ISSN:0975-9646

Route Management Protocol for Misbehavior in Ad Hoc Networks

A. Shalini
JNTU Hyderabad

Krishna Rao
Sri Datta Engg & Tech, Hyderabad

Sridhar G
Dept of Maths, S.V.K.P. & Dr.K.S.Raju A&S College,, Penugonda

Abstract-Ad hoc wireless networks have emerged as one of the key growth areas for wireless networking and computing technology. Ad hoc networks are a new wireless networking paradigm for mobile hosts. Unlike traditional mobile wireless networks, ad hoc networks do not rely on any fixed infrastructure. Instead, hosts rely on each other to keep the network connected. Most of the routing protocols in wireless ad hoc networks, such as DSR, assume nodes are trustworthy and cooperative. One of the major factors effecting the ad hoc communication is the misbehaving of nodes. Although an efficient power management scheme is applied to an ad hoc network, a misbehaving node may result in the improper routing of packet which may extend to the complete collapsing of the network also. Existing approaches such as economic incentives or secure routing by cryptographic means alleviate the problem to some extend with limitations. The main objective of this project is to cope with misbehavior. This paper task, address the question of how to enable a system to operate despite the presence of misbehavior in a mobile ad-hoc network. How the network can be functional for normal nodes when other nodes do not route and forward the packet correctly. In this paper an optimal routing protocol for misbehaving network called RMP (Route Management Protocol) to cope with misbehavior for ad hoc network is proposed. The protocol enables nodes to detect misbehavior by first-hand observation and use of second-hand information provided by other nodes. A fully distributed reputation system that can cope with false information and effectively use second information in a safe way is proposed. The approach uses a modified Bayesian estimation and classification procedure for the isolation of malicious and selfish node in a given network.

***Key words*: Misbehavior; Route management protocol(RMP); Ad hoc networks; Bayesian analysis**

1. INTRODUCTION

Wireless networking grows rapidly because of the human desires for mobility and for freedom from limitation, i.e., from physical connections to communication networks. Recent advances in wireless technology have equipped portable computers, such as notebook computers and personal digital assistants with wireless interfaces that allow networked communication even while a user is mobile. A particular kind of wireless network called mobile ad hoc networks is presently under development. A mobile ad hoc network is a self-organizing and rapidly deployable network in which neither a wired backbone nor a centralized control exists. The network nodes communicate with one another over scarce wireless channels in a multi-hop fashion. The ad hoc network is adaptable to the highly dynamic topology resulted from the mobility of network nodes and the changing propagation conditions. These networks are used in emergency disaster rescue operation, tactical military communication and law enforcement. In these applications, where a fixed backbone is not available, a readily deployable wireless network is needed. Mobile ad hoc networks are also a good alternative in rural areas or third world countries where basic communication infrastructure is not well established The limited resources in MANETs have made designing of an efficient and reliable routing strategy a very challenging problem. An intelligent routing strategy is required to efficiently use the limited resources while at the same time being adaptable to the changing network conditions such as: network size, traffic density and network partitioning. In parallel with this, the routing protocol may need to provide different levels of QoS to different types of applications and users.

Butty_an and Hubaux proposed incentives to cooperate by means of so-called nuglets [3] that serve as a per-hop payment in every packet or counters [4] in a secure module in each node to encourage forwarding. One of their findings is that increased cooperation is beneficial not only for the entire network but also for individual nodes, which conforms to our results. The main differences to the RMP protocol are that nuglets or counters are limited to a one-to-one interaction, whereas in the RMP protocol, misbehavior results in a bad reputation propagating to more than one node. Marti, Giuli, Lai, and Baker [5] observed that throughput increased in mobile ad-hoc networks by complementing DSR with a `watchdog' for detection of non-forwarding nodes and a `pathrater' (for trust management and routing policy, every path used is rated), which enable nodes to avoid non-forwarding nodes in their routes. Ratings are kept about every node in the network and the rating of actively used nodes is updated periodically. Their approach does not punish malicious nodes that do not cooperate, but rather relieves them of the burden of forwarding for others, whereas their messages are forwarded without complaint. A collaborative reputation mechanism proposed by Michiardi and Molva [6], also has a watchdog component, however it is complemented by a reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior), which are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. Regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request obtain reputation values. Nodes only exchange positive reputation information, thus making the same trade-off between robustness against lies and detection speed as the watchdog and path rater scheme, but in addition, false praise can make malicious nodes harder to detect.

