International Conference on Information Engineering, Management and Security
2015 [ICIEMS 2015]

ISBN	978-81-929742-7-9
Website	www.iciems.in
Received	10 - July - 2015
Article ID	ICIEMS006

VOL	01
eMail	iciems@asdf.res.in
Accepted	31- July - 2015
eAID	ICIEMS.2015.006

EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET

Siva Ranjani[1], Ranjani N[1], Sri Abarna[1], Ananthi K[1], Parkavvi[1]
[1]Thiagarajar College Of Engineering

INTRODUCTION

In MANET there is no fixed communication infrastructure. Each node is free to move in an arbitrary manner. Hence it is necessary for nodes to maintain updated position information with the immediate neighbor. Also there will be frequent changes in the topology of the mobile nodes in MANET. In geographic routing, the destination node and the node in the forwarding path can be mobile. In such case it is necessary to reduce the effects caused by the changing topology, which is a difficult task in geographic routing to reconstruct the network topology in presence of changing topology. To obtain the location of node's neighbor, each node exchanges its location information with its neighbor by periodic broadcasting of beacons. This periodic beaconing is not fair in terms of update cost collision of, packet delivery ratio and may lead to collision of data packet with beacon packet .To overcome this drawback, in this paper we propose an efficient beacon scheme called GPSR(Greedy Perimeter Stateless Routing Protocol) which dynamically adjust the frequency for beacon update based on nodes mobility. GPSR comprises two rules : The first rule, referred as Mobility Prediction (MP),which is used to significantly reduce the frequency of beacon overhead. The second rule, referred as On-Demand Learning (ODL), aims at improving the accuracy of local topology among the communicating nodes. Certain nodes considering their limited resources, mainly energy do not forward the data packet to its successive node although they are considered as active nodes in the neighbor list configuration. These nodes are identified as false nodes or selfish nodes and they are removed from the neighbor list and an alternate path is chosen to forward the packet. In this paper, we propose to reduce the beacon packet overhead and identify the false node in MANET.

2 LITERATURE SURVEY:

We gone through some of the literatures and acquired knowledge for choosing technique for efficient routing.

1)" False Node Detection Algorithm in Cluster Based MANET"- Mobile Ad hoc network are collection of mobile nodes that can dynamically form temporary networks, it is necessary to bring the smart technologies in the Ad hoc network environment. Huge amount of time and resources are wasted while travelling due to traffic congestion. The idea behind clustering is to group the network nodes into a number of overlapping clusters. In the clusters of MANET the resource constraints leads to a big problem as decrease in performance and the network partitioning leads to poor data accessibility due to false and selfish node. In our proposal the MANET area has been split into a number of size clusters having cluster head and storage capability according to connectivity degree, RSS (relative signal strength) as per the cluster formation algorithm given. In this cluster architecture they try to find false node inside clusters of MANET using a modified algorithm and try to remove them. Inside the cluster one node that manages the cluster activities is cluster head. Inside the cluster, there are ordinary nodes also that have direct access only to this one cluster head, and gateway. Gateways are nodes that can hear two or more cluster heads. Ordinary nodes send the packets to their cluster head that either distributes the packets inside the cluster, or (if the destination is outside the cluster) forwards them to a gateway node to be delivered

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

to the other clusters. Several nodes will be take part in the MANET for data forwarding and data packets transmission between source and destination.

They must forward the traffic which other nodes sent to it. Among all the nodes some nodes will behave selfishly, these nodes are called selfish nodes. In our paper we called selfish node as false node. Selfish nodes only to cooperate partially , or not at all, with other nodes. These selfish nodes could then reduce the overall data accessibility in the network. Selfish nodes use the network for their own communication, but simply decline to cooperate in forwarding packets for other nodes in order to save battery power. In the clusters of MANET the false nodes leads to a big problem as increase congestion. The idea behind splitting MANET into a number of size clusters having cluster head and storage capability as per the cluster formation algorithm given .But cluster formation is very difficult in MANET.

2)" Adaptive Position Update for Geographic Routing in Mobile Ad-hoc Networks"-In geographic routing, nodes need to maintain up-to-date positions of their immediate neighbors for making effective forwarding decisions. Periodic broadcasting of beacon packets that contain the geographic location coordinates of the nodes is a popular method used by most geographic routing protocols to maintain neighbor positions. We contend and demonstrate that periodic beaconing regardless of the node mobility and traffic patterns in the network is not attractive from both update cost and routing performance points of view. We propose the *A*daptive Position Update (APU) strategy for geographic routing, which dynamically adjusts the frequency of position updates based on the mobility dynamics of the nodes and the forwarding patterns in the network. APU is based on two simple principles: (i) nodes whose movements are harder to predict update their positions more frequently (and vice versa), and (ii) nodes closer to forwarding paths update their positions more frequently (and vice versa). Our theoretical analysis, which is validated by NS2 simulations of a well known geographic routing protocol, Greedy Perimeter Stateless Routing Protocol (GPSR), shows that APU can significantly reduce the update cost and improve the routing performance in terms of packet delivery ratio and average end-to-end delay in comparison with periodic beaconing and other recently proposed updating schemes. The benefits of APU are further confirmed by undertaking evaluations in realistic network scenarios, which account for localization error, realistic radio propagation and sparse network.

3) "EAACK-A Secure Intrusion Detection System for MANET"- The migration to wireless network from wired network has been a global trend in the past few decades. The open medium and wide distribution of nodes make MANET vulnerable to malicious attackers. A new technique EAACK (Enhanced Adaptive Acknowledgement) method designed for MANET was proposed for intrusion detection. EAACK demonstrates higher malicious-behavior-detection rates in certain circumstances while does not greatly affect the network performances. MANET is vulnerable to various types of attacks because of open infrastructure, dynamic network topology, lack of central administration and limited batterybased energy of mobile nodes. But most of these schemes become worthless when the malicious nodes already entered the network or some nodes in the network are compromised by attacker. Such attacks are more dangerous as these are initiated from inside the network. Routing protocols are generally necessary for maintaining effective communication between distinct nodes. Routing protocol not only discovers network topology but also built the route for forwarding data packets and dynamically maintains routes between any pair of communicating nodes. Routing protocols are designed to adapt frequent changes in the network due to mobility of nodes. MANET is capable of creating a self-configuring and self-maintaining network without the help of a centralized infrastructure, which is often infeasible in critical mission applications like military conflict or emergency recovery.

