Context-free Protocol and N-ACK scheme for Secure Routing in MANET

Snehal Jadhav¹, Prof. Sonali Patil²

Computer Department, Alard College of Engineering, Pune, India
¹ jadhavsneha_9@yahoo.co.in, ² sonali.patil@live.com

Abstract—Routing protocols for MANETs are built on the basis of the presumptions that all including hubs are completely cooperative. Then again, because of the open structure and limited available battery-based energy, node rogueish activities may exist. One such routing misbehavior is that some selfish nodes will take an interest in the route discovery and maintenance processes however decline to forward data packets. A danger to such multi-hop transmission is posted by selfish nodes, which may drop others packets to spare their own bandwidth and battery life. So, packet forwarding is an important issue for wireless ad hoc networks. Solutions proposed so far are either reputation based or credit-exchange based, which are in far-reaching way context-based solutions. They have to precisely distinguish selfish behaviors, securely keep up the context, and suitably punish selfish nodes. These pre-requirements are hard to fulfill. This proposed technique develops a context-free protocol that does not depend on observation and selfish behavior identification. Given a way, a context-free protocol can transmit packets through it without knowing whether the mediator nodes are selfish or not. In this planning, the data of a packet ought to be encrypted and the identity of the destination ought to just be uncovered after all nodes sent the packet agreeably. Additionally, N-ACK scheme is presented to overcome this issue of misbehaving nodes. In NACK scheme, ACK packet is sent between source and destination. On receiving the packet, destination sends an N-ACK packet to source.

Keywords—Mobile Ad Hoc Network Selfish nodes, context-free protocol, N-ACK

1. INTRODUCTION

A mobile Ad-Hoc network is a self-organising infrastructure less network of mobile devices connected with wireless links. Every device is allowed to freely move in MANET in any direction and will change its connection frequently. The main problem in building a MANET is preparing every device to maintain the data to appropriately route traffic. Such networks can work independently or can be connected to big networks. They may contain one or more transceivers between the nodes. This result is highly dynamic. Consequently, the network topology of MANET may change quickly or un-predictably, without main router, when the node is willing to send the router to remote destination it depends on other nodes to forward the packet.

On the other hand, devices in an ad hoc network may have relate to distinctive authorities for example diverse person or organization, wireless node may act selfish and are not willing to spend their own bandwidth and battery power to transmit packet to different users. An individual mobile node may try to profit from different nodes yet refuses to share its own resources such nodes are called as narrow selfish or misbehaving nodes and their result is termed as selfishness or misbehavior. This is important issue in ad-hoc network and number of solution have been proposed to animate node cooperation.

Selfish nodes:-
As there is no dedicated infrastructure, the node needs to co-operate and self-organise to build a working communication network. Node takes parts and forward other nodes' packets. Then every node needs to consider the constrained resources. Such as its energy. So every node is persuaded to help as little as could reasonably be expected of its own.

Usually it is expected that all node forwarded as required, but still different policies are conceivable also. In any manner the MANET’s protocols and policies imply a regulating expectation on each participating node: a) to act as indicated by agreed protocols and b) to forward a huge amount of other node’s packets as required. As long as all nodes hold to this and cooperate, the MANET ought to work without issues. A most important among the vital issues in designing MANET protocols is the way to manage nodes that do not cooperate. Selfish nodes are those nodes that do not cooperate to forward other’s packets and thus expanding their profit to the expense of all others. They are expected to always behave rationally, so they cheat when it gives advantage to them.

Watchdog finds whether the following node is able to forward the packet in the route. If the following node fails to forward the packets, then it is known as misbehavior. The benefits of Watchdog mechanism is that it can find misbehaving nodes not in sending level but also in the level of connection. As it were, it finds nodes not just in the link layer, additionally in the network layer. Development of Watchdog is generally simple. In Kachirski O et. al. [2], Watchdog has a some disadvantages. Such as, due to the resistance in nodes, it might be not able to find misbehaving nodes in circumstances, for example, 1. Receiver collision 2. Ambiguous collision 3. Restricted transmission power 4. Minor dropping 5. Collision 6. False misbehaving.

