited transmission power. The main issue is that the event of successful packet reception can only be accurately determined at the receiver of the next-hop link, but the watchdog technique only monitors the transmission from the sender of the next-hop link.

Noting that a misbehaving node can either be the sender or the receiver of the next-hop link, we focus on the problem of detecting misbehaving links instead of misbehaving nodes.

In the next-hop link, a misbehaving sender or a misbehaving receiver has a similar adverse effect on the data packet. It will not be forwarded further. The result is that this link will be tagged in [13]. Our approach discussed here significantly simplifies the detection mechanism.

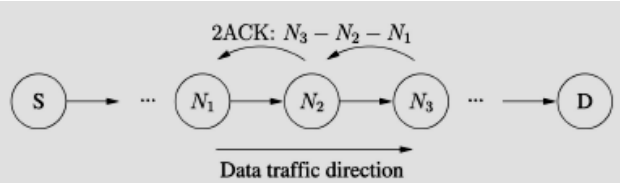

Figure. 1: The 2ACK scheme

F. Details of The 2ACK Scheme

The 2ACK scheme is a network-layer technique to detect misbehaving links and to mitigate their effects. It can be implemented as an add-on to existing routing protocols for MANETs, such as DSR. The 2ACK scheme detects misbehavior through the use of a new type of acknowledgment packet, termed 2ACK. A 2ACK packet is assigned a fixed route of two hops (three nodes) in the opposite direction of the data traffic route.

Figure. 1 illustrates the operation of the 2ACK 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 misbehaving nodes.

N_2 Next Hop Receiver	N_3 Second Hop Receiver	C_{pkts} Packets Transmitted	C_{mis} 2ACK packets Missed	LIST List of data packet IDs

Figure.2 : Data structure maintained by the observing node.

The 2ACK scheme requires an explicit acknowledgment to be sent by N_3 to notify N_1 of its successful reception of a data packet: When node N_3 receives the data packet successfully, it sends out a 2ACK packet over two hops

to N_1 (i.e., the opposite direction of the routing path as shown), with the ID of the corresponding data packet. The triplet $[N_1 \rightarrow N_2 \rightarrow N_3]$ is derived from the route of the original data traffic. Such a triplet is used by N_1 to monitor the link $N_2 \rightarrow N_3$. For convenience of presentation, we term N_1 in the triplet $[N_1 \rightarrow N_2 \rightarrow N_3]$ the 2ACK packet receiver or the observing node and N_3 the 2ACK packet sender.

Such a 2ACK transmission takes place for every set of triplets along the route. Therefore, only the first router from the source will not serve as a 2ACK packet sender. The last router just before the destination and the destination will not serve as 2ACK receivers. To detect misbehavior, the 2ACK packet sender maintains a list of IDs of data packets that have been sent out but have not been acknowledged. For example, after N_1 sends a data packet on a particular path, say, $[N_1 \rightarrow N_2 \rightarrow N_3]$ in Fig. 1, it adds the data ID to LIST (refer to Fig. 2,which illustrates the data structure maintained by the observing node), i.e., on its list corresponding to $N_2 \rightarrow N_3$. A counter of forwarded data packets, *Cpkts*, is incremented simultaneously.

At N_1, each ID will stay on the list for τ seconds, the timeout for 2ACK reception. If a 2ACK packet corresponding to this ID arrives before the timer expires, the ID will be removed from the list. Otherwise, the ID will be removed at the end of its timeout interval and a counter called Cmis will be incremented.

When N_3 receives a data packet, it determines whether it needs to send a 2ACK packet to N_1. In order to reduce the additional routing overhead caused by the 2ACK scheme, only a fraction of the data packets will be acknowledged via 2ACK packets. Such a fraction is termed the acknowledgment ratio, Rack. By varying Rack, we can dynamically tune the overhead of 2ACK packet transmissions.