3.1 PROBLEM DEFINITION:

The problem with AODV(Ad-hoc Ondemand Distance Vector Routing) is that there is route setup latency when a new route is needed, because AODV queues data packets while discovering new routes and the queued packets are sent out only when new routes are found. This situation causes throughput loss in high mobility scenarios, because the packets get dropped quickly due to unstable route selection. Similarly,periodic beaconing used in AODV is not suitable for all nodes. Adaptive Position Strategy(APU) can be used to overcome this.

3.2 PROBLEM DESCRIPTION:

In geographic routing, nodes need to maintain up-to-date positions of their immediate neighbors for making effective forwarding decisions. Periodic broadcasting of beacon packets that contain the geographic location coordinates of the nodes is a popular method used by most geographic routing protocols to maintain neighbor positions. To demonstrate the periodic beaconing regardless of the node mobility and traffic patterns in the network is not attractive from both update cost and routing performance points of view. Adaptive Position Update (APU) strategy for geographic routing, which dynamically adjusts the frequency of position updates based on the mobility dynamics of the nodes and the forwarding patterns in the network. APU is based on two simple principles: (i) nodes whose movements are harder to predict update their positions more frequently (and vice versa), and (ii) nodes closer to forwarding paths update their positions more frequently (and vice versa). A poorly adjusted rate of beacon transmissions may lead to vast resource usage (power and bandwidth) on one side, or may lead to poor throughput on the other side. We use a general model without assuming a particular mobility model. The model is instantiated for periodic and exponential beaconing and it is then applied to compare two-way beaconing with one-way beaconing. The disadvantage of this protocol is it is not scalable in large networks and it does not support asymmetric links. Periodic beaconing consumes network bandwidth,increase update cost,end-to-end delay. Thus Packet delivery ratio will get decreased.Beacon packets traffic will be overhead for data packets and most of the data packets will be dropped. Average end-to-end delay is more in periodic beaconing,because neighbor list is updated periodically not based on mobility of nodes.False nodes in the routing path affects routing performance. These nodes do not forward data packets in order to save their energy. Alternate path for forwarding should be chosen. The unreachability of even a small fraction of destinations on static networks because of the failure of the no-crossing heuristic is also problematic; such routing failures are permanent, not transitory. The power of greedy forwarding to route using only neighbor nodes' positions comes with one attendant drawback: there are topologies in which the only route to a destination requires a packet move temporarily farther in geometric distance from the destination. In Distance Source Vector (DSV) routing, by caching the negative information, the link may get broken, this cause the problem in the system. Source routes in use may be automatically shortened if one or more intermediate hops in the route become no longer necessary.

4.1 GPSR PROTOCOL:

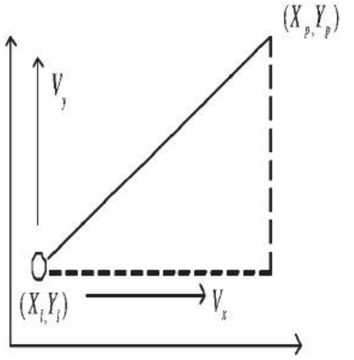
Greedy Perimeter Stateless Routing (GPSR), a novel routing protocol for wireless datagram networks that uses the positions of routers and a packet's destination to make packet forwarding decisions. Geographic routing is also called georouting or position based routing which is a routing principle that relies on geographic position information. It is mainly proposed for wireless networks and based on the idea that the source sends a message to the geographic location of the destination instead of using the network address. The idea of using position information in the area of packet radio networks and interconnection networks. Geographic routing requires that each node can determine its own location and that the source is aware of the location of the destination. With this information a message can be routed to the destination without knowledge of the network topology or a prior route discovery. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. GPSR scales better in per router state than shortest path and adhoc routing protocols as the number of network destinations increases. GPSR can use local topology information to find correct new routes quickly. However, in situations where nodes are mobiles or when nodes often switch off or on ,the local topology rarely remain static. Hence, its necessary that each node broadcasts its updated location information to all of its neighbors.These location updated packets are usually referred as a beacons.Greedy perimeter stateless routing protocol shows that APU can significantly reduce the update cost and improve the routing performance in terms of packet delivery ratio and average end to end delay in comparison with periodic beaconing and other recently proposed updating schemes.GPSR protocol use extensive simulation of mobile wireless networks to compare its performance with Dynamic Source Routing. In networks of wireless stations, communication between source and destination nodes may require traversal of multiple hops, as radio ranges are finite. A community of adhoc network researchers has proposed, implemented, and measured a variety of routing algorithms for such networks. The observation that topology changes more rapidly on a mobile, wireless network than on wired networks, where Link State Protocol is used. In a linkstate protocol, the only information passed between the nodes is information used to construct the connectivity maps. GPSR benefits from geographic routing use of only immediateneighbor information in forwarding decision. GPSR allows nodes to figure out who its closest neighbors are also close to the destination the information is supposed to travel.

4.2 MOBILITY PREDICTION RULE :

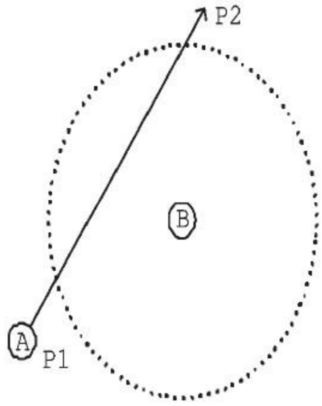
A Mobile Ad hoc NETwork (MANET) is a collection of wireless mobile nodes forming a network without using any existing infrastructure. All mobile nodes function as mobile routers that discover and maintain routes to other mobile nodes of the network and therefore, can be connected dynamically in an arbitrary manner. The mobility attribute of MANETs is a very significant one.

