IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 4, Ver. II (Jul.-Aug. 2016), PP 25-37
www.iosrjournals.org

Phernome Based Node Tracking and Routing For Adhoc Networks

Ugendhar Addagatla [1], **Dr. V. Janaki** [2]
[1]*(Department of C.S.E, Guru Nanak Institutions Technical Campus, Ranga Reddy, Telangana-501506)*
[2]*(Department of C.S.E, Vaagdevi Engineering College, Warangal, Telangana-506005)*

Abstract: *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. The main objective of this paper is to cope with weight. The proposed Route Tracking Protocol helps an existing system to cope with weight. As a concrete instantiation of such an existing system, we chose mobile ad-hoc networks (MANETs) running Dynamic Source Routing (DSR) and applied ANT to it.*

Keywords: *Dynamic Source Routing, MANETs, Mobility, Route Tracking Protocol, Wireless Interfaces*

I. Introduction

The 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 [1] 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. The lack of infrastructure and organizational environment of mobile ad hoc networks offer special opportunities to attackers. Without proper security, it is possible to gain various advantages by interfering behavior: better service than cooperating nodes, monetary benefits by exploiting incentive measures or trading confidential information; saving power by blocking behavior, preventing someone else from getting proper service, extracting data to get confidential information, and so on. Weight means deviation from normal routing and forwarding behavior. Without appropriate countermeasures, the effects of weight have been shown to dramatically decrease network performance. Depending on the proportion of blocking nodes and their specific strategies, network throughput can be severely degraded, packet loss increases, nodes can be denied service, and the network can be partitioned. These detrimental effects of weight can endanger the functioning of the entire network. The problem we want to solve is the following. How can we make an existing system keep working despite the presence of weight. As a specific application to the case of a mobile ad-hoc network, how can we keep the network functional for normal nodes when other nodes do not route and forward correctly.

The approach used in ANT is to detect blocking nodes and to render them harmless, regardless of the reason of their weight, be it blocking, interfering, or faulty. The response to detected blocking nodes is to isolate them, so that weight will not pay off but result in denied service and thus cannot continue. ANT detects blocking nodes by means of direct observation or second-Node information about several types of blockage, thus allowing nodes to route around blocking nodes and to isolate them. Buttyan 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 con- forms to our results. The main differences to the ANT protocol are that nuglets or counters are limited to a one-to-one interaction, whereas in the ANT protocol, weight 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 `path rater' (for reliability tracking 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 interfering 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 interfering nodes harder to detect. A formal model for reliability in dynamic networks based on intervals and a policy language has been proposed by Carbone, Nielsen, and Sassone [7]. They express both reliability and the uncertainty of it as reliability ordering and information ordering, respectively. They consider the delegation of reliability to other principals. In their model, only positive information influences reliability, such that the information ordering and the reliability ordering can differ. In our system, both positive and negative information influence the reliability 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 [8]. Byzantine robustness [9] in the sense of being able to tolerate a number of erratically behaving servers or in these case nodes is the goal of a reputation system in mobile ad-hoc networks. Here, the detection of interfering nodes by means of the reputation systems has to be followed by a response in order to render these nodes harmless.

This work analyze the performance of a Ad hoc network with blocking node falling in data transfer route, and propose a protocol to overcome the blocking node to enhance the performance of Ad hoc networks. The proposed algorithm for tracking scheme would be best suited for following applications:

1. The battlefield scenario where each node has to transfer various important information, any blocking node may loose some very important information to be communicated.
2. Disaster recoveries
3. Environmental monitoring etc

In this paper, the ANT protocol is applied on DSR. Nodes in ad hoc network have a monitor for observations, reputation records for Direct-Node observation and reliable second-Node observations about routing and forwarding behavior of other nodes, reliability records to control reliability given to received warnings, and a path manager to adapt their behavior according to reputation and to take action against weight nodes. The term reputation is used to evaluate the routing and forwarding behavior according to the network protocol, whereas the term reliability is used to evaluate participation in the ANT Protocol.

The approach in this paper is as follows;

- Place N mobile nodes on a plane
- Find all possible paths from a source node to a destination node
- Find shortest path from the discovered paths
- Select M nodes to be interfering on the shortest path
- For each node select F readys
- Generate traffic between them
- Gather statistics on throughput /overhead etc by varying parameters.

A network has to be created in a selected area with randomly distributed nodes. With the user selection a source and a destination node are to be chosen. All possible paths are to be found from source to destination. With the user option, nodes on the shortest path can be chosen either as normal or blocking and then based on the number of blocking nodes on the selected paths; the performance of the protocol will be evaluated.