Node N_1 observes the behavior of link $N_2 \rightarrow N_3$ for a period of time termed Tobs. At the end of the observation period, N_1 calculates the ratio of missing 2ACK packets as $Cmis = Cpkts$ and compares it with a threshold Rmis. If the ratio is greater than R_{mis}, link $N_2 \rightarrow N_3$ is declared misbehaving and N_1 sends out an RERR (or the misbehavior report) packet.

The data structure of RERR is shown in Figure. 3 Since only a fraction of the received data packets are acknowledged, R_{mis} should satisfy $R_{mis} > 1 - R_{ack}$ in order to eliminate false alarms caused by such a partial acknowledgment technique. The optimum value of the threshold R_{mis} should be greater than 1- R_{ack} . In a sense, the difference between R_{mis} and $1 - R_{ack}$ serves as the buffer to avoid false alarms. Each node receiving or overhearing such an RERR marks the link $N_2 \rightarrow N_3$ as misbehaving and adds it to the blacklist of such misbehaving links that it maintains. When a node starts its own data traffic later, it will avoid using such misbehaving links as a part of its route. The 2ACK scheme can be summarized in the pseudo code provided in the appendix for the 2ACK packet sender side (N_3) and the observing node side (N_1).

G. Authenticating The 2ACK Packets

We look into the problem of 2ACK packet fabrication in this subsection. Since the 2ACK packets are forwarded by an intermediate node (e.g., node N_2 in Figure. 1), without proper protection, a misbehaving node *N2* can simply fabricate 2ACK packets and claim that they were sent by node N_3. Therefore, an authentication technique is needed in order to protect 2ACK packets from being forged.

Our scheme uses the one-way hash chain [17], [18], [19] to protect the 2ACK packets against fabrication. Hash chain is used to authenticate 2 ACK packets in such a way that allows every node which receives the message including intermediate node and final destination to verify that the 2 ACK packets has not been modified by any attacker. A hash chain is formed by applying a one way hash function repeatedly to a seed.

To create a one-way hash chain, a node picks up a random initial value $x \in \{1, 0\}^p$ and computes its hash value. The first number in the hash chain h_0 is initialized to x. By using the general formula

$hi = H(h_{i-1})$, for $0 < i \leq n$, for some n, a chain of hi is formed:

$$h_0, h_1, h_2, h_3, \ldots, h_n. \quad (1)$$

It can be proven that, given an existing authenticated element of a one-way hash chain, it is feasible to verify the other elements preceding it. For example, given an authenticated value of hn, a node can authenticate h_{n-3}, by computing $H(H(H(h_n - 3)))$ and comparing the result with h_n [14].

In order to use the one-way hash chain in (1) to authenticate 2ACK packets, node N_3 must distribute the

Option Type	Opt data len	Error Type 2ACK Report Misbehavior	Reserved	Salvage	error source address N_1 (Misbehaving report sender)	Destination S Report receiver	Type-specific information $N_2 -> N_3$ Misbehaving Link

Figure. 3 : Data structure of the RERR packet

h_n element to N_1. A traditional approach for such information distribution is through a trusted certificate authority. However, in a MANET, nodes roam from one place to another and there is usually no central server or base station to act as a trusted certificate entity.

We propose a technique to distribute the initial authentication element h_n from node N_3 to node N_1.

This technique is the "transmission extension" mechanism. Using this technique, N_3 increases the transmission power to send the h_n element directly to *N1*. This technique bypasses N_2, the potential threat to the distribution of h_n. While such a technique consumes more energy from node N_3, it takes place rather infrequently. It will be seen later that every 2ACK packet uses one element in the one-way hash chain in (1). The distribution of a new h_n element is only needed when the entire chain has been used.

N_2 Next Hop Receiver	N_1 Destination	ID sequence number	MAC Signature	h_i hash release

$$MAC = [N_2, N_1, ID]_{h_{i-1}}$$

Figure.4: The packet format of 2ACK

Assume that h_{i-1} has been disclosed (initially = n-1). When node N_3 needs to send a 2ACK packet, it calculates a Message Authentication Code (MAC) based on h_{i-1}, $[N_2, N_1, ID]_{hi-1}$, and attaches the MAC and the *hi* value to the 2ACK packet. Figure. 4 illustrate the packet format of a 2ACK packet.