The mobile nodes may follow different mobility patterns that may affect connectivity, and in turn protocol mechanisms and performance. Mobility prediction may positively affect the service-oriented aspects as well as the application-oriented aspects of ad hoc networking. At the network level, accurate node mobility prediction may be critical to tasks such as call admission control, reservation of network resources, pre-configuration of services and QoS provisioning. At the application level, user mobility prediction in combination with user's profile may provide the user with enhanced location-based wireless services, such as route guidance, local traffic information and on-line advertising. In this chapter we present the most important mobility prediction schemes for MANETs in the literature, focusing on their main design principles and characteristics. This rule adapts the beacon generation rate to the frequency with which the nodes change the characteristics that govern their motion (velocity and heading).The motion characteristics are included in the beacons broadcast to a node's neighbors. The neighbors can then track the node's motion using simple linear motion equations. Nodes that frequently change their motion need to frequently update their neighbors, since their locations are changing dynamically. On the contrary, nodes which move slowly do not need to send frequent updates. A periodic beacon update policy cannot satisfy both these requirements simultaneously, since a small update interval will be wasteful for slow nodes, whereas a larger update interval will lead to inaccurate position information for the highly mobile nodes. The MP rule, thus, tries to maximize the effective duration of each beacon, by broadcasting a beacon only when the predicted position information based on the previous beacon becomes inaccurate. This extends the effective duration of the beacon for nodes with low mobility, thus reducing the number of beacons. Further,highly mobile nodes can broadcast frequent beacons to ensure that their neighbors are aware of the rapidly changing topology.

This rule adapts the beacon generation rate to the mobility of nodes. Nodes which contains highly mobile need to frequently update their neighbors since their locations are changing dynamically. At the same time, nodes which move slowly do not need to send frequent updates. This MP rule adapts the beacon generation rate to the frequency with which the nodes change the characteristics that govern their motion (velocity). The motion characteristics are included in the beacons broadcast to a node's neighbors. The neighbors can then track the node's motion using simple linear motion equations. Nodes that frequently change their motion need to frequently update their neighbors, since their locations are changing dynamically. Nodes which move slowly do not need to send frequent updates. A periodic beacon update policy cannot satisfy both these requirements simultaneously, since a small update interval will be wasteful for slow nodes, whereas a larger update interval will lead to inaccurate position information for the highly mobile nodes.



4.2.1 Example of MP Rule



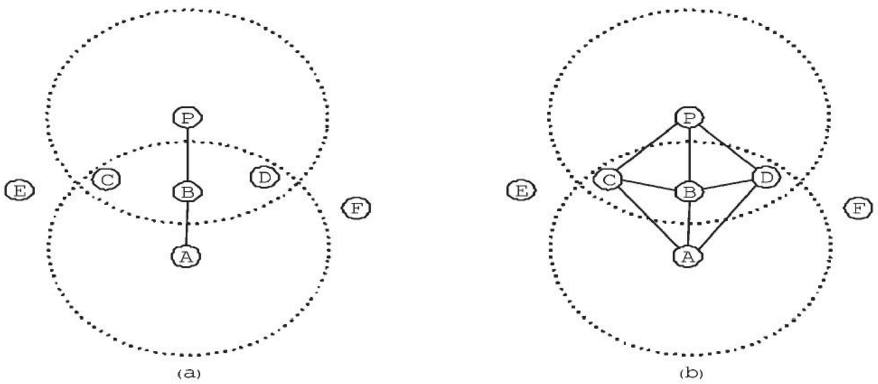
4.2.2 Drawback of MP Rule

Variables	Definition
(Xl,Yl)	The coordinate of node i at time Tl (included in the previous beacon)
(Vx,Vy)	The velocity of node i along the direction of the x and y axes at time Tl (included in the previous beacon)
Tl,Tc	The time of the last beacon broadcast and current time
(Xp,Yp)	The predicted position of node i at the current time

4.3 ON DEMAND LEARNING RULE:

A node broadcasts beacons response to data forwarding activities that occur in the vicinity of that node. Whenever a node overhears a data transmission from a new neighbor, it broadcasts a beacon as a response, it implies a neighbour who is not contained in the neighbor list of this node. A node waits for a small random time interval before responding with the beacon to prevent collisions with other beacons. The location updates are piggybacked on the data packets and that all nodes operate in the promiscuous mode, which allows them to overhear all data packets transmitted in their vicinity. Since the data packet contains the location of the final destination, any node that overhears a data packet also checks its current location and determines if the destination is within its transmission range. According to this rule,whenever a node overhears a data transmission from a new neighbor, it broadcasts a beacon as a response. By a new neighbor, we imply a neighbor who is not contained in the neighbor list of this node. In reality, a node waits for a small random time interval before responding with the beacon to prevent collisions with other beacons.Recall that, we have assumed that the location updates are piggybacked on the data packets and that all nodes operate in the promiscuous mode, which allows them to overhear all data packets transmitte in their vicinity.In addition, since the data packet contains the location of the final destination, any node that overhears a data packet also checks its current location and determines if the destination is within its transmission range. If so,the destination node is added to the list of neighboring nodes, if it is not already present. Note that, this particular check incurs zero cost, i.e. no beacons need to be transmitted. The MP rule solely may not be sufficient for maintaining an accurate local topology. In the worstcase, assuming no other nodes were in the nearby range, the data packets would not be transmitted at all here To maintain a more accurate local topology devise a mechanism in those regions of the network . This is precisely On-Demand Learning (ODL) rule aims to achieves this. As the name suggests, a node broadcasts beacons packet on-demand, i.e. in response to data forwarding node that occur in activities involve the vicinity of that node According to this rule, whenever a node overhears a data transmission from a new neighbor, it broadcasts a beacon as a response. Node waits for a small random time interval before responding with the beacon to prevent collisions with other beacon .In addition, since the data packet contains the location of the final destination, any node that overhears data packet also checks its current location and determines if the destination is within its transmission range. If so, the destination node is added to the list of nodes neighbor if it is not added. Note that, this particular check incurs turns to zero cost, i.e., no beacons need to be transmitted.