A formal model for trust in dynamic networks based on intervals and a policy language has been proposed by Carbone, Nielsen, and Sassone [8]. They express both trust and the uncertainty of it as trust ordering and information ordering, respectively. They consider the delegation of trust to other principals. In their model, only positive information influences trust, such that the information ordering and the trust ordering can differ. In our system, both positive and negative information influence the trust and the certainty. One node can have varying reputation records with other nodes across the network, and the subjective view of each node determines its actions. Byzantine robustness [10] in the sense of being able to tolerate a number of erratically behaving servers or in this case nodes is the goal of a reputation system in mobile ad-hoc networks. Here, the detection of malicious nodes by means of the reputation systems has to be followed by a response in order to render these nodes harmless.

2. MODIFIED BAYESIAN METHOD

2.1. *Gathering First-hand Information*

Node i overhears j forward the packet to the next hop on the route, say node k. It compares the overheard packet with the information in the PACK queue and verifies that the changes are legitimate. It thus infers correct reception of the packet by j and the attempt of j to forward it to k. Node i interprets this as normal behavior by j and removes the packet from the PACK queue. To reflect this observation of j, node i creates a first-hand information rating for j, which we call $F_{i,j}$.

2.2. *Updating First-Hand Information*

The first-hand information record $F_{i,j}$ has the form(α, β). It represents the parameters of the Beta distribution assumed by node i in its Bayesian view of node j's behavior as an actor in the network. Initially, it is set to (1, 1). The standard Bayesian method gives the same weight to each observation, regardless of its time of occurrence. We want to give less weight to evidence received in the past to allow for reputation fading. We therefore developed a modified Bayesian update approach by introducing a moving weighted average as follows. Node i just made one individual observation about j. Let S=1 if this observation is qualified as misbehavior by RMP, and S=0 otherwise, the update is $\alpha := u\alpha+s$, $\beta := u\,\beta +(1\text{-}s)$. The weight u is a discount factor for past experiences, which serves as the fading mechanism. In our case, node i classified the behavior of node j as normal, since it overheard the packet re-transmission and detected no illegitimate changes, therefore

$$F_{i,j} = F_{i,j}\ (u\alpha,\ u\ \beta+1)$$

In addition, during inactivity periods, we periodically decay the values of α, β as follows.

Whenever the inactivity time expires, we let $\alpha: = u\alpha$ $\beta := u\,\beta$

This is to allow for redemption even in the absence of observations. Node i thus periodically discounts the parameters of $F_{i,j}$.

2.3. *Updating Reputation Ratings*

When node i updates its first-hand information $F_{i,j}$, it also updates its reputation rating for j, namely $R_{i,j}$ in the same way. The reputation rating $R_{i,j}$ is also defined by two numbers, (α^1, β^1). Initially, it is set to (1, 1). It is updated on two types of events:

(1) When first-hand observation is updated

(2) When a reputation rating published by some other node is copied. Here we discuss the first case.

So far, node i has made one first-hand observation of node j. Since it made a positive experience with node j, it changes $R_{i,j} = R_{i,j}\,(u\alpha^1,\ u\beta^1+1)$. If the update to the first-hand information is due to inactivity, the formula is

$$\alpha^1: = u\alpha^1,\ \beta^1:= u\ \beta^1$$

2.4. *Using Trust*

To speed up detection, nodes can also use trust to accept second-hand information even if it is incompatible. Assume node i receives the reported first-hand information $F_{k,j}$ from node k. If $F_{i,k}$ is high enough, it will accept $F_{k,j}$ to slightly modify its own $R_{i,j}$ even if it fails the deviation test. Node i updates $R_{i,j}$ in any case. If passed the k deviation test, δ will be increased, otherwise γ.

2.5. *Classifying Nodes*

Every time node i updates its ratings about j, it checks whether it is still within the boundaries of its misbehavior tolerance. This is done to provide a basis for decisions about how to treat j. Node i thus classifies j as normal, if $R_{i,j}$ is smaller than t, as misbehaved otherwise.

2.6. *Sending Packets, Detecting Misbehavior*

For each packet node i sends, it keeps the same procedure of storing the information in he PACK queue and setting the PACK timer. When the PACK timer goes off, it means that node i did not overhear the retransmission of the packet by the next hop j. In this case, node i interprets this as an instance of misbehavior by node j and updates its first hand information and reputation rating about node j, such that

$F_{i,j} = F_{i,j}$ (uα^1+1, u β^1) and

$R_{i,j}$ (α^1, β^1) = $R_{i,j}$ (uα^1+1, u β^1).