II. Route Governing in Ad Hoc Network

Mobile ad hoc networks (MANETs) rely on the cooperation of all the participating nodes. The more nodes cooperate to transfer traffic, the more powerful a MANET gets. But supporting a MANET is a cost-intensive activity for a mobile node. Detecting routes and forwarding packets consumes local CPU time, memory, network-bandwidth, and most important the energy. Therefore there is a strong motivation for a node to deny packet forwarding to others, while at the same time using their services to deliver own data. There are two approaches of dealing with blocking nodes. The Direct approach tries to give a motivation for participating in the network function. The authors suggest to introduce a virtual currency called Nuglets that is earned by relaying foreign traffic and spent by sending own traffic. The major drawback of this approach is the demand for reliable hardware to secure the currency. There are arguments that tamper-resistant devices in general might be next to impossible to be realized. A similar approach without the need of tamper proof hardware has been suggested by Zhong. There exist also other unresolved problems with virtual currencies, like e.g. nodes may starve at the edge of the network because no one needs them for forwarding etc. Most of the existing work in this field concentrates on the second approach: detecting and excluding blocking nodes. The Direct to propose a solution to the problem of blocking (or as they call it "blocking") nodes in an ad hoc network were Marti, Giuli, Lai and Baker. Their system uses a watchdog that monitors the neighboring nodes to check if they actually relay the data the way they should do. Then a component called path rater will try to prevent paths which contain such blocking nodes. As they indicate their detection mechanism has a number of severe drawbacks. Relying only on overhearing transmissions in promiscuous mode may fail due to a number of reasons. In case of sensor failure, nodes may be falsely accused of weight. The second drawback is that blocking nodes profit from being recognized as blocking. The paths in the network are then routed around them, but there is no exclusion from service.

A wireless or mobile ad hoc network (MANET) is formed by a group of wireless nodes which agree to forward packets for each other. One assumption made by most ad hoc routing protocols is that every node is reliable and cooperative. In other words, if a node claims it can reach another node by a certain path or distance, the claim is reliability. If a node reports a link break, the link will no longer be used. Although such an assumption can simplify the design and implementation of ad hoc routing protocols, it does make ad hoc networks vulnerable to various types of denial of service (DoS) blockage. One class of DoS blockage is interfering packet dropping. An interfering node can silently drop some or all of the data packets sent to it for further forwarding even when no congestion occurs. Interfering packet dropping attack presents a new threat to wireless ad hoc networks since they lack physical protection and strong access control mechanism. An adversary can easily join the network or capture a mobile node and then starts to disrupt network communication by silently dropping packets. It is also a threat to the Internet since the various software vulnerabilities would allow attackers to gain remote control of routers on the Internet. If interfering packet dropping attack is used along with other attacking techniques, such as shorter distance fraud, it can create more powerful blockage (i.e., black hole) which may completely disrupt network communication. Current network protocols do not have the capability to detect the interfering packet dropping attack. Network congestion control mechanisms do not apply here since packets are not dropped due to congestion. Link layer acknowledgment, such as IEEE 802.11 MAC protocol, can detect link layer break, but cannot detect forwarding level break. Although upper layer acknowledgment, such as TCP ACK, allows for detecting end-to end communication break, it can be inefficient and it does not indicate the node at which the communication breaks. Moreover such mechanism is not available in connectionless transport layer protocols, such as UDP. Therefore, it is important to develop mechanisms to render networks the robustness for resisting the interfering packet dropping attack.

III. Properties of Weight Monitoring in Ad Hoc Network

We found the following ways of attacking DSR, targeting availability, integrity, confidentiality, non-repudiation, authentication, access control or any combination thereof:

1) Incorrect forwarding: acknowledge ROUTE REQUEST, send new request or do not forward at all. This works only until upper layers find out.
2) Bogus routing information or traffic attraction: reply to ROUTE REQUEST, also gratuitous, to advertise a non-existent or wrong route.
3) Salvage a route that is not broken. If the salvage bit is not set, it will look like the source is still the original one.
4) Choose a very short reply time, so the route will be prioritized and stay in the cache longer.
5) Set good metrics of bogus routes for priority and remaining time in the cache.
6) Manipulate flow metrics for the same reason.
7) Do not send error messages in order to prevent other nodes from looking for alternative routes.
8) Use bogus routes to attract traffic to intercept packets and gather information.
9) Use promiscuous mode to listen in on traffic destined for another node.
10) Cause a denial-of-service attack caused by overload by sending route updates at short intervals.

IV. Detection of Blockage in DSR

With the exception of the promiscuous listening in 9), the entire blockage listed above corresponds to observable events the monitor component in each node can detect either at once or at the latest when they happen repeatedly:

1) Forwarding: this can be detected by passive acknowledgement, i.e. keeping a copy of a packet until having confirmed correct forwarding by listening to the transmission of the next hop node.
2) Bogus routing: a strong indication would be when an intermediate node sees itself advertised on a route it does not have. As a last resort, if a node cannot tell whether a route is real or bogus, it can at least detect

the lack of forwarding as in 1). Unusually increased frequency of route advertising can be detected as in 10).

3) Salvaging: indicated by the reception of a salvaged packet without having received a link error message Direct.
4) Reply time too short: can be detected by comparing reply time to actual route length.
5) Metrics of bogus routes too good: detectable by comparing metrics to actual quality.
6) Lack of error messages: indicated in the case when a node receives a link error message from its own link layer but no explicit error message by other nodes in the range.
7) Route updates too frequent: detectable by keeping timestamp of last update to compare.

A. Reluctant Nodes

The suggested scheme works as an extension to a routing protocol. In this example, normal DSR information flow (ROUTE REQUEST, ROUTE REPLY messages) as explained takes place. Once non-cooperative behavior has been detected and exceeds threshold values, an ALARM message is sent. Fig. s given show the flow of messages and data from route discovery to the detection of interfering behavior and subsequent rerouting.

In more detail: Fig. 1 shows DSR route discovery for a path from node A to node E. Every node forwards the request to its neighbors unless it has already received the same route request or has a path cache entry for the desired destination.