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

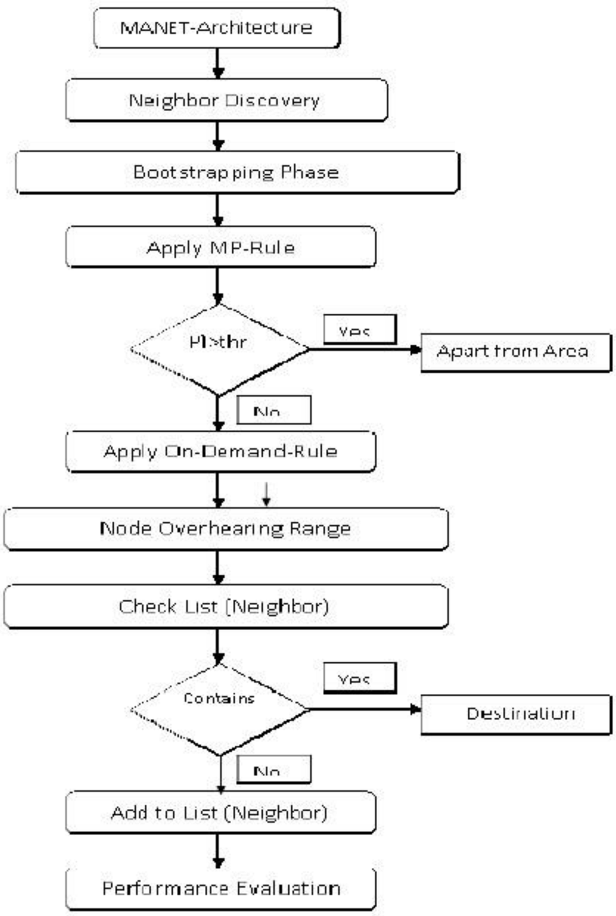


4.3.1 Example for ODL rule

4.4 FALSE NODE DETECTION:

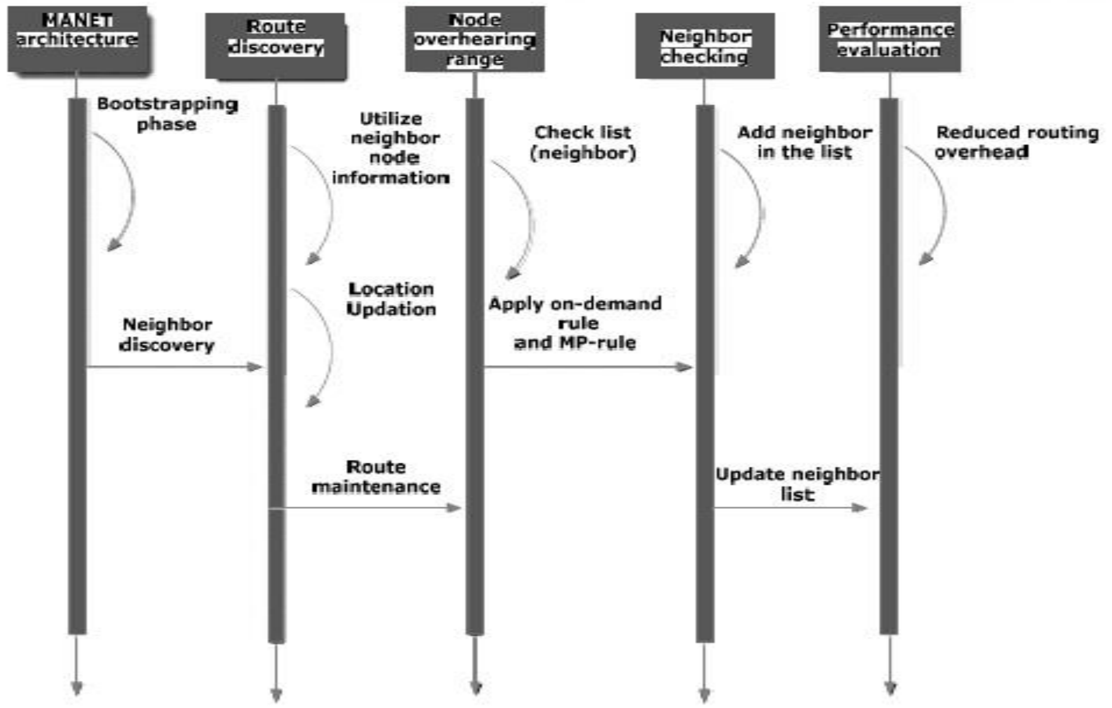
The nodes participating in the packet forwarding should co-operate, if these nodes are not forwarding the packets to the destination then these nodes are considered as the selfish nodes. These selfish nodes detection is an important factor in the network performance. The detected selfish nodes are avoided from the routing path to avoid the lost of the packets. The amount of packets can be saved from these selfish nodes and thus can enhance the network performance through the detection of these nodes. Selfish nodes are inclined to get the greatest profits from the networks and at the same time these nodes trying to conserve their own resources like bandwidth, batterylife or hardware. A selfish node only communicates to other nodes if its data packet is required to send to some other node and refuses to cooperate other nodes whenever it some data packets or routing packets are received by it that it has no interest in. Hence data packets are either refused to retransmit or are dropped for being received by a selfish node. The nodes which don't send RREQ packets don't impact the network, this sort of selfish nodes can increase end to end delay because the number of nodes in the transmission path will increase. If a hello message is not accepted from a neighbour inside two seconds of the last message, connectivity lost is determined to that neighbor node.