The PACK timer going off is only one case of a misbehavior indication, another one is when node i detects an illegitimate modification of the packet when it overhears the retransmission by j. When there are no packets being sent, node i updates $F_{i,j}$ and $R_{i,j}$ using the decay factor u.

2.7. Managing Paths

When i classify j as misbehaving, it deletes all routes containing node j from its path cache. If it still has packets to send and there is an alternate path that does not include j, node i proceeds to send packets over that path, otherwise it sends out a new route request. In addition, node i puts node j on its list of misbehaving nodes and increases its reputation tolerance threshold r. Assume now that node j wants the services of node i for forwarding a packet node originating from j or providing a route for j. Node i deny service to j in order to retaliate and isolate it. In our approach, we do not punish nodes that are categorized as untrustworthy but merely restrict their influence. The reasons for this are that testimonial inaccuracy can not be proved beyond doubt, deviations can arise because nodes discover misbehavior before others do, and punishment discourages the publication of ratings.

3. Monitoring by Enhanced passive Acknowledgement

When a RMP node, say node i joins a mobile ad-hoc network running DSR, its path cache is empty and it has no first-information, trust, or reputation ratings about others. When it has a packet to send, it first sends out a route request, and after receiving route replies according to DSR, it chooses the shortest path and puts it in its route cache. Let node j be the next-hop node on the source route to the destination. Node I then sends its packet to node j. After sending the packet to node i, node j put packet information into the queue for passive acknowledgment (PACK) and sets a PACK timer. Every time i overhear a packet, it checks whether it matches an entry in the PACK table.

4. Design Approach

4.1 Architecture

The tasks RMP carries out are, to gather information to classify first-hand experience, to exchange this information and to consider the second-hand information thus received, to update the belief about the behavior of others, which is called the reputation rating, taking into account both first and second-hand information, to classify other nodes based on the reputation rating, and to adapt one's own behavior according to that classification. RMP consists of several components that fulfill these tasks. The architecture of the protocol is as shown in figure 4.1. The Monitor, the Reputation System, the Path Manager, and the Trust Manager are the components that are present in every node and they are described in detail subsequently.

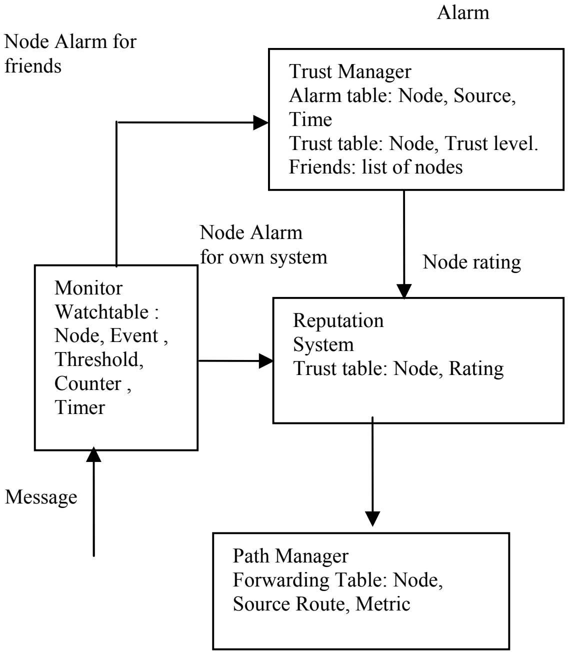
Alarm
Node Alarm for friends
Trust Manager
Alarm table: Node, Source, Time
Trust table: Node, Trust level.
Friends: list of nodes
Node Alarm for own system
Node rating
Monitor
Watchtable : Node, Event , Threshold, Counter , Timer
Reputation System
Trust table: Node, Rating
Message
Path Manager
Forwarding Table: Node, Source Route, Metric