In Buchegger et al [3], presented a CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad hoc Networks), which is an addition of DSR protocol. This procedure is like Watchdog and Pathrater. Every node monitors the behavior of neighbor nodes in its radio range and learns from them. This protocol solves the Watchdog and Pathrater issue, involving that it does punish misbehaving nodes by not utilizing them as a part of routing and not sending parcels through them. Moreover, when a node finds a misbehaving node, it sends an alarm to all other node and they do not utilize this not either. CONFIDANT protocol comprises of Monitoring System, Reputation System, Trust Manager and Path Manager. Their assignments are divided into two segments: the tactics to handle their own particular observations and the one to handle reports from trustful nodes. For observations, the observing node utilizes a “neighborhood watch” in its radio range to find any malicious behavior. If a suspicious event finds, then observing node then reports it to the disrepute framework. Around then, the reputation framework performs a number of checks and upgrades the rating of the reported node in the reputation table. If the rating result is doubtful, it advances the data to the path.

2. RELATED WORK

As there is no committed infrastructure in ad-hoc network all nodes ought needs to co-ordinate with one another in routing and forward packet however actually they delete or change them, as they are nasty. And entire framework performance can be reduced by only few of misbehaving nodes. A few techniques and protocols have been proposed to recognize and avoid such misbehaving nodes by Farzaneh et al. [1].

A. Reputation Based technique

Reputation based technique mainly depend on building a reputation metric for every node as per its behavioral pattern. Following are some examples of reputation based techniques.

In Kachirski O et. al. [2], the method differentiated misbehaving node by eavesdropping on the transmission of the following hop. At the point when a node forwards packets,
manager, which then excludes all paths containing the misbehavior node.

At that point the trust administrator sends An ALARM to alert different nodes that consider these nodes as companions. [3]. At the point when the observing hub gets an ALARM message from trusted nodes, right away the trust manager assesses the message to find whether the source node is trustworthy. If so, the ALARM message with the trust level will be added in the alarm table [3]. In Michiardi et. al [4], the method finds selfish nodes and make them to coordinate too. Like CONFIDENT, This strategy is focused on monitoring system and reputation system, which includes both direct and indirect reputation from the framework. Sometimes nodes do not misbehave purposefully; for instance when their battery is low, they should not be viewed as misbehaving nodes and remove from the system. To do in this way, the reputation ought to be rated based on past reputation, which is zero at the starting. Also, participation in the network can be categorized into a number of functions, for example, routing discovery in DSR or sending packets. The difference in CORE and CONFIDANT is that CORE only permits positive reports to pass through however CONFIDANT permits the negative ones. This implies that CORE counter acts false reports, and hence it keeps a DOS attack which CONFIDANT cannot do. When a node cannot coordinate, it is given a negative rating and its reputation reduces. moreover a positive rating is given to a node from which a positive report is received and afterward its reputation increments.

In Bansal et al. [8], likewise proposed a protocol called OCEAN (Observation-based Cooperation Enforcement in Adhoc Networks) which is an extension of the DSR protocol. OCEAN like past methods utilizes a monitoring and a reputation framework. Then again, in spite of past tactics OCEAN depends only on its observation to keep away from the new vulnerability of false allegation from second-hand reputation exchanges. Thus, OCEAN can be viewed as separate architecture.

OCEAN grouped routing misbehavior into two types: misleading and selfish. If a node takes part in the route discovery But still a packet, its class misleading as it misleads different modes to route packets through it. In any case if a node does not even take part in the route discovery, it is considered as selfish. So as to find the misleading routing behaviors, a node buffers the packet check-sum after sending a packet to a neighbor, then it can monitor if the neighbor actions to forward the packet in a given time. As a consequence of monitoring, either a negative or positive event is created to upgrade the neighbor rating. In the event that the rating is lower than the faulty threshold, that neighbor node is added to a faulty list and afterward to the RREQ as an avoid-list. Also, all the traffic from the misbehaving neighbor node will be dismissed.