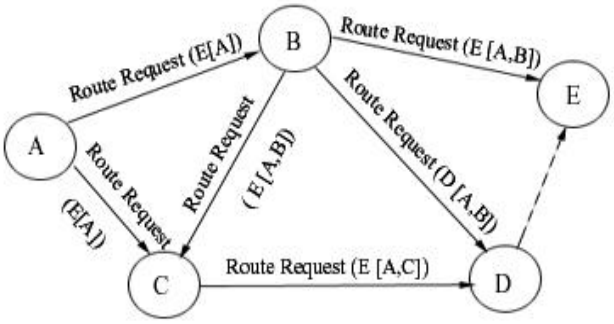


Fig. 1: Route request from node A to E

Fig. 2 shows the reply messages of the destination node itself, node E, and from node D, which has a path to E. The reply message contains the reversed source route to the destination and is sent to the source. In the case of unidirectional links, or if generally the route can not be reversed, node E would send the reply along a path to A that it has in its route cache. If there is no path to A in the route cache, E has to perform a route discovery itself to get to A. In this route request, the already found path from A to E is included.

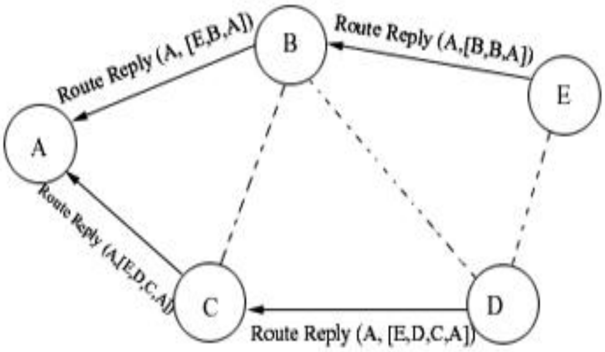


Fig. 2: Route Reply to node A

In Fig. 3 data flow is from node A to node E via node C and D. In this case, node A has chosen this route according to some metrics and preferred it over the route via B. During the data flow, node C detects that node D does not behave correctly. In this example, node D does not forward the data destined for node E. After the occurrence of the bad behavior of node D was observed by node C for a number exceeding a threshold, node C triggers an ALARM message to be sent to the source, node A.

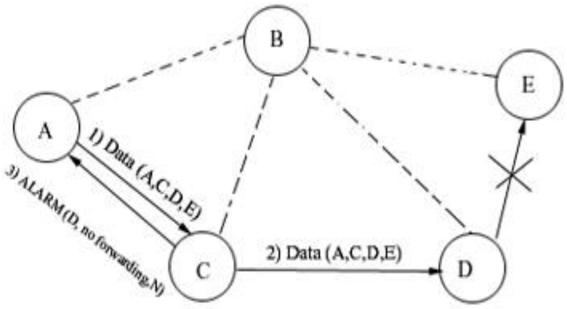


Fig. 3: Data flow and alarm message

Upon reception of the ALARM message as shown in Fig. 4, node A acknowledges the message to the reporting node C and decides to use the alternate path via node B to send the data to the destination node E.

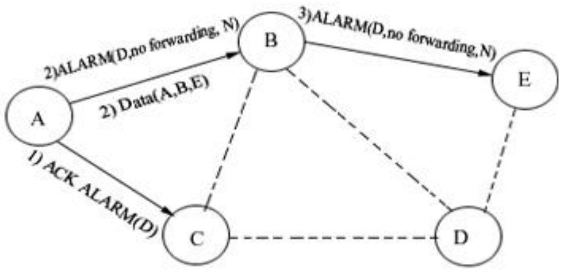


Fig. 4: Data flow through alternate path

Now if node D sends a Route request to the neighboring nodes as shown in Fig. 5, all the nodes do not forward the packet and thus isolates node D.

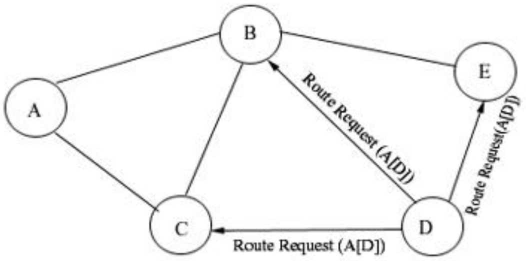


Fig. 5: Isolation of Node D

B. Tracking Scheme

In general, reference node will send the packets to the side nodes as shown in Fig. 6.

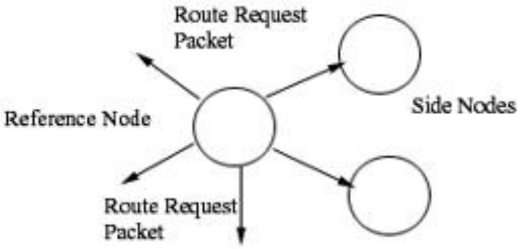


Fig. 6: Route request by reference node

In ANT protocol each node monitor their neighborhood and detect several kinds of weight by means of an enhanced passive acknowledgment mechanism designed. This means that every time a node sends a packet, it listens to overhear whether the next-hop node on the route forwards the packet correctly. Consider the following scenario as depicted in Fig. 7.