Figure 4.1: RMP Architecture within each Node

4.1.1. The Monitor (Neighborhood Watch)

In a networking environment, the nodes most likely to detect non-compliant 'criminal' behavior are the nodes in the vicinity of the criminal and in some case the source and the destination, if they detect unusual behavior or do not get proper responses. The latter is not always the case, for instance in the case of replay. One approach to protocol enforcement and detection of damaging behavior (intrusion, misuse of cooperation incentives, denial of service, etc.) suggested here is the equivalent of a 'neighborhood watch', where nodes locally look for deviating nodes. The neighbors of the neighborhood watch can detect deviances by the next node on the source route by either listening to the transmission of the next node or by observing route protocol behavior. By keeping a copy of a packet while listening to the transmission of the next node, any content change can also be detected. In this paper we focused on the detection of observable routing and forwarding misbehavior in DSR as listed in section 5.2. In general, the following types of misbehavior can be indicated:

- no forwarding (of control messages nor data),
- unusual traffic attraction (advertises many very good routes or advertises routes very fast, so they are deemed good routes),
- route salvaging (i.e. rerouting to avoid a broken link),although no error has been observed,
- lack of error messages, although an error has been observed,
- unusually frequent route updates,
- silent route change (tampering with the message header of either control or data packets).

As a component within each node, the monitor registers these deviations of normal behavior. As soon as a given bad behavior occurs, the reputation system is called.

4.1.2. The Trust Manager

In an ad hoc environment, trust management has to be distributed and adaptive [2]. This component deals with incoming and outgoing ALARM messages. ALARM messages are sent by the trust manager of a node to warn others of malicious nodes. Incoming ALARMS originate from outside friends, whereas the node itself generates outgoing ALARMS after having experienced, observed or been reported malicious behavior.

The following functions are performed by the trust manager:

- Trust function to calculate trust levels, trust table entries management for trust level administration,
- Forwarding of ALARM messages,
- Filtering of incoming ALARM messages according to the trust level of the reporting node.
- The trust manager consists of the following components,
- Alarm table containing information about received alarms,
- Trust table managing trust levels for nodes,
- Friends list containing all friends a node sends alarms to.
- The trust manager administers a table of friends, i.e. nodes that are candidates to receive ALARM messages from a given node, and how much they are trusted.

Trust is important when making a decision about the following issues:

i. providing or accepting routing information,
ii. accepting a node as part of a route,
iii. taking part in a route originated by some other node.

4.1.3. The Reputation System (Node Rating)

In order to avoid centralized rating, local rating lists and/or black lists are maintained at each node and potentially exchanged with friends. The nodes can include black sheep in the route request to be avoided for routing, which also alarms nodes on the way. Nodes can look up senders in the black list containing the nodes with bad rating before forwarding anything for them. The problem of how to distinguish alleged from proven malicious nodes and thus how to avoid false accusations can be lessened by timeout and subsequent recovery or revocation lists of nodes that have behaved well for a specified period of time. Another problem is scalability and how to avoid blown-up lists, which can also be addressed by timeouts. The reputation system in this protocol manages a table consisting of entries for nodes and their rating. The rating is changed only when there is enough evidence for malicious behavior that is significant for a node and that has occurred a number of times exceeding a threshold to rule out coincidences. The rating is then changed according to a rate function that assigns different weights to the type of behavior detection:

- Own experience: greatest weight,
- Observations: smaller weight,
- Reported experience: weight function according to PGP trust.

Once the weight has been determined, the entry of the node that misbehaved is changed accordingly. If the rating of a node in the table has deteriorated so much as to fall out of a tolerable range, the path manager is called for action. Bearing in mind that malicious behavior will hopefully be the exception and not the rule, the reputation system is built on negative experience rather than positive impressions.

4.1.4. The Path Manager

Once a node i classifies another node j as misbehaving, i isolates j from communications by not using j for routing and forwarding and by not allowing j to use i. This isolation has three purposes. The first is to reduce the effect of misbehavior by depriving the misbehaving node of the opportunity to participate in the network. The second purpose is to serve as an incentive to behave well in order not to be denied service. Finally, the third purpose is to obtain better service by not using misbehaving nodes on the path. The path manager performs the following functions:

- Path re-ranking according to security metric,
- Deletion of paths containing malicious nodes,
- Action on receiving a request for a route from a malicious node (e.g. ignore, do not send any reply)
- Action on receiving request for a route containing a malicious node in the source route (e.g. also ignore, alert the source).

The dynamic behavior of RMP is as follows [2]. Nodes monitor their neighbors and change the reputation accordingly. If they have reason to believe that a node misbehaves, i.e. when the reputation rating is bad, they take action in terms of their own routing and forwarding. They thus route around suspected misbehaved nodes. Depending on the rating and the availability of paths to the destination, the routes containing the misbehaved node are either re-ranked or deleted from the path cache. Future requests by the badly rated node are ignored. In addition, once a node has detected a misbehaved node, it informs other nodes by sending an ALARM message.