5.1 FLOWCHART:



Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

5.2 SEQUENCE DIAGRAM:



5.3 SYSTEM CONFIGURATION

5.3.1 HARDWARE CONFIGURATION:

Processor : Intel Pentium dual core

RAM : 2 GB

Clock speed : 1.6 GHz

Hard disk : 40 GB

5.3.2 SOFTWARE CONFIGURATION:

Operating System : Windows XP /Red Hat Linux 9.0

Tools : NS2

Languages : TCL/Tk,awk,GCC

6.1 ADAPTIVE POSITION UPDATE:

In this paper, we propose a novel beaconing strategy for geographic routing protocols called Adaptive Position Updates strategy (APU). Our scheme eliminates the drawbacks of periodic beaconing by adapting to the system variations. APU incorporates two rules for triggering the beacon update process. The first rule uses a simple mobility prediction scheme to estimate when the location information broadcast in the previous beacon becomes inaccurate. The next beacon is broadcast only if the predicted error in the location estimate is greater than a certain threshold, thus tuning the update frequency to the mobility of the nodes. The second rule proposes an on-demand learning strategy, whereby beacons are exchanged in response to data packets from new neighbors in a node's vicinity. This ensures that nodes involved in forwarding data packets maintain a fresh view of the local topology. On the contrary, nodes that are not in the vicinity of the forwarding path are unaffected by this rule and do not broadcast beacons. By reducing the beacon updates, APU reduces the power and bandwidth utilization, resources which are scarce in MANETs. It also decreases the chance of link-layer collisions with the data packets and consequently reduces the end-to-end delay. Note that, APU simply governs the beacon update strategy and is hence compatible with any geographic routing protocol. In this work, we have incorporated the APU strategy within GPSR (Greedy Perimeter Stateless Routing) [2] as a representative example. We have carried out simulations to evaluate the performance improvement achieved by APU with randomly generated network topologies and mobility patterns. We have also performed some initial experiments with realistic movement patterns of buses in a metropolitan city. Our initial results indicate that APU significantly reduces beacon overhead without having any noticeable impact on the data delivery rate.

6.2 GPSR(GREEDY PERIMETER STATELESS ROUTING PROTOCOL):

Greedy Perimeter Stateless Routing, GPSR, is a responsive and efficient algorithms before it, which use graph-theoretic notions of shortest paths and transitive reachability to find routes, GPSR exploits the correspondence between node and connectivity in a wireless network, by using the positions of nodes to make packet forwarding decisions. In this paper,we aim at reducing the beacon overhead.In case of MANET Upon initialization, each node broadcasts a beacon informing its neighbors about its presence and its current location and velocity. Following this, in most geographic routing protocols such as GPSR, each node periodically broadcasts its current location information. The position information received from neighboring beacons is stored at each node. Based on the position updates received from its neighbors, each node continuously updates its local topology, which is represented as a neighbor list. Only those nodes from the neighbor list are considered as possible candidates for data forwarding. Thus, the beacons play an important part in maintaining an accurate representation of the local topology.GPSR uses greedy forwarding to forward packets to nodes that are always progressively closer to the destination. In regions of the network where such a greedy path does not exist (*i.e.,* the only path requires that one move temporarily farther away from the destination), GPSR recovers by forwarding in *perimeter mode*, in which a packet traverses successively closer *faces* of a planar sub graph of the full radio network connectivity graph, until reaching a node closer to the destination, where greedy forwarding resumes. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. By keeping state only about the local topology, GPSR scales better in per-router state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobility's frequent topology changes, GPSR can use local topology information to find correct new routes quickly.

Greedy Forwarding: As mentioned in the introduction, under GPSR, packets are marked by their originator with their destinations' locations. As a result, a forwarding node can make a locally optimal, greedy choice in choosing a packet's next hop. Specifically, if a node knows its radio neighbors' positions, the locally optimal choice of next hop is the neighbor geographically closest to the packet's destination. Forwarding in this regime follows successively closer geographic hops, until the destination is reached. An example of greedy nexthop choice appears in Figure 1. Here, x receives a packet destined for D. x's radio range is denoted by the dotted circle about x, and the arc with radius equal to the distance between y and D is shown as the dashed arc about D. x forwards the packet to y, as the distance between y and D is less than that between D and any of x's other neighbors. This greedy forwarding process repeats, until the packet reaches D. A simple beaconing algorithm provides all nodes with their neighbors' positions: periodically, each node transmits a beacon to the broadcast MAC address, containing only its own identifier (e.g., IP address) and position. We encode position as two four-byte floatingpoint quantities, for x and y coordinate values. To avoid synchronization of neighbors' beacons, as observed by Floyd and Jacobson , we jitter each beacon's transmission by 50% of the interval B between beacons, such that the mean inter-beacon transmission interval is B, uniformly distributed in [0:5B; 1:5B]. Upon not receiving a beacon from a neighbor for longer than timeout interval T, a GPSR router assumes that the neighbor has failed or gone out-ofrange, and deletes the neighbor from its table. The 802.11 MAC layer also gives direct indications of link-level retransmission failures to neighbors; we interpret these indications identically. We have used T = 4:5B, three times the maximum jittered beacon interval, in this work. Greedy forwarding's great advantage is its reliance only on knowledge of the forwarding node's immediate neighbors. The state required is negligible, and dependent on the density of nodes in the wireless network, not the total number of destinations in the network.1 On networks where multi-hop routing is useful, the number of neighbors within a node's radio range must be substantially less than the total number of nodes in the network. The position a node associates with a neighbor becomes less current between beacons as that neighbor moves. The accuracy of the set of neighbors also decreases; old neighbors may leave and new neighbors may enter radio range. For these reasons, the correct choice of beaconing interval to keep nodes' neighbor tables current depends on the rate of mobility in the network and range of nodes' radios. We show the effect of this interval on GPSR's performance in our simulation results. We note that keeping current topological state for a one-hop radius about a router is the minimum required to do any routing; no useful forwarding decision can be made without knowledge of the topology one or more hops away. This beaconing mechanism does represent pro-active routing protocol traffic, avoided by DSR and AODV. To minimize the cost of beaconing, GPSR piggybacks the local sending node's position on all data packets it forwards, and runs all nodes' network interfaces in promiscuous mode, so that each station receives a copy of all packets for all stations within radio range. At a small cost in bytes (twelve bytes per packet), this scheme allows all packets to serve as beacons. When any node sends a data packet, it can then reset its inter-beacon timer. This optimization reduces