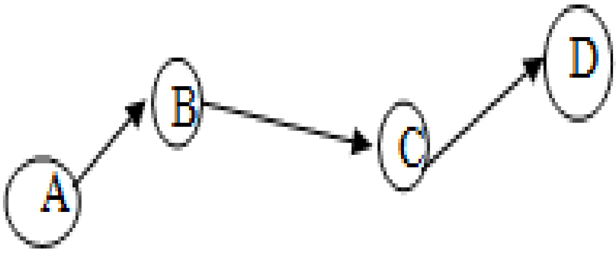


Fig. 7: Packet forwarding between nodes

Node A sends packets via nodes B and C to the destination D. For every packet, nodes keep track of the behavior of the next-hop node and remember whether it has forwarded the packet correctly. A stores ratings about B, B about C, etc., which is called as Direct-Node information, since the ratings are derived from direct observation. Suppose that C misbehaves by dropping the packet instead of forwarding it, as shown in Fig. 2.10. B's rating of C then becomes bad. Since A is not in range with C, it cannot directly observe its behavior and thus cannot find out about C's weight.

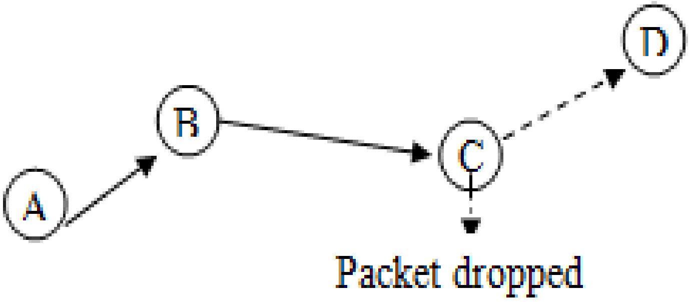


Fig. 8 Packet dropping at node C

In this paper this problem is solved by allowing the use of second-Node information as follows: In addition to keeping track of direct observation, nodes publish their Direct-Node information from time to time by local broadcasts to exchange information with other nodes. This information is termed as second-Node information. A thus receives information from its neighbor B about node C. Again, since A has no Direct-Node information about C, it can only find out about C's weight by second-Node information. There is, however, a problem since second-Node information can be false. A node could for instance make false accusations about another node.

In this paper a combination of two mechanisms is used to cope with spurious second-Node information. Direct, we only consider second-Node information that is not incompatible, i.e. that does not deviate too much from the reputation rating. Our motivation behind this is, that when second-Node information deviates substantially from the rating a node has built over time using previously received second-Node information from several sources and potentially its own Direct-Node information, it is more likely to be false. Second, even when second-Node information is compatible, we only allow it to slightly influence the reputation rating. We modified Bayesian model merging to implement these mechanisms.

Nodes use the reputation ratings they keep about other nodes to classify them. This classification provides a basis for decision-making about providing or accepting routing information, accepting a node as part of a route, and taking part in a route originated by some other node. Nodes classify other nodes as blocking if their reputation rating is worse than their threshold for weight tolerance. Once a node classifies another as blocking, it isolates it from the network by not using it for routing in forwarding and in turn not allowing to be used by it.

V. Monitoring Passive Acknowledgement

When an ANT node, say node i joins a mobile ad-hoc network running DSR, its path cache is empty and it has no Direct-information, reliability, or reputation ratings about others. When it has a packet to send, it direct 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 puts 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.

Gathering Direct-Node 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 Direct-Node information rating for j, which we call F i,j.

Updating Direct-Node Information

The Direct-Node information record Fi,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 weight by ANT, and S=0 otherwise. The update is

$$\alpha: = u\alpha+s$$
$$\beta := u\ \beta +(1-s)$$

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

$$F\ i,j=Fi,j\ (u\alpha,\ u\ \beta+1)$$

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 Fi,j.

When i classifies j as blocking, 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 blocking 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 unreliability worthy but merely restrict their influence. The reasons for this are that testimonial inaccuracy cannot be proved beyond doubt, deviations can arise because nodes discover weight before others do, and punishment discourages the publication of ratings.

VI. Experimental Results

This paper is implemented using 5 modules, they are;

1. Network creation
2. Evaluating a path between source and destination
3. Finding node as a ready or interfering
4. Isolation of interfering node based on Bayesian Approach
5. The Network Performance Evaluation

The above said modules are explained subsequently:

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. 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 blocking based on which the reputation of each node is evaluated. 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 blocking else will be declared as a ready. During blocking evaluation this module reads few network parameters as r,t,α',β',γ,ϑ for deciding the node property and allowance .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.

All the modules specified in section 5 are tested by giving the input parameters as specified to check whether the system efficiently works or not in the presence of interfering node.