The fields in Figure. 4 are explained below:

- N_2 : the receiver of the next hop, in the opposite direction of the route.
- N_1 : the destination of the 2ACK packet, the observing node, that is two-hop away from the 2ACKpacket sender.
- *ID*: the sequence number of the corresponding data packet.
- $[N_2, N_1, ID]_{hi-1}$: Message Authentication Code (MAC), signed with h_{i-1}.
- *hi*: the newly disclosed element in the One-way hashchain, 0 < i < n.

Since h_{i+1} is known to N_1, it compares H(hi) with h_{i+1}. If the results match, the h_i element is accepted and recorded. The 2ACK message must have been sent from node N_3. However, the integrity of the 2ACK packet can only be proven when the next 2ACK packet arrives (with h_{i-1}). When h_{i-1} is disclosed to *N1*, it can be used to verify the integrity of the 2ACK packet received last time by calculating the MAC and comparing it with the received one.
This is the so-called “delayed disclosure” technique due to K.Sanzgiri et al.[20].

4. Performance Evaluation

A. Performance Analysis Metrics

The metrics for evaluating the performance of AODV with add on technique 2ACK scheme are as follows for detailed routing protocol analysis.

- Packet delivery ratio – The ratio between the numbers of packets originated by the application layer to those delivered to the final destination.
- Throughput – It is defined as the total useful data received per unit of time.
- Routing overhead – The number of routing packets transmitted per data packet delivered at the destination.
- Path Optimality (Average End-End Delay) - the difference between the number of hops a packet took to reach its destination and the length of the shortest path that physically existed through the network when the packet was originated.

When misbehaving links appear on a route and the acknowledgments from the destination are missing, the source node of a TCP session may slow down or even stop sending packets. Therefore, a more reasonable performance metric is the total number of packets that are received at the destination. We compared a relative packet delivery ratio, end to end delay , a normalized number of packets that are received, of AODV and AODV+2ACK schemes in the TCP traffic scenario.

B. Simulation Environment

Glomosim simulator version 2.03 [21],[22] has been used to analyze the reactive routing protocol AODV+2ACK. The underlying MAC protocol defined by IEEE 802.11 was used with a channel data rate of 11 Mbps. The data packet size was 512 bytes.[23] The wireless transmission range of each node was 250 m. Traffic sources of constant bit rate (CBR) based on TCP have been used. The CBR and TCP mobility scenario of 70 nodes with maximum speed of 20m/sec and for simulation area of 700 x 700 flat area. A random way-point mobility model was assumed with a maximum speed of 0, 10, 20 m/sec and a pause time of 0 second.
Each simulation included 10 CBR sessions, each of which generated four packets per second. Each simulation ran 10 Telnet sessions. The source and the destination nodes were randomly chosen among all nodes in the network. The total simulation time was 800 seconds. For each data point, 20 simulations (with different seeds) were run to obtain the average value.
The snapshot in figure.5 indicates packet transmissions among nodes within the power range. Yellow link indicates links within range, green indicates successful reception and red line is unsuccessful reception. The elements of each of these nodes in the ad-hoc network has a set of protocol layers clearly defined. In each of these layers, important protocol events are generated whose sequences are of particular interest in this work. For the AODV,DSR protocol block, main events of concern are the data packets sent/request, route packet sent/request, broken link error packets received, 2 ACK packet received etc.

