A Framework for Routing Misbehavior Recognition in MANETS

Dr K Butchi Raju

Professor, Department of CSE, GRIET, Hyderabad, India

Abstract—Mobile Ad-hoc Networks (MANETs) operate on the basic underlying assumption that all participating nodes fully collaborate in self-organizing functions. However, performing network functions consumes energy and other resources. Therefore, some network nodes may decide against cooperating with others. These nodes are called selfish / misbehaving nodes. Misbehavior of suspicious nodes in MANETs is detected, and the information is propagated throughout the network, so that the misbehaving node will be cut off from the rest of the network. We propose a network-layer acknowledgment-based scheme, termed the 2ACK scheme, which can be simply added-on to any source routing protocol. The proposed scheme detects misbehaving nodes, and then seeks to alleviate the problem by notifying the routing protocol to avoid them in future routes. Our results show that proposed scheme reasonably improves the packet delivery ratio, with some additional routing overhead.

Keywords—MANET, selfish routing, misbehaving nodes, 2ACK.

I. INTRODUCTION

Mobile Ad Hoc Network (MANET) can be described as an autonomous collection of mobile nodes (users) that communicate over relatively low capacity wireless links, without a centralized infrastructure[1,2]. In these networks, nodal mobility and the wireless communication links may lead to dynamically changing and highly unpredictable topologies. All network functions such as routing, multi-hop packet delivery and mobility management have to be performed by the member nodes themselves, either individually or collectively[3,4]. So, network performance becomes highly dependent on collaboration of all member nodes. We can found Manet’s applications in various fields ranging from low-power military wireless sensor networks to large-scale civilian applications, and emergency search/rescue operations [5].

Characteristics of MANETs:
A MANET is an autonomous system of mobile nodes with routing capabilities connected by wireless links, the union of which forms a communication network[6]. A MANET can either be a standalone entity or it can be an extension of a wired network. There are many application areas of MANETs, such as:

- Military tactical operations - for fast and possibly short term establishment of military communications for troop deployments in hostile and/or unknown environments.
- Search and rescue missions - for communication in areas with little or no wireless infrastructure support.
- Disaster relief operations - for communication in environments where the existing infrastructure is destroyed or left inoperable.
- Law enforcement - for secure and fast communication during law enforcement operations.
- Commercial use - for enabling communications in exhibitions, conferences, and large gatherings.

Applications of MANETs:
- Industrial and commercial applications involving cooperative mobile data exchange.
- Mesh-based mobile networks can be operated as robust, inexpensive alternatives or enhancements to cell-based mobile network infrastructures.
- There are also existing and future military networking requirements for robust, IP-compliant data services within mobile wireless communication networks many of these networks consist of highly-dynamic autonomous topology segments.
- The Developing technologies of "wearable" computing and communications may provide applications for MANET technology.
- When properly combined with satellite-based information delivery, MANET technology can provide an extremely flexible method for establishing communications for fire/safety/rescue operations or other scenarios requiring rapidly-deployable communications with survivable, efficient dynamic networking.

Types of MANETs:
There are two types of MANETs:
- Open
- Closed

An open MANET comprises of different users, having different goals, sharing their resources to achieve global connectivity, as in civilian applications [1-4].
In closed MANET, the nodes are all controlled by a common authority, have the same goals, and work toward the benefit of the group as a whole.

Misbehaving nodes in MANETs:
Open environment of a MANET may lead to misbehaving nodes. Misbehaving nodes come into existence in a network due to several reasons[1]:

- Mobile hosts lack adequate physical protection (due to the open communication medium), making them prone to be captured and compromised.
- Usually mobile hosts are resource constrained computing devices. Performing network functions consumes significant energy of participating nodes, as communication is relatively costly. Selfish nodes are unwilling to spend their precious resources for operations that do not directly benefit them. MANETs lack a centralized monitoring and management point, making it a challenging task to detect such misbehaving nodes effectively.

Types of Selfish nodes
Three types of selfish nodes related to routing such as Dynamic Source Routing (DSR) are defined.

- Selfish Nodes Type 1 (SN1) – These nodes participate in the DSR Route Discovery and Route Maintenance phases, but refuse to forward data packets.
- Selfish Nodes Type 2 (SN2) – These nodes participate in neither the Route Discovery phase, nor forwarding data packets. They only use their energy for transmissions of their own packets.
- Selfish Nodes Type 3 (SN3) – These nodes behave (or misbehave) differently based on their energy levels.

The main focus of this is only on the detection and mitigation of SN1 type misbehavior. The 2ACK scheme is used 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” acknowledgment 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.