When a node receives such an ALARM either directly or by promiscuously listening to the network, it evaluates how trustworthy the ALARM is based on the source of the ALARM and the accumulated ALARM messages about the node in question. It can then decide whether to take action against the misbehaving node. Note that simply not forwarding is just one of the possible types of misbehavior in mobile ad-hoc networks. Several others, mostly concerned with routing rather that forwarding have been suggested, such as black hole routing, gray hole routing, worm hole routing. Other kinds of misbehavior aim at draining energy, such as the sleep deprivation attack. RMP is not restricted to handling any particular kind of misbehavior but can handle any attack that is observable. Even if the observation cannot precisely be attributed to an attack but is the result of another circumstance in the network such as a collision, RMP can make use of it. If it is a rare accident, it will anyhow not influence the reputation rating significantly, and if it happens more often, it means the observed node has difficulties performing its tasks.

4.2. Context Diagram for RMP

Context Diagram for RMP sa shown in Figure 4.2. Every node uses RMP Process for the transferring of packets to route around the malicious nodes and to evaluate the performance in terms of Percentage of Misbehaving nodes, numbers of rejected path, total Hop count, transmission delay and Good put.

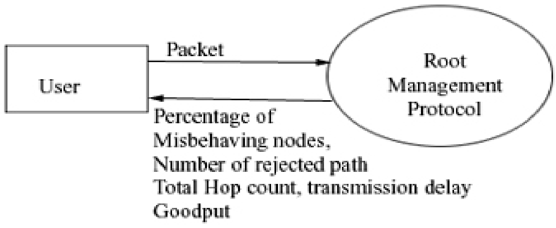

Figure 4.2: Context Diagram for Route Management Protocol(RPM)

4.3. Level 1 DFD for Route Management Protocol

Level 1 DFD for Route Management Protocol as shown in Figure 4.3. The monitor process receives PACK message & observes the behavior of neighboring nodes, sends alarm to trust management process and Reputation system process if the node misbehaves. The Trust Manager process in turn sends this alarm to friend nodes, and if it receives alarm massages, it evaluates the node rating and sends to the reputation system. The reputation system process evaluates the reputation rating and sends to the path manager. The path manager process modifies the path information based on the reputation rating.

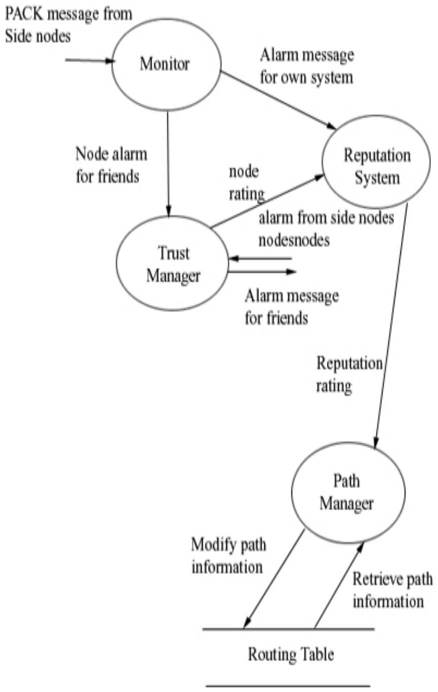

Figure 4.3: Level 1 DFD for Route Management Protocol

4.4. Level 2 DFD for Trust Manager

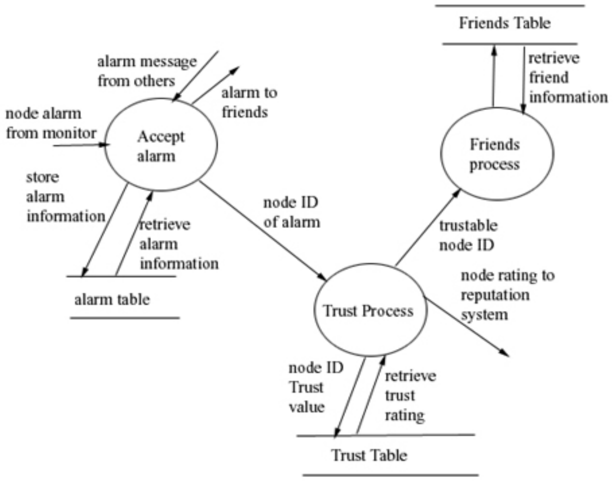

Figure 4.4: Level 2 DFD for Trust Manager

Level 2 DFD for Trust Manager as shown in the Figure 4.4. Accept alarm process receives the alarm messages from monitor and nodes and stores in the alarm table and it retrieves the node ID of alarm received to the Trust process. It also sends alarms to the friend nodes. The trust process retrieves the Trust rating from Trust table and sends the node rating to the path Manager process.

4.5. Level 2 DFD for Reputation System and Path Manager

Level 2 DFD for Reputation System and Path Manager as shown in the Figure 4.5 The weight process receives the input from the Monitor process, Trust manager process, and calculates a weight and sends the reputation information to the Rating function process, which in turn calculates the reputation rating and sends to the path Manager process, The Path Manager process compares the reputation rating with the tolerable range and either changes the ranking of the path or deletes the path from the routing table.

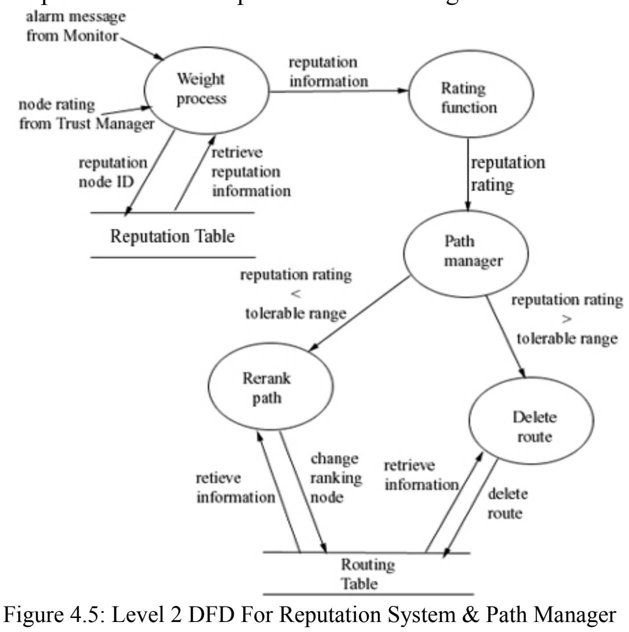

Figure 4.5: Level 2 DFD For Reputation System & Path Manager

5. IMPLEMENTATION

5.1. Network Creation

For the creations of the network for simulation, an area of 280*300 units is chosen. The nodes are randomly created by allocating their coordinates and with random BW and ID allocated. These nodes are plotted over a scale is randomly chosen with a destination. This module then implements a DSR protocol where a packet is generated from the source with a structure explained in section two. This packet is forwarded to their neighboring nodes maintaining a node list during forwarding the packets and return back an acknowledge from the destination from the same node as maintained in the list once the destination is reached. The module carries out this operation for all randomly distributed nodes to extract all possible paths from source to destination. Based on the number of Hops in the path the shortest path is chosen for analysis.

5.2. Evaluating a path between source and destination

For the source chosen, the packets generated rate transferred over the shortest path and observed whether a destination is reached or not. This module gives an option for selecting a particular node as regular or misbehaving based on which the reputation of each node is evaluated.

5.3. Finding a Friend or Malicious node

Based on the PACK received from the next node in the path, the HOP count field and the TTL field are compared with the same fields of the packet in PACK queue to determine whether the next node has forwarded the packet or not. If these fields are found randomly modified, the node will be processed for misbehaving else will be declared as a friend. During misbehaving evaluation this module reads few network parameters as r, t, α^1, β^1, γ, ϑ for deciding the node property and trustworthiness.

5.4 Isolation of Malicious node based on Bayesian Approach

This module evaluates the node performance and decides to retain the node in path or isolates based on modified Bayesian approach. The modified Bayesian approach is presented in section- 2. This module reads the network parameter from previous module.

5.5. The Network Performance Evaluation

This module simulates the network for various combinations with misbehaving varying from 0 to maximum limit. This module evaluates transmitting delay, excess HOP count, good put and number of rejected paths to decide the efficiency of RMP for randomly distributed Ad-hoc network.

6. TESTING AND VALIDATION

6.1. Verification and Validation

Verification and Validation is the generic name given to checking processes, which ensures that software conforms to its specification and meets the specification of the customer.

The difference between Verification and Validation

Validation: are we building the right product?

Verification: are we building the product right?

Verification involves checking that the program conforms to its specification; Validation involves checking that the program as implemented meets the expectations of the customers. In this project also verification and validation are used for testing the system, whether it meets user requirements are not. the user requirements are discussed in the section-3 of this report.

6.2. Testing

All the modules specified in section 5 are tested by giving the input parameters as specified in section 7, table 7.1 to check whether the system efficiently works or not in the presence of malicious node

7. RESULTS ANALYSIS

7.1. Considered Network for Simulation

Considered Network for Simulation and the Network Parameters as given in Table 7.1 & 7.2

Distribution	**Random**
Number of Nodes	17
Region	280 X 300 units
Communication Range	80 units
Mobility	Static
MAC	802.11
Packet Size	61 BITS
Weight (w)	0.1
Trustworthy Threshold (t)	0.75
Node status threshold (r)	0.5

Table 7.1 Network Parameters

Percentage of Misbehaving Nodes	**20%**	**40%**	**60%**	**80%**
No. of rejected paths	2	2	4	6
Total Hops under communication	2	2	2	6
Transmission delay in seconds	2	2	2	6
% Good put	66.67	66.67	33.33	0

Observations Table 7.2

7.2. Analysis

Average path rejections with respect to misbehaving nodes as shown in Figure 7.9 The average rejected paths increases if percentage of malicious nodes increases but with the use of RMP average paths rejected remains constant even if the percentage of malicious nodes increases to 40%.

Total Hops under communication with respect to percentage of misbehavior plot as shown in Figure 7.10. The number of rejected path from the source to destination increases as percentage of misbehaving nodes increases hence the number of hop counts required for communication also increases. The total hop counts for communication remains constant with the use of RMP (Route Management protocol) even if percentage of malicious nodes increases to 60%.

Transmission Delay versus % Misbehavior plot. The packet transmission delay increases with the increase in percentage of malicious nodes but with use of RMP (Route Management Protocol) the transmission delay remains constant even if the percentage of malicious nodes increases to 60%.Good put plot for the network and the observations are given in table 7.2.

8. CONCLUSION

Ad Hoc network is one of the evolving research and application area in wireless communication. The network finds its need in various fields such as battlefields, natural disaster etc where no other communication system provided to be better. However, this network is constrained by its own limitations and results in lower performance in real time scenario. One of the major limitations found in today's Ad hoc network is the issue of misbehavior. This paper explores this issue on a randomly distributed network and proposes a protocol called RMP to overcome this limitation. The protocol is integrated with modified Bayesian approach to desire the node network whether it is misbehaving or not. From the observation made during the simulation of the network, it is found that with increase in percentage of misbehaving node in the network the paths available from source to destination fall down and almost collapse when it becomes maximum, the number of Hops taken increases, transmission delay increases and good put decreases. From all the above observation made it is concluded that node with RMP can sustain the network with efficient data transmission for 50% of misbehaving node.

REFERENCES

[1] A review of routing protocols for mobile ad hoc networks: by Mehran Abolhasan, Tadeusz wysocki, Eryk Dutkiewicz 2003.

[2] Self policing mobile ad hoc network, Sonja Bucher and Jean-Yves le, EPFL-IC- lca, LCA, CH-1015, Lausanne.

[3] Levente Butty_an and Jean-Pierre Hubaux. Enforcing service availability in mobile Ad hoc wans. In Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA, August, 2000.

[4] Levente Butty_an and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. MONET Journal of Mobile Networks, to appear 2002.

[5] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM 2000*, pages 255–265, 2000.

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

[7] Marco Carbone, Mogens Nielsen, and Vladimiro Sassone. A formal model for trust in dynamic networks. BRICS Report RS- 03-4, 2003.

[8] Dave B. Johnson and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks.Internet Draft, Mobile Ad Hoc Network (MANET) Working Group, IETF, October 1999.

[9] A Robust reputation system for mobile ad hoc networks EPFL-IC-LCA, CH-105 Lausanne.

Shalini received my B. Tech Degree from JNTU Hyderabad in Computer Science and also received M.Tech Degree from JNTU Hyderabad. Currently working as Assistant professor, having interest to research in network systems, network security environments

Krishna Rao received his B.Tech Degree from JNTU Anantapur, masters Degree from JNTU Hyderabad. Currently working as associate professor, and pursuing Ph. D. He is having interest in networking and data warehousing.