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

beacon traffic in regions of the network actively forwarding data packets. In fact, we could make GPSR's beacon mechanism fully reactive by having nodes solicit beacons with a broadcast "neighbor request" only when they have data traffic to forward. We have not felt it necessary to take this step, however, as the one-hop beacon overhead does not congest our simulated networks. The power of greedy forwarding to route using only neighbor nodes' positions comes with one attendant drawback: there are topologies in which the only route to a destination requires a packet move temporarily farther in geometric distance from the destination . A simple example of such a topology is shown in . Here, x is closer to D than its neighbors w and y. Again, the dashed arc about D has a radius equal to the distance between x and D. Although two paths, (x ! y ! z ! D) and (x ! w ! v ! D), exist to D, x will not choose to forward to w or y using greedy forwarding. x is a local maximum in its proximity to D. Some other mechanism must be used to forward packets in these situations.

6.3 MOBILITY PREDICTION RULE:

To avoid periodic beaconing in the routing strategy,APU adapts the beacon update intervals to the mobility dynamics of the nodes and the amount of data being forwarded in the neighborhood of the nodes. To achieve this APU employs MP rule. The beacons transmitted by the nodes contain their current position and speed. Nodes estimate their positions periodically by employing linear kinematic equations based on the parameters announced in the last announced beacon. If the predicted location is different from the actual location, a new beacon is broadcast to inform the neighbors about changes in the node's mobility characteristics The Mobility Prediction rule is triggered when there is change in the location of the node. The change in the location of the node cannot be predicated feasibly because the nodes move in the random fashion. This rule adapts the beacon generation rate to the frequency with which the nodes change the characteristics that govern their motion (velocity and heading). The motion characteristics are included in the beacons broadcast to a node's neighbors. The neighbors can then track the node's motion using simple linear motion equations.

Nodes that frequently change their motion need to frequently update their neighbors, since their locations are changing dynamically. On the contrary, nodes which move slowly do not need to send frequent updates. A periodic beacon update policy cannot satisfy both these requirements simultaneously, since a small update interval will be wasteful for slow nodes, whereas a larger update interval will lead to inaccurate position information for the highly mobile nodes. In our scheme, upon receiving a beacon update from a node *i*, each of its neighbors records node *i'*s current position and velocity and periodically track node *i's* location using a simple prediction scheme based on linear kinematics (discussed below). Based on this position estimate, the neighbors can check whether node *i* is still within their transmission range and update their neighbor list accordingly. The goal of the MP rule is to send the next beacon update from node *i* when the error between the predicted location in the neighbors of *i* and node *i's* actual location is greater than an acceptable threshold.The neighbours estimate the current position of node I by using linear kinematics equation.On the contrary node *i* uses the same prediction scheme to keep track of its predicted location among its neighbors.Node *i* then computes the deviation with this information.If the deviation is greater than a certain threshold, known as the Acceptable Error Range (AER), it acts as a trigger for node *i* to broadcast its current location and velocity as a new beacon. The MP rule, thus, tries to maximize the effective duration of each beacon, by broadcasting a beacon only when the predicted position information based on the previous beacon becomes inaccurate. This extends the effective duration of the beacon for nodes with low mobility, thus reducing the number of beacons. Further, highly mobile nodes can broadcast frequent beacons to ensure that their neighbors are aware of the rapidly changing topology. In this method, the mobility prediction rule (MP rule) helps in reducing the amount of beacon packets transmitted in the MANET. Mobility prediction rule help in reducing the traffic of beacon overhead and enabling the increase of packet delivery ratio. Mobility Prediction rule also helps in reducing update cost,bandwidth,end-to-end delay. Mobility Prediction (MP) uses a simple mobility prediction scheme to estimate when the location information broadcast in the previous beacon becomes inaccurate. The next beacon is broadcast only if the predicted error in the location estimate is greater than a certain threshold, thus tuning the update frequency to the dynamism inherent in the node's motion.

6.4 ON-Demand Learning Rule :

The MP rule solely may not be sufficient for maintaining an accurate local topology. Consider the example illustrated in which node A moves from P1 to P2 at a constant velocity. Now, assume that node A has just sent a beacon while at P1. Since node B did not receive this packet, it is unaware of the existence of node A. Further, assume that the AER is sufficiently large such that when node A moves from P1 to P2, the MP rule is never triggered. However, node A is within

the communication range of B for a significant portion of its motion. Even then, neither A nor B will be aware of each other. Now, in situations where neither of these nodes are transmitting data packets, this is perfectly fine since they are not within communicating range once A reaches P2. However, if either A or B was transmitting data packets, then their local topology will not be updated and they will exclude each other while selecting the next hop node. In the worst case,

assuming no other nodes were in the vicinity, the data packets would not be transmitted at all. Hence, it is necessary to devise a mechanism, which will maintain a more accurate local topology in those regions of the network where significant data forwarding activities are on-going. This is precisely what the On- Demand Learning rule aims to achieve. As the name suggests, a node broadcasts beacons on-demand, i.e., in response to data forwarding activities that occur in the vicinity of that node. According to this rule, whenever a node overhears a data transmission from a new neighbor, it broadcasts a beacon as a response. By a new neighbor, we imply a neighbor who is not contained in the neighbor list of this node. In reality, a node waits for a small random time interval before responding with the beacon to prevent collisions with other beacons. Recall that, we have assumed that the location updates are piggybacked on the data packets and that all nodes operate in the promiscuous mode, which allows them to overhear all data packets transmitted in their vicinity. In addition, since the data packet contains the location of the final destination, any node that overhears a data packet also checks its current location and determines if the destination is within its transmission range. If so, the destination node is added to the list of neighboring nodes, if it is not already present. Note that, this particular check incurs zero cost, i.e., no beacons need to be transmitted. We refer to the neighbor list developed at a node by virtue of the initialization phase and the MP rule as the basic list. This list is mainly updated in response to the mobility of the node and its neighbors. The ODL rule allows active nodes that are involved in data forwarding to enrich their local topology beyond this basic set. In other words, a rich neighbor list is maintained at the nodes located in the regions of high traffic load. Thus, the rich list is maintained only at the active nodes and is built reactively in response to the network traffic. All inactive nodes simply maintain the basic neighbor list. By maintaining a rich neighbor list along the forwarding path, ODL ensures that in situations where the nodes involved in data forwarding are highly mobile, alternate routes can be easily established without incurring additional delays. ODL diagram illustrates the network topology before node A starts sending data to node P. The solid lines in the figure denote that both ends of the link are aware of each other.