A node acknowledges the receipt of a data packet by sending back a two-hop 2ACK packet along the active source route. If the sender/forwarder of a data packet does not receive a 2ACK packet corresponding to a particular data packet that was sent out, the next-hop’s forwarding link is claimed to be misbehaving and the forwarding route broken. Based on this claim, the routing protocol avoids the accused link in all future routes, resulting in an improved overall throughput performance for the network.

The 2ACK can be implemented as a simple add-on to any source routing protocol such as DSR.

II. LITERATURE SURVEY

a. Schemes to Prevent Selfishness in MANETs
Various techniques have been proposed to prevent selfishness in MANETs. These schemes can be broadly classified [7-12] into two types.

- Reputation-based schemes
- Credit-based schemes

In [1], a reputation-based approach, nodes (either individually or collectively) detect, and then declare another node to be misbehaving. This declaration is then propagated throughout the network, leading to the misbehaving node being avoided in all future routes. A credit-based approach, [11] on the other hand, uses the concept of virtual currency. Nodes pay virtual money for services (networking resources) that they get from other nodes, and similarly, get paid for providing services to other nodes [6].

- The 2ACK scheme is reputation-based.

b. Dynamic Source Routing (DSR)
DSR is an on-demand, source routing protocol. Every packet has a route path consisting of the addresses of nodes that have agreed to participate in routing the packet. The protocol is referred to as on-demand because route paths are discovered at the time a source sends a packet to a destination for which the source has no path [13].

We divide DSR into two main functions:
- Route Discovery
- Route Maintenance.

Node S (the source) wishes to communicate with node D (the destination) but does not know any paths to D. S initiates a route discovery by broadcasting a route request packet to its neighbors that contains the destination address D. The neighbors in turn append their own addresses to the route request packet and rebroadcast it. This process continues until a route request packet reaches D. D must now send back a route reply packet to inform S of the discovered route. Since the route request packet that reaches D contains a path from S to D, D may choose to use the reverse path to send back the reply (bidirectional links are required here) or to initiate a new route discovery back to S. This is shown in Figure 1. Since there can be many routes from a source to a destination, a source may receive multiple route replies from a destination. DSR caches these routes in a route cache for future use.
The second main function in DSR is route maintenance, which handles link breaks. A link break occurs when two nodes on a path are no longer in transmission range. If an intermediate node detects a link break when forwarding a packet to the next node in the route path, it sends back a message to the source notifying it of that link break. The source must try another path or do a route discovery if it does not have another path.

Two modules called watchdog and pathrater are implemented at each node, to detect and mitigate, respectively, routing misbehaviors in MANETs. Nodes operate in a promiscuous mode wherein, 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 in the buffer too long, 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.

**c. Watchdog**

The watchdog method detects misbehaving nodes. Figure 2 illustrates how the watchdog works. Suppose there exists a path from node S to D through intermediate nodes A, B, and C. Node A cannot transmit all the way to node C, but it can listen in on node B’s traffic. Thus, when A transmits a packet for B to forward to C, A can often tell if B transmits the packet. If encryption is not performed separately for each link, which can be expensive, then A can also tell if B has tampered with the payload or the header.

---

**Fig. 1. DSR route discovery.**

**Fig. 2: Watchdog**
Fig 2 When B forwards a packet from S toward D through C, A can overhear B’s transmission and can verify that B has attempted to pass the packet to C. The solid line represents the intended direction of the packet sent by B to C, while the dashed line indicates that a is within transmission range of B and can overhear the packet transfer.

We implement the watchdog by maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node.

The watchdog technique has advantages and weaknesses. DSR with the watchdog has the advantage that it can detect misbehavior at the forwarding level and not just the link level. Watchdog’s weaknesses are that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping.

d. Path rater

The path rater, run by each node in the network, combines knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable. Each node maintains a rating for every other node it knows about in the network. It calculates a path metric by averaging the node ratings in the path. We choose this metric because it gives a comparison of the overall reliability of different paths and allows path rater to emulate the shortest length path algorithm when no reliability information has been collected, as explained below. If there are multiple paths to the same destination, we choose the path with the highest metric. Note that this differs from standard DSR, which chooses the shortest path in the route cache.

Further note that since the path rater depends on knowing the exact path a packet has traversed, it must be implemented on top of a source routing protocol.