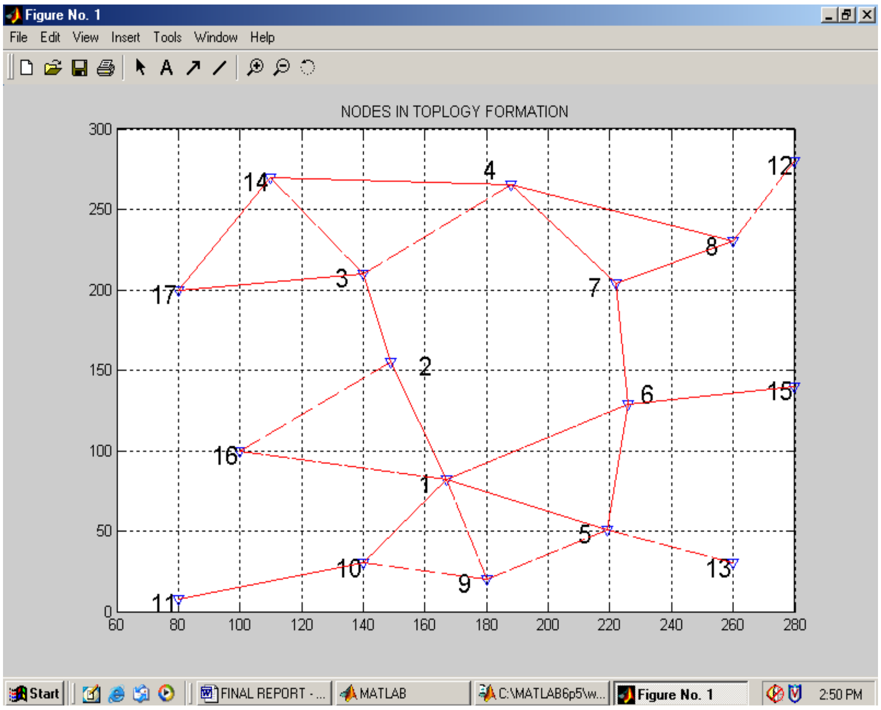
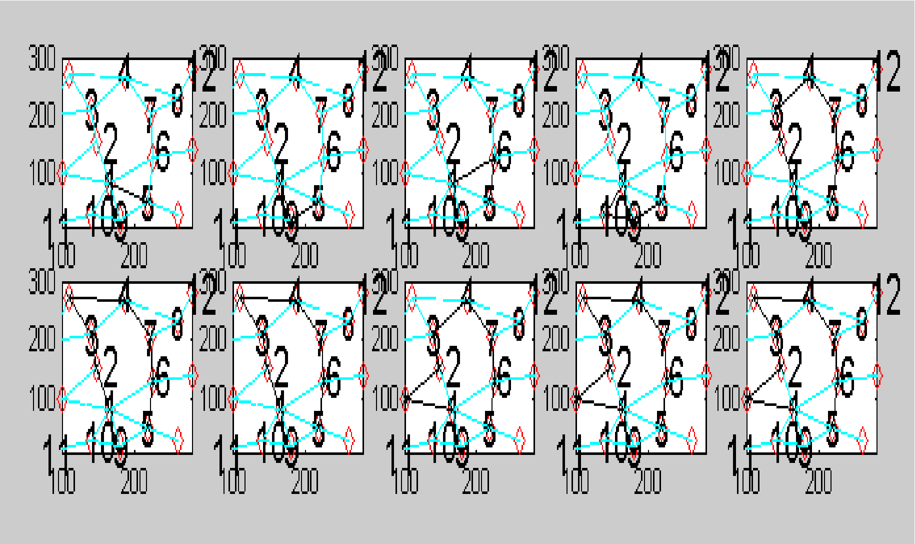


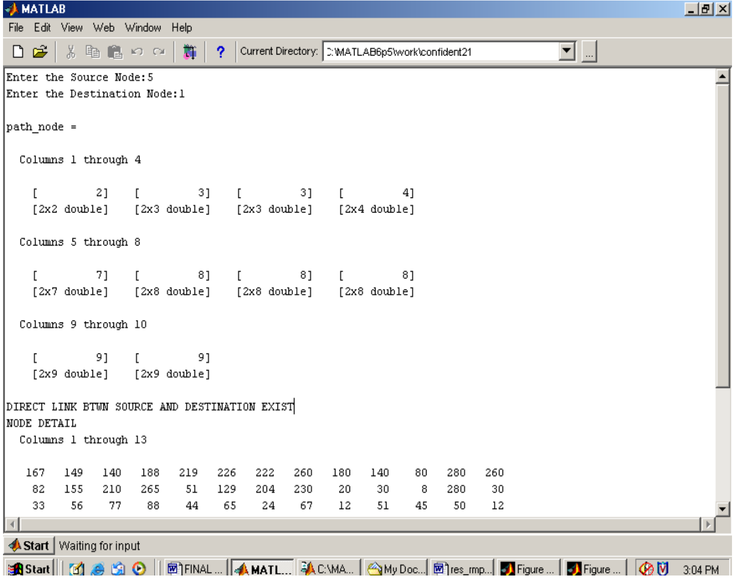
Fig. 9: A randomly distributed network considered for simulation

Simulation Results:

I. **CASE I:** SHORTEST PATH, 1 HOP (DIRECT LINK BETWEEN SOURCE AND DESTINATION)



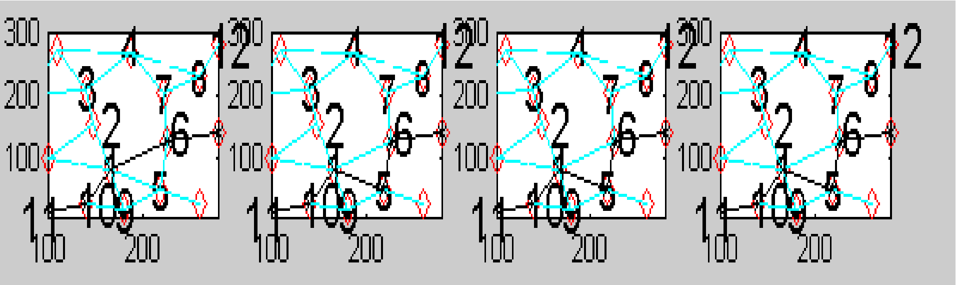
a) Fig. 10: Possible paths from source to Destination with 1 hop link



```
Enter the Source Node:5
Enter the Destination Node:1

path_node =

  Columns 1 through 4

    [         2]    [         3]    [         3]    [         4]
    [2x2 double]    [2x3 double]    [2x3 double]    [2x4 double]

  Columns 5 through 8

    [         7]    [         8]    [         8]    [         8]
    [2x7 double]    [2x8 double]    [2x8 double]    [2x8 double]

  Columns 9 through 10

    [         9]    [         9]
    [2x9 double]    [2x9 double]

DIRECT LINK BTWN SOURCE AND DESTINATION EXIST
NODE DETAIL
  Columns 1 through 13

   167   149   140   188   219   226   222   260   180   140    80   280   260
    82   155   210   265    51   129   204   230    20    30     8   280    30
    33    56    77    88    44    65    24    67    12    51    45    50    12
```