6.5 PERFOMANCE EVALUATION:

6.5.1 PACKET DELIVERY RATIO:

The ratio of the number of delivered data packet to the destination. This illustrates the level of delivered data to the destination. The greater value of packet delivery ratio means the better performance of the protocol.

PDR=Σ Number of packet receive / Σ Number of packet send

6.5.2 END-TO-END DELAY:

The average time taken by a data packet to arrive in the destination. It also includes the delay caused by route discovery process and the queue in data packet transmission. Only the data packets that successfully delivered to destinations that counted. The lower value of end to end delay means the better performance of the protocol.

End-to-End delay=Σ (arrive time – send time) / Σ Number of connections

6.5.3 PACKET LOSS:

The total number of packets dropped during the simulation. The lower value of the packet lost means the better performance of the protocol.

Packet lost = Number of packet send – Number of packet received .

IMPLEMENTATION:

WIRELESS-GPSR.TCL:

```
set opt(chan) Channel/WirelessChannel
set opt(prop) Propagation/TwoRayGround
set opt(netif) Phy/WirelessPhy
set opt(mac) Mac/802_11
set opt(ifq) Queue/DropTail/PriQueue ;# for dsdv
set opt(ll) LL
set opt(ant) Antenna/OmniAntenna
set opt(x) 670 ;# X dimension of the topography
set opt(y) 670 ;# Y dimension of the topography
set opt(cp) "./cbr100.tcl"
set opt(sc) "./grid-deploy10x10.tcl"
set opt(ifqlen) 50 ;# max packet in ifq
set opt(nn) 100 ;# number of nodes
set opt(seed) 0.0
set opt(stop) 250.0 ;# simulation time
set opt(tr) trace.tr ;# trace file
set opt(nam) out.nam
set opt(rp) gpsr ;# routing protocol script (dsr
or dsdv)
set opt(lm) "off" ;# log movement
LL set mindelay_ 50us
LL set delay_ 25us
LL set bandwidth_ 0 ;# not used
Agent/Null set sport_ 0
Agent/Null set dport_ 0
Agent/CBR set sport_ 0
```

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

```
Agent/CBR set dport_ 0
Agent/TCPSink set sport_ 0
Agent/TCPSink set dport_ 0
Agent/TCP set sport_ 0
Agent/TCP set dport_ 0
Agent/TCP set packetSize_ 1460
Queue/DropTail/PriQueue set Prefer_Routing_Protocols 1
# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1.5 meters above
it
Antenna/OmniAntenna set X_ 0
28
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 1.5
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0
# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_ 10.0
Phy/WirelessPhy set CSThresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0
# The transimssion radio range
#Phy/WirelessPhy set Pt_ 6.9872e-4 ;# ?m
Phy/WirelessPhy set Pt_ 8.5872e-4 ;# 40m
#Phy/WirelessPhy set Pt_ 1.33826e-3 ;# 50m
#Phy/WirelessPhy set Pt_ 7.214e-3 ;# 100m
#Phy/WirelessPhy set Pt_ 0.2818 ;# 250m
#
proc usage { argv0 } {
puts "Usage: $argv0"
puts "\tmandatory arguments:"
puts "\t\t\[-x MAXX\] \[-y MAXY\]"
puts "\toptional arguments:"
puts "\t\t\[-cp conn pattern\] \[-sc scenario\] \[-nn nodes\]"
puts "\t\t\[-seed seed\] \[-stop sec\] \[-tr tracefile\]\n"
}
proc getopt {argc argv} {
global opt
lappend optlist cp nn seed sc stop tr x y
for {set i 0} {$i < $argc} {incr i} {
set arg [lindex $argv $i]
if {[string range $arg 0 0] != "-"} continue
set name [string range $arg 1 end]
set opt($name) [lindex $argv [expr $i+1]]
}
}
proc log-movement {} {
global logtimer ns_ ns
set ns $ns_
# source ../tcl/mobility/timer.tcl
Class LogTimer -superclass Timer
29
LogTimer instproc timeout {} {
global opt node_;
for {set i 0} {$i < $opt(nn)} {incr i} {
$node_($i) log-movement
}
```

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

```
$self sched 0.1
}
set logtimer [new LogTimer]
$logtimer sched 0.1
}
getopt $argc $argv
if { $opt(x) == 0 || $opt(y) == 0 } {
usage $argv0
exit 1
}
if {$opt(seed) > 0} {
puts "Seeding Random number generator with $opt(seed)\n"
ns-random $opt(seed)
}
#
# Initialize Global Variables
#
set ns_ [new Simulator]
set chan [new $opt(chan)]
set prop [new $opt(prop)]
set topo [new Topography]
set tracefd [open $opt(tr) w]
$ns_ trace-all $tracefd
#set namfile [open $opt(nam) w]
#$ns_ namtrace-all $namfile
#modified
set namfile [open $opt(nam) w]
$ns_ namtrace-all-wireless $namfile $opt(x) $opt(y)
$topo load_flatgrid $opt(x) $opt(y)
$prop topography $topo
#
# Create God
#
set god_ [create-god $opt(nn)]
30
$ns_ node-config -adhocRouting gpsr \
-llType $opt(ll) \
-macType $opt(mac) \
-ifqType $opt(ifq) \
-ifqLen $opt(ifqlen) \
-antType $opt(ant) \
-propType $opt(prop) \
-phyType $opt(netif) \
-channelType $opt(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace OFF
source ./gpsr.tcl
for {set i 0} {$i < $opt(nn) } {incr i} {
gpsr-create-mobile-node $i
$node_($i) namattach $namfile
$ns_ at 0.0 "$node_($i) setdest [ expr { rand() * 670 } ] [ expr {
rand() * 670 } ] 10.0"
}
#
# Source the Connection and Movement scripts
#
if { $opt(cp) == "" } {
```