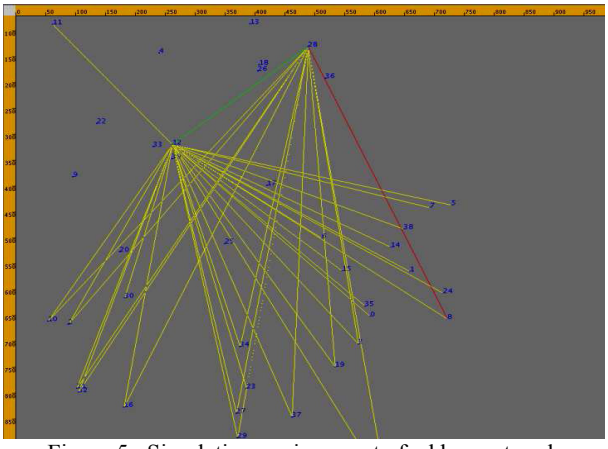

Figure.5 : Simulation environment of ad hoc network

5. Simulation Results

The simulation results of AODV and AODV+2ACK are as follows:
Figure. 6 compares the packet delivery ratio of the AODV+2ACK,and the AODV protocol as a function of misbehavior ratio ρ_m . We varied misbehavior ratio from 0 (all of the nodes are well behaved) to 0.4 (40 percent of the nodes misbehave). The maximum speed is 20 m/sec. From the figure.6, we can observe that most packets were delivered by AODV+2ACK when there is no misbehaving node. The packet delivery ratio decreases as misbehavior

ratio increases. Compared with the original AODV scheme, the AODV+2ACK maintain a much higher PDR. For example, the AODV+2ACK scheme delivered over 90 percent of data packets even when $\rho_m = 0.4$. The rest of the packets were dropped because no well-behaved routes could be found from the source to the destination. On the other hand, AODV delivered about 40 percent of the packets in the same scenario.

Compared with the AODV+2ACK, since the AODV does not detect a misbehaving node/link, it may choose an alternate route which still contains the misbehaving node. Also AODV takes more time to detect misbehaving links, causing more packets being dropped before an alternate route is used.