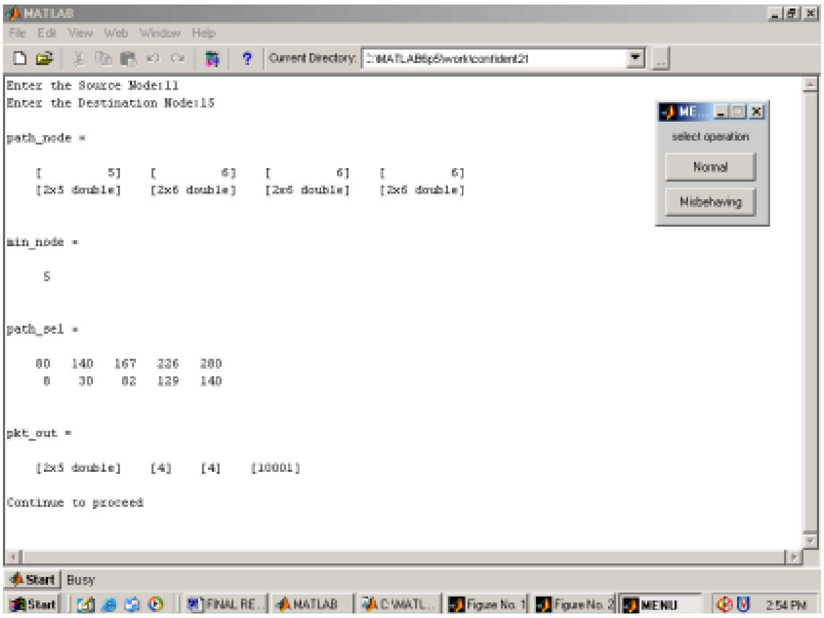
b) Fig. 11: Output of Direct link between source and destination

II. **CASE II:** SHORTEST PATH, MORE THAN 1 HOP (WITH INTERMEDIATE NODES BETWEEN SOURCE AND DESTINATION)



a) **Fig. 12:** Paths from source to destination with more than one hop

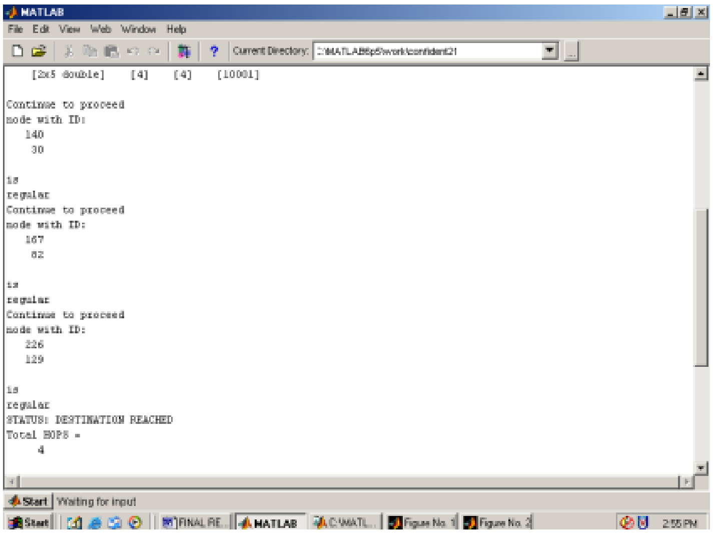
a) With Regular Nodes



```
Enter the Source Node:11
Enter the Destination Node:15

path_node =

    [         5]    [         6]    [         6]    [         6]
    [2x5 double]    [2x6 double]    [2x6 double]    [2x6 double]


min_node =

     5


path_sel =

    80   140   167   226   280
     8    30    82   129   140


pkt_out =

    [2x5 double]    [4]    [4]    [10001]

Continue to proceed
```