III. PROBLEM STATEMENT

In order to demonstrate the adverse effect of routing misbehavior, we estimate the probability of misbehaving routes first. A route is defined as misbehaving when there is at least one router along the route that can be classified as misbehaving. Our analysis is based on the following assumptions:

- The network nodes are randomly distributed over the entire network area. Each node’s location is independent of all other nodes' locations. There are N nodes in the network area of size X *Y
- The source and the destination of each transaction are chosen randomly among all nodes.
- Nodes (other than the source and the destination) are chosen as misbehaving nodes, independently, with probability p_m.

A selfish node does not perform the packet forwarding function for data packets unrelated to it. However, it operates normally in the Route Discovery and the Route Maintenance phases of the DSR protocol, they may be included in the routes chosen to forward the data packets from the source.

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.

The 2ACK Scheme has been implemented on top of DSR. It is also possible to implement the 2ACK scheme over other routing schemes. The main challenge is how the ACK sender and observing node are informed such information.

IV. MODULES

The modules that are included in this project are

- Route Request for Identifying the misbehavior Node
- Message Transfer to the available path
- Route Maintenance
- Cache Updating for misbehavior node identification

a. Route Request for identifying the misbehavior Node

When a source node wants to send packets to a destination to which it does not have a route, it initiates a Route Discovery by broadcasting a ROUTE REQUEST. The node receiving a ROUTE REQUEST checks whether it has a route to the destination in its cache and also check if it is misbehavior node or not. If it has, it sends a ROUTE REPLY to the source including a source route, which is the concatenation of the source route in the ROUTE REQUEST and the cached route. If the node does not have a cached route to the destination, it adds its address to the source route and rebroadcasts the ROUTE REQUEST. When the destination receives the ROUTE REQUEST, it sends a ROUTE REPLY containing the source route to the source. Each node forwarding a ROUTE REPLY stores the route starting from itself to the...
destination. When the source receives the ROUTE REPLY, it caches the source route. If any node not sends acknowledgement then we easily identified that is misbehavior node. So find out the alternative path and forwarding the data to the destination.

b. **Message Transfer to the available path**
The Message transfer relates with that the sender node wants to send a message to the destination node after the path is selected also find out that node is not a misbehavior node and status of the destination node through is true. The receiver node receives the message completely and then it send the acknowledgement to the sender node also nearby nodes through the router nodes where it is received the message.

c. **Route Maintenance**
Route Maintenance, the node forwarding a packet is responsible for confirming that the packet has been successfully received by the next hop. If no acknowledgement is received after the maximum number of retransmissions, the forwarding node sends a ROUTE ERROR to the source (also source easily understand that node is misbehavior), indicating the broken link. Each node forwarding the ROUTE ERROR removes from its cache the routes containing the broken link.

d. **Cache Updating for misbehavior node identification**
When a node detects a link failure, our goal is to notify all reachable nodes that have cached that link to update their caches if it is not a misbehavior node. To achieve this goal, the node detecting a link failure, identifying the misbehavior node needs to know which nodes have cached the broken link and needs to notify such nodes efficiently. Our solution is to keep track of identifying the misbehavior node in a distributed manner.

V. **RESULTS**
In this paper we implemented a misbehaving mechanism using JAVA, which is a platform independent, robust and user friendly language. Screens used in this implementation are discussed in next sections.

Fig 3 shows the All Nodes in one Screen, which consists of node1,node2,node3, node4(here we consider four nodes as like different clients)

![Fig 3: All nodes in one screen](image1)

Fig 4 Shows Sending Message from one Node to another Node without backend intervention.
Fig. 4: Sending message in between nodes

Fig 5 shows the received Message of Node2 & Sending Message shown by Node4

Fig 5: Shows message receiving

Fig 6 shows the Message sending from Node3 to Node2 via the PATH ‘A&C&B’. In this path A behaves like Misbehave Node then it fine another path, the other path contains misbehave node then display this message
Fig. 6: Message passing (with misbehavior)

Fig 7 shows the Message sending from Node4 to Node2 via the PATH ‘A&C&B’. In the Figure 6 A is Misbehave then change the node status to Node Active then gives the message.

Fig. 7: Message passing (without misbehavior)

VI. CONCLUSION
In this paper, we investigated the performance degradation caused by selfish (misbehaving) nodes in MANETs. The proposed 2ACK technique detects and mitigates the effect of routing misbehaviors. This 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 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 DSR in MANETs. One advantage of 2ACK scheme is its Flexibility. In this, we focus only on the node misbehavior. In order to decide the behavior of a node and punish it, we may need to check the behavior of links around that node. This is potential direction for our future work. Our future work we will investigate how to add the 2ACK scheme to other types of routing schemes.

REFERENCES