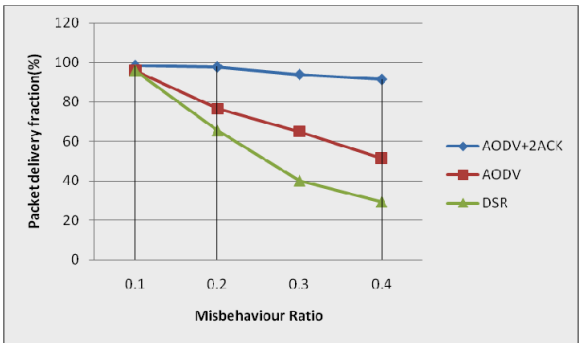

Figure. 6: Packet Delivery Ratio of AODV+2ACK , AODV and DSR

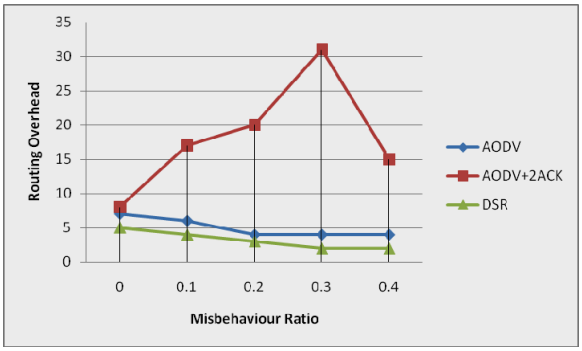

Figure. 7: Routing overhead of AODV+2ACK, AODV and DSR

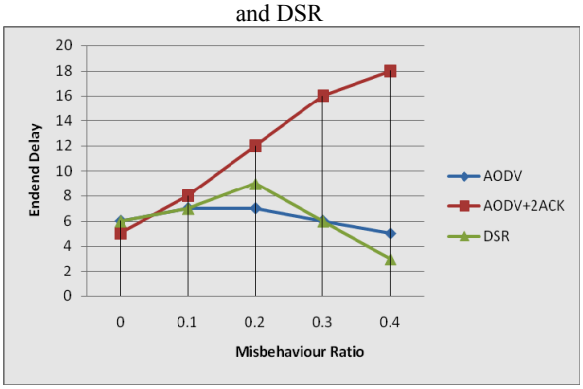

Figure. 8 : Average end to end delay of AODV+2ACK, AODV and DSR

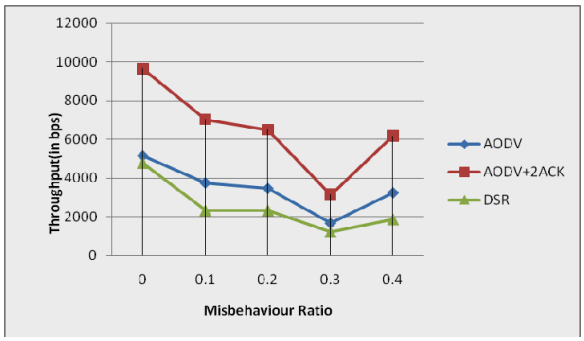

Figure. 9: Throughput of 2ACK+AODV, AODV and DSR

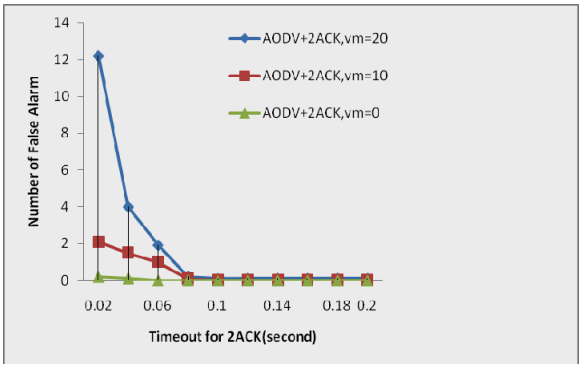

Figure.10: Number of false alarms in 2ACK ($\rho_m = 0$).

In figure 7, we compare the routing overhead of the AODV+2ACK, the AODV, The higher routing overhead in the AODV+2ACK is due to the transmission of extra acknowledgment packets. The extra routing overhead of the AODV scheme is caused by the extra route discovery processes.

The overhead of 2ACK increases with the increase of misbehavior percentage. This is because more RERR (the misbehavior report) and RREQ packets are sent to report misbehaviors and to find alternate routes in a more hostile network environment.

Figure. 8 shows the average end to end delay of AODV+2ACK with AODV. For less number of misbehaving sources DSR have lower delay than AODV and AODV+2ACK.However delay performances worsens with large number of misbehaving sources and gives about twice too much delay than AODV. In Figure. 9, we present the relative throughput, normalized number of packets received, when the AODV+2ACK, AODVand the DSR are used. The relative throughput reduces when ρ_m increases due to higher chances of using routes with misbehaving links and longer time being spent to switch to good routes. Also, we can observe that the AODV+2ACK outperform AODV and the DSR in terms of relative throughput, especially in the networks with larger ρ_m. The relative throughput of the 2ACK scheme is slightly lower than that of the DSR scheme at $\rho_m = 0$. This is due to the false alarm reports in the 2ACK scheme in a high mobility network.

In figure 10, we show the number of false alarms as a function of *timeout* value,τ, for different maximum

speeds Vm. It can be observed that the number of false alarms reduces as timeout increases. The number of false alarms increases when the nodes move more rapidly. This is due to the fact that routes are broken more frequently in a high mobility network and, in some rare cases; the 2ACK scheme may treat such broken routes as misbehaving.

6. Conclusion and Future Work

Mobile Ad Hoc Networks is highly dependent on the cooperation of all of its members to perform networking functions. This makes it highly vulnerable to selfish nodes. One such misbehavior is related to routing. When such misbehaving nodes participate in the Route Discovery phase but refuse to forward the data packets, routing performance may be degraded severely.

In this paper, we have proposed and evaluated a technique, termed 2ACK, to detect and mitigate the effect of such routing misbehavior. The 2ACK technique is based on a simple 2-hop acknowledgment packet that is sent back by the receiver of the next-hop link. Compared with other approaches [24], [25] to combat the problem, such as the overhearing technique, the 2ACK scheme overcomes several problems including ambiguous collisions, receiver collisions, and limited transmission powers. The 2ACK scheme can be used as an add-on technique to routing protocols such as AODV in MANETs.