B. Credit Based Technique: -

The main idea credit based technique is to give encouragement to the node to loyally perform networking objectives. To get done this objective virtual currency or similar payment framework may be set. for giving services to other nodes nodes are paid. When they ask other nodes for help them for sending packets. They uses same payment framework to pay for such services.

There are two different models: the Packet Purse Model and the Packet Trade Model [6]. In the Packet Purse Model, nuggets are heaped into the parcel before it is sent. The sender puts a particular number of nuggets on the data packet to be sent. Each intermediate node acquires nuggets in exchange for sending the packets. If the packet exhausts its chunks before arriving at its Destination, then it is dropped. In the Packet Trade Model, each intermediate node “purchases” the packet from the past node for a few nuggets and “offers” it to the other node for more nuggets. Therefore, each one moderate hub gains a few nuggets for giving the forwarding service and the cost of sending the packet is borne by the destination. The primary issue with credit-based schemes is that they generally oblige tamper resistant equipment and/or
supplementary security for the virtual money or the payment system.

3. PROPOSED SYSTEM

A. System Architecture

![System Architecture Diagram]

Fig.1: System Architecture

The main problem in mobile ad hoc network is delivering the packet to destination because of the selfish behavior of the middle node to solve this issue a scheme is proposed for detecting selfish nodes and keeping them out of routine path, context free protocol and N_ACK are presented. In context free protocol entire packet is encrypted with K keys and key k is encrypted and other two parts are hashed results Nothing can be utilized to guess the identity of the destination. Thus, the nodes cannot get the identity of the destination from the packet itself. In Context Free Protocol, there are three steps which are shown in fig.1 and expressed as follows:

1) Context Free Path:: Suppose node A is the source node, D is the destination, B, C, and N are other nodes in the Network, and the routing path is A-B-C-D. In context free path, destination node D is also an intermediate node, so source node A gets neighbor node N of D and routing path is changed to A-B-C-N-D. Thus packet will arrive at the destination node twice.

2) Encryption :: Source node A encrypts data packet using random key K. After that, key is encrypted by using public key of all intermediate nodes in route path in reverse order.

3) Packet Forwarding:: Packet is broadcasted to all the nodes in the network. Then nodes will decrypt the cipher path and compare with hash key to see whether it is destination or not. If it is destination, K is decrypted out and packet body will be open, or it will forward to other nodes. In the N-ACK scheme, an end to end ACK packet is necessary to be sent between the source and the destination. After receiving packet from the source node, the destination sends back the ACK packet. Every node maintains a list of data packets sent and other list of data packets forwarded. When node sends data packet as a source, it includes the ID of the packet to the list of data packet sent. As the node gets the N-ACK packet for the data packet it removes the comparing data packet ID from the data packet sent list. The N-Ack packet format is shown in fig.2 below

![N-Ack Packet Format]

Fig.2: N-Ack Packet Format

B. Algorithm

Algorithm 1
1: Node1 the source wants to send a packet to Node5 the destination and the route is Node2!Node3!Node4.
2: Node1 put the id of the packet to a wait list.
3: Node1 sends the packet to Node2 and waits for ACK.
4: If there is no ACK within the stipulated time Node1 again sends for K times after that it announces that Node2 to be misbehaving
5: Then Node1 waits for the N ACK packet from the destination.
6: It sets up a timer.
7: Each intermediate node maintains a list of IDs for a data packet sent on a path
8: Each packet ID will stay for a time T
9: If ACK arrives within T, the ID is removed
10: Else ID will be removed after the timeout
11: Node 5 has to send back the N ACK packet to the source.
12: Each intermediate node has to forward the N ACK packet to the source in the same path in which the initial transmission took place.
13: Each intermediate node also has to send to its immediate source node an ACK packet.
14: Each node maintains a black list of potential misbehaving nodes.
15: If the ACK is not received by a particular node then the node to which it has forwarded the packet and has failed to receive the ACK is added to the list.
16: After m failed attempts to send the packet without receiving the ACK misbehaving node.