Fig. 13: Output of shortest path consisting of regular nodes

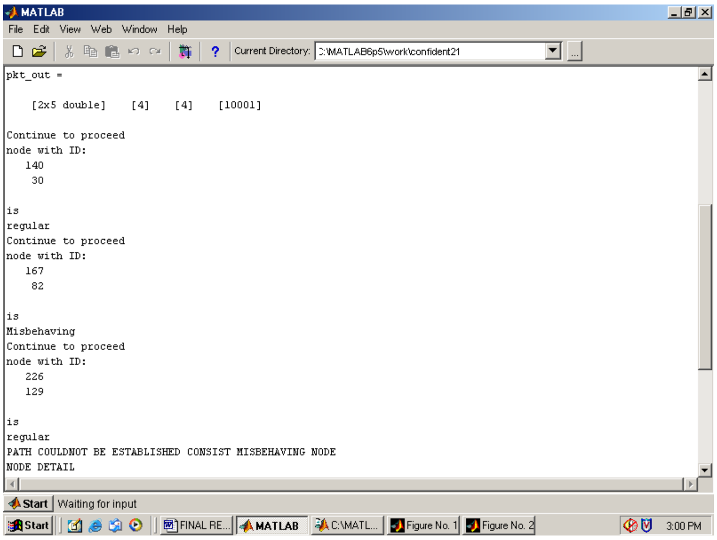
DOI: 10.9790/0661-1804022537



```
    [2x5 double]    [4]    [4]    [10001]

Continue to proceed
node with ID:
   140
    30

is
regular
Continue to proceed
node with ID:
   167
    82

is
regular
Continue to proceed
node with ID:
   226
   129

is
regular
STATUS: DESTINATION REACHED
Total HOPS =
     4
```

Fig. 14: Details of data flow with regular nodes

b) With Interfering nodes

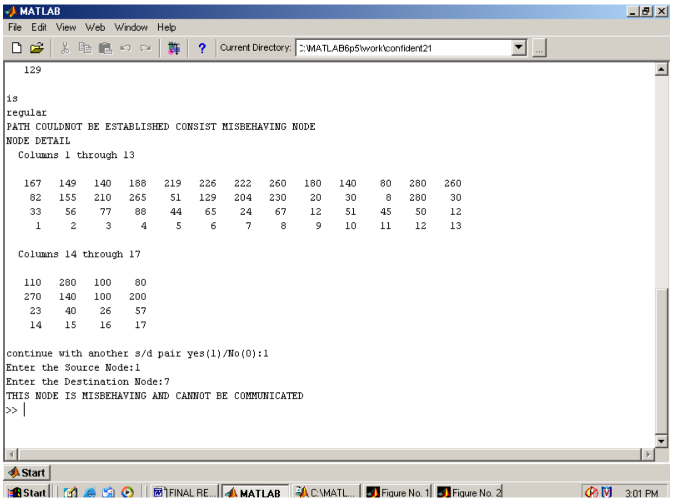


```
pkt_out =

    [2x5 double]    [4]    [4]    [10001]

Continue to proceed
node with ID:
   140
    30

is
regular
Continue to proceed
node with ID:
   167
    82

is
Misbehaving
Continue to proceed
node with ID:
   226
   129

is
regular
PATH COULDNOT BE ESTABLISHED CONSIST MISBEHAVING NODE
NODE DETAIL
```

Fig. 15: Outputs off shortest path consisting of interfering nodes

III. CASE III

Interfering Node as Source


MATLAB
File Edit View Web Window Help
Current Directory: :\MATLAB6p5\work\confident21
Start
FINAL RE...
MATLAB
C:\MATL...
Figure No. 1
Figure No. 2
3:01 PM

```
   129

is
regular
PATH COULDNOT BE ESTABLISHED CONSIST MISBEHAVING NODE
NODE DETAIL
  Columns 1 through 13

   167   149   140   188   219   226   222   260   180   140    80   280   260
    82   155   210   265    51   129   204   230    20    30     8   280    30
    33    56    77    88    44    65    24    67    12    51    45    50    12
     1     2     3     4     5     6     7     8     9    10    11    12    13

  Columns 14 through 17

   110   280   100    80
   270   140   100   200
    23    40    26    57
    14    15    16    17

continue with another s/d pair yes(1)/No(0):1
Enter the Source Node:1
Enter the Destination Node:7
THIS NODE IS MISBEHAVING AND CANNOT BE COMMUNICATED
>> |
```

Fig. 16: Output with interfering node as source

Analysis

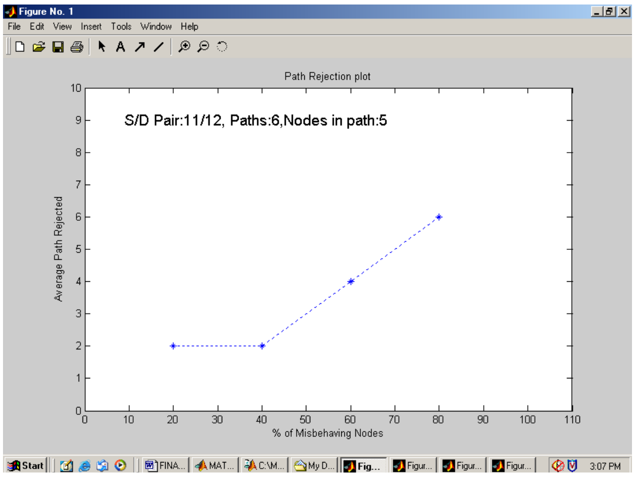


Fig. 17: Average path rejections w.r.to Blocking nodes

The average rejected paths increases if percentage of interfering nodes increases but with the use of ANT average paths rejected remains constant even if the percentage of interfering nodes increases to 40%.