Our simulation results show that the AODV+ 2ACK scheme maintains up to 91 percent packet delivery ratio even when there are 40 percent misbehaving nodes in the MANETs that we have studied. The regular AODV and DSR scheme can only offer a packet delivery ratio of 50 percent and 30 percent only. The false alarm rate and routing overhead of the 2ACK scheme are investigated as well. One advantage of the 2ACK scheme is its flexibility to control overhead. The robustness of AODV+2ACK has higher performance compared to AODV and DSR in the wide range of operating conditions.

The 2ACK scheme has been implemented on top of AODV. It is also possible to implement the 2ACK scheme over other routing schemes. The main challenge is how to derive the triplet information so that the 2ACK sender and the observing node are informed of such information. Knowledge of topology of the 2-hop neighborhood may be used. In addition, the 2ACK scheme can only work in managed MANETs. In our future work, we will investigate how to add the 2ACK scheme to other types of routing schemes and open networks. Also, the energy extension for both TCP and UDP traffic need to be explored to demonstrate the adverse effect of routing misbehavior and the performance of our proposed scheme.

In MANETs, node mobility often results in frequent topology changes, which presents a significant challenge when designing QoS routing protocols.[26] High node mobility can make satisfying QoS requirements unreachable. Consequently, it is required that the network be *combinatorically stable* in order to achieve QoS support. This means that the changes in network topology must be slow enough within a particular time window to allow the topology updates to propagate successfully as required in the network. QoS support of MANETs requires availability of network state. However, due to mobility and constant topology changes, the cost of maintenance of the network state is expensive especially in large networks. Our future work will also concentrate on the *imprecise network statemodel,* which provides a cost-effective method for providing QoS support based on imprecise network information.

7. References

[1] 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 Orleans, LA, Feb. 1999.

[2] C. E. Perkins, E. M. Royer, and S. Das. RFC 3561: Ad Hoc On Demand Distance Vector (AODV) Routing. http://www.ietf.org/rfc/rfc3561, July 2003.

[3] P. Papadimitratos, Z. J. Haas, and P. Samar. The Secure Routing Protocol (SRP) for Ad Hoc Networks.draft-papadimitratos-secure-routing-protocol-00.txt, Dec. 2002.

[4] L. Buttyan and J.-P. Hubaux, "Security and Cooperation in Wireless Networks," http://secowinet.epfl.ch/, 2006.

[5] R.Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Sustaining Cooperation in Multi-Hop Wireless Networks," Proc. Second Symp.Networked Systems Design and Implementation, Apr. 2005.

[6] V. Srinivasan, P. Nuggehalli, C.F. Chiasserini, and R.R. Rao,"Cooperation in Wireless Ad Hoc Networks," Proc. INFOCOM,Mar.-Apr. 2003.

[7] Sonja Buchegger and Jean-Yves Le Boudec. IBM Research Report: The Selfish Node: Increasing Routing Security in Mobile Ad Hoc Networks. RR 3354, 2001.

[8] S.Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks.In Mobile Computing and Networking, pages 255–265,2000. also available as http://citeseer.nj.nec.com/marti00mitigating.html.

[9] H.Yang, X.Meng, and S.Lu, " Self-organized network layer security in mobile ad hoc networks," in Workshop on Wireless Security (Wise'02), Septemper 2002.

[10] Peng Ning and Kun Sun, "*How to Misuse AODV: A Case study of insider attacksagainst mobile ad-hoc routing protocols*", Proceedings of the 2003 IEEE workshop on Information Assurance.

[11] 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.

[12] 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.

[13] K. Balakrishnan, J. Deng, and P.K. Varshney, "TWOACK:Preventing Selfishness in Mobile Ad Hoc Networks," Proc. IEEEWireless Comm. and Networking Conf. (WCNC '05), Mar. 2005.

[14] Kejun Liu, Jing Deng, , Pramod K. Varshney, and Kashyap Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs," In IEEE Transactions On Mobile Computing, pages 488 – 502,VOL. 6, NO. 5, MAY 2007.

[15] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In Mobile Computing and Networking, pages 85–97, 1998.

[16] P. Papadimitratos and Z. J. Haas. Secure Link State Routing for Mobile Ad Hoc Networks. In IEEE Workshop on Security and Assurance in Ad hoc Networks, in conjunction with the 2003 International Symposium on Applications and the Internet, Orlando, FL, Jan. 2003.

[17] L.Zhou and Z. J. Haas. Securing Ad Hoc Networks.IEEE Network, 13(6):24–30, 1999.

[18] 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. 1, pp. 175-192, 2003.

[19] H. Miranda and L. Rodrigues, "Preventing Selfishness in Open Mobile Ad Hoc Networks," Proc. Seventh CaberNet Radicals Workshop, Oct. 2002.

[20] K. Sanzgiri et al., "A Secure Routing Protocol for Ad Hoc Networks," *Proc. 10th IEEE Int'l Conf. Network Protocols* (ICNP'02), IEEE Press, 2002, pp. 78-87.

[21] The Network Simulator (ns-2)," http://www.isi.edu/nsnam/ns/, 2005.

[22] A Comprehensible GloMoSim Tutorial compilation by Jorge Nuevo, INRS - Universities du Quebec nuevo@inrs-telecom.uquebec.ca March 4, 2004.

[23] S.Anuradha , G.Raghuram, K.E.Sreenivasa murthy, B.Gurunath Reddy, "New Routing Technique to improve Transmission Speed of Data Packets in Point to Point Networks", ICGST-CNIR Journal, Volume 8, Issue 2, January 2009.

[24] N. Asokan and P. Ginzboorg. Key agreement in ad hoc networks. Computer Communications, 23:1627–1637,2000.

[25] V. R. Ghorpade, Y. V. Joshi and R. R. Manthalkar, "Fuzzy Logic based Trust Management Framework for MANET," DSP Journal, Volume 8, Issue 1,December, 2008.

[26] B. Wang and J. C. Hou. Multicast routing and its QoS extension: problems,algorithms, and protocols. *Network, IEEE*, 14(1):22–36, January-February 2000.

T.V.P. Sundararajan received the BE Degree in Electronics and Communication from Kongu Engineering College , Perundurai in 1993 and the ME Degree in Applied Electronics from the Government college of technology, coimbatore in 1999.
He is Assistant Professor ,working in Bannari Amman Institute of Technology, Sathyamangalam. He is doing a part time doctoral research in Anna University, Chennai . His current research focuses on mobile ad hoc networks and wireless security. He is member of the IEEE, ISTE and the IEEE computer society.
E-mail : tvpszen@yahoo.co.in
URL : http://www.bitsathy.ac.in

Dr.A.Shanmugam received the BE Degree in PSG College of Technology in 1972, Coimbatore and ME Degree from College of Engineering, Guindy, Chennai in 1978 and Doctor of Philosophy in Electrical Engineering from Bharathiar University, Coimbatore in 1994.From 1972–76, he worked as Testing Engineer in Testing and Development Centre, Chennai. He was working as a Lecturer Annamalai University in 1978. He was the Professor and Head of Electronics and Communication Engineering Department at PSG College of Technology, Coimbatore during 1999 to2004. Authored a book titled "Computer Communication Networks" which is published by ISTE, New Delhi, 2000.He is currently the Principal, Bannari Amman Institute of Technology, Sathyamangalam..He is on the editorial board of International Journal Artificial Intelligence in Engineering & Technology (ICAIET), University of Malaysia, International Journal on "Systemics, Cybernetics and Informatics (IJSCI)" Pentagram Research Centre, Hyderabad, India. He is member of the IEEE, the IEEE computer society.
E-mail : principal@bitsathy.ac.in