C. Mathematical Model

1) Set Theory: \( N_w = \) is the set of all network nodes
\( N_w = \{N_1, N_2, N_3, \ldots, N_n\}\) \( N_i = \) ith Node in the network
\( S_N = \) Source Node
\( D_N = \) Destination Node
\( W_l = \) is the Waiting List
\( N_w = \{NID_1, NID_2, NID_3, \ldots, NID_n\}\)
\( NID_i = \) ID of ith Node in the network
\( P = \{p_1, p_2, p_3, \ldots, p_n\}\)
P is set of sending packets.
\( T_p = \) packet sending Time
\( T_{rp} = \) Packet Receiving time
\( B_L = \{NID_1, NID_2, NID_3, \ldots, NID_n\}\)

\( B_L = \) Blacklist containing misbehaving nodes
\( NID = \) Misbehaving node ID.

2) Packet delivery Ratio calculation:

The ratio of number of packets received to the destination number of packet sent by source

\[
PDR = \sum_{i=0}^{N} \frac{P_{received}}{P_{sent}}
\]

Where,
\( P_{received} = \) Number of packet received.
\( P_{sent} = \) Number of packet sent.

3) NP- analysis:- In this system we predicts the content free algorithm with N-ACK scheme against selfish intermediate nodes and it is NP hard algorithm. We predict the content free algorithm with N-ACK scheme within a polynomial type. Here sender sends packets to intermediate node and sender node wait for acknowledgement, if it does not arrive in specified time then sender predicts that it is selfish node. This is NP hard because we predict that all the process should be completed within polynomial time and this algorithm detect selfish node in network accurately. This algorithm will be NP complete after performing implementation and performing testing on it.

4. EXPERIMENTAL RESULTS

The system is built using Java (Version JDK 8) to evaluate the efficiency, effectiveness. The development tool used is Net-Beans (Version 8). The experiments performed on Core2Duo Intel processor, 2GB RAM under Windows XP Professional. The system does not require any specific hardware to run; any standard machine is capable of running the application.

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Existing System</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>40</td>
<td>7</td>
<td>24</td>
</tr>
<tr>
<td>50</td>
<td>9</td>
<td>30</td>
</tr>
</tbody>
</table>

TABLE I. PACKET DELIVERY RATIO.

Table 1 gives no. of packets loss between existing system and proposed system. It shows estimated
analysis of present system and proposed system with respect to total number of packets sent and average packet loss in network. Packet loss is done by selfish nodes to save their own energy, so packet delivery ratio is reduced. In existing system packet are sent with specific destination address whereas in proposed system packets are sent to next hop and waits for N-Ack, if N-Ack is not received in time from next hop neighbor node then node is said to be selfish node and it will send the data through next hop node. Due to this the packet delivery is increased in the network. Fig. 3 shows the graph based on values in Table 1. It appears that the average packet loss for proposed algorithm is less than present system.

Fig. 3: No Of Packet Vs packets Loss

5. CONCLUSION

In last few years mobile ad-hoc network has been area of active search though it is having main drawback of packet forwarding. Because of the selfish and misbehaving nodes packets do not get delivered to destination to solve this issue two schemes are proposed context free protocol and N-ACK scheme. In context free protocol packets are encrypted and forwarded to the network as node is encrypted no node can infer what is the destination of packet and in N-ACK scheme end to end packet is necessary to be sent between source and destination, after receiving packet from the source node the destination sent back the ACK packet by this packet delivery is ensured.

ACKNOWLEDGMENT

The authors would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. We also thank the college authority for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

REFERENCES