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

```
puts "*** NOTE: no connection pattern specified."
set opt(cp) "none"
} else {
puts "Loading connection pattern..."
$ns_ at 10.0 "$ns_ trace-annotate \"Loadin connection pattern
...........\""
source $opt(cp)
}
#
# Tell all the nodes when the simulation ends
#
for {set i 0} {$i < $opt(nn) } {incr i} {
$ns_ at $opt(stop).000000001 "$node_($i) reset";
}
$ns_ at $opt(stop).00000001 "puts \"NS EXITING...\" ; $ns_ halt"
if { $opt(sc) == "" } {
31
puts "*** NOTE: no scenario file specified."
set opt(sc) "none"
} else {
puts "Loading scenario file..."
$ns_ at 0.1 "$ns_ trace-annotate \"Loading Scenario
File...........\""
source $opt(sc)
puts "Load complete..."
$ns_ at 0.15 "$ns_ trace-annotate \"Load complete...........\""
}
#added by zhou
for {set i 0} {$i < $opt(nn)} {incr i} {
$ns_ initial_node_pos $node_($i) 10
}
##
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp $opt(rp)"
puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"
puts "Starting Simulation..."
proc finish {} {
global ns_ tracefd namfile
$ns_ flush-trace
close $tracefd
close $namfile
exec nam out.nam &
exit 0
}
$ns_ at $opt(stop) "finish"
$ns_ run
DATASHORT.PL:
#!/usr/bin/perl
$ofile="simresult.txt";
$nNodes=10;
$inEnergy=100;
open OUT, ">$ofile" or die "$0 cannot open output file $ofile: $!";
print "Please Stand By. Analyzing File: simple.tr in "; print `pwd`;
print "\n";
open OUT, ">$ofile" or die "$0 cannot open output file $ofile: $!";
print OUT "=================== Simulation Result
============================\n";
print OUT " Date:"; print OUT `date`;
print OUT "\n Analyzed File: simple.tr in "; print OUT `pwd`;
32
```

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

```
printQQ22\A\\QQ2 OUT
"\n=========================================================================\
n";
while(<>){
@mline = split(':', $_);
@mline2 = split('\[', $mline[0]);
@word = split('\]',$mline2[2]);
@eng = split(" ",$word[0]);
@tline = split('_', $_);
$src=$tline[1];
$Emin[$src] = $eng[1];
}
for ($i=0;$i < $nNodes; $i++) {
# print "Node($i) : $Emin[$i]\n";
# print OUT "Node($i) : $Emin[$i]\n";
$total = $total + $Emin[$i];
}
$consume=($total/($nNodes*$inEnergy))*100;
$average=$total/$nNodes;
for ($i=0;$i<$nNodes;$i++) {
$sub = $average - $Emin[$i];
if($sub < 0) {
$sub = $sub * -1;
}
$sub_total = $sub_total + $sub;
}
$pyuncha=$sub_total/$nNodes;
#Primary Information
print OUT " Total Remained Energy : $total\n";
print OUT " Average Remained Energy : $average\n";
print OUT " Energy Difference : $pyuncha\n";
close OUT;
```

SCREEN SHOTS:

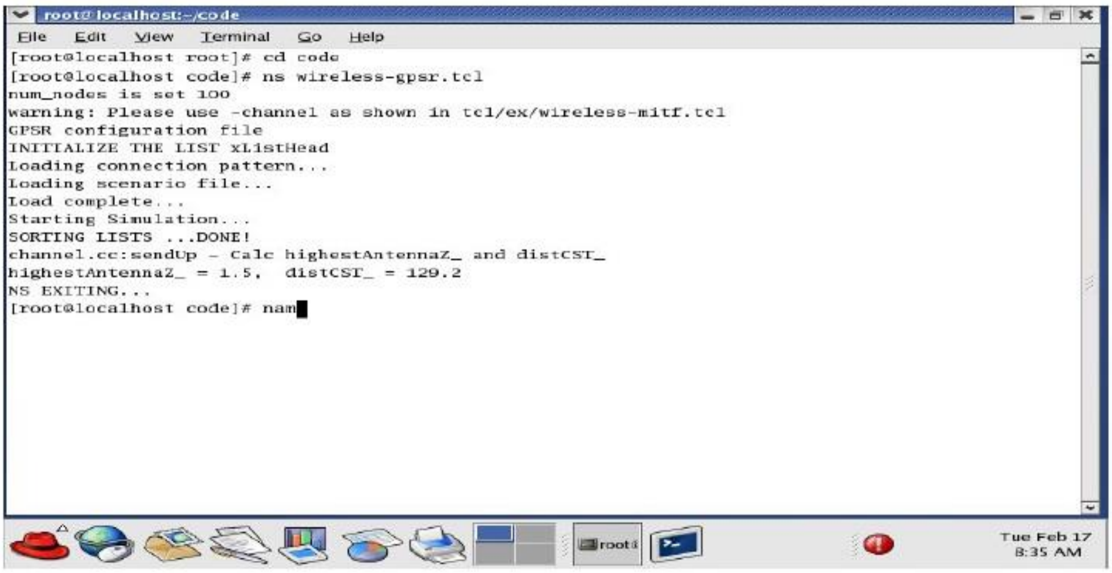


FIG 8.1 RUNNING OF PROGRAM

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