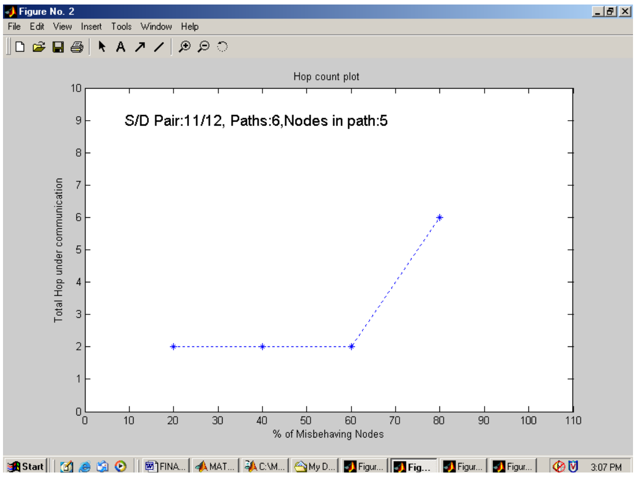


Fig. 18: Total Hops under communication wrt. Percentage of weight plot

The number of rejected path from the source to destination increases as percentage of blocking 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 ANT (Route Tracking protocol) even if percentage of interfering nodes increases to 60%.

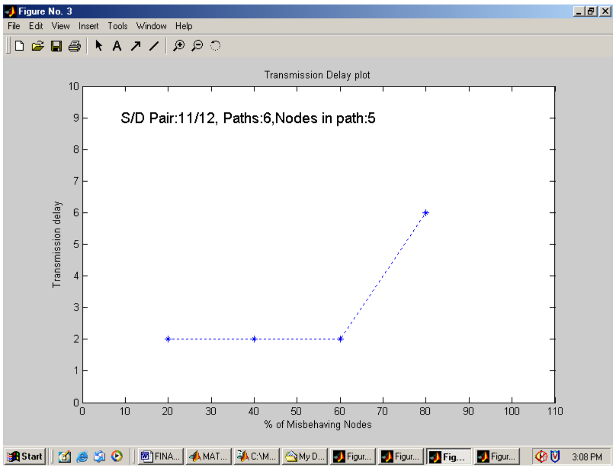


Fig. 19: Transmission Delay versus % Weight plot

The packet transmission delay increases with the increase in percentage of interfering nodes but with use of ANT (Route Tracking Protocol) the transmission delay remains constant even if the percentage of interfering nodes increases to 60%.

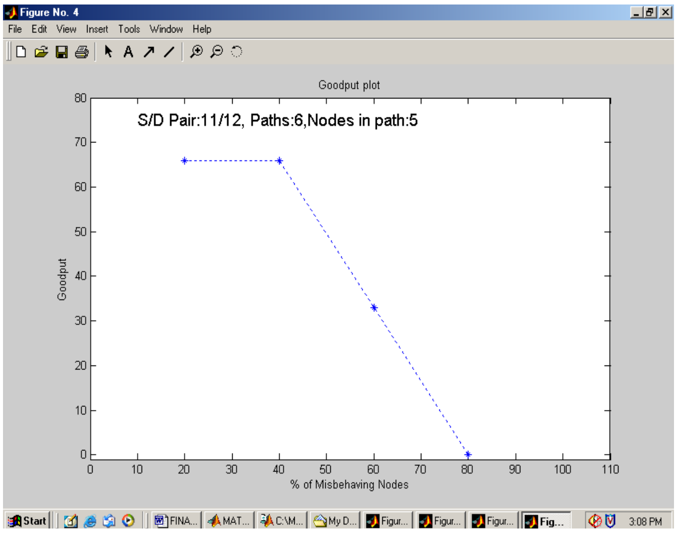


Fig. 20: Goodput plot for the network

VII. 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 weight. This paper explores this issue on a randomly distributed network and proposes a protocol called ANT to overcome this limitation. The protocol is integrated with modified Bayesian approach to desire the node network whether it is blocking or not. From the observation made during the simulation of the network, it is found that with increase in percentage of blocking 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 ANT can sustain the network with efficient data transmission for 50% of blocking node.

References

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

[2] 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.

[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 weight 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 reliabilty in dynamic networks. BRICS Report RS-03-4, 2003.

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

[9] Radia Perlman. Network layer protocols with byzantine robustness. PhD. Thesis Massachussetts Institute of Technology, 1988.

Author Profile

Ugendhar Addagatla presently working as Associate professor in the department of Computer Science and Engineering at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Hyderabad, Telangana State, INDIA. He has 12 years of teaching experience. He is associated with ISTE and CSI as life member. He has obtained B. Tech. degree in Computer Science and Engineering from Christu Jyothi Institute of Technology and Science, Warangal, Jawaharlal Nehru Technological University Hyderabad, in 2003, M.Tech. degree in Software Engineering from Ramappa Engineering College, Warangal, Jawaharlal Nehru Technological University Hyderabad, in 2008 and my area of Research interest is Mobile

Computing, Ph.D (CSE) from Jawaharlal Nehru Technological University, Hyderabad and it is my part of Research work.

Dr. V. Janaki received Ph.D degree from J.N.T. University Hyderabad, India in 2009 and M.Tech degree from R.E.C Warangal, Andhra Pradesh, India in 1988. She is currently working as Head and Professor of CSE, Vaagdevi Engineering College, Warangal, India. She has been awarded Ph.D for her research work done on Hill Cipher. Her main research interest includes Network security, Mobile Adhoc Networks and Artificial Intelligence. She has been involved in the organization as a chief member for various conferences and workshops. She published more than 50 research papers in National and International journals and conferences. She is presently supervising nearly 10 scholars for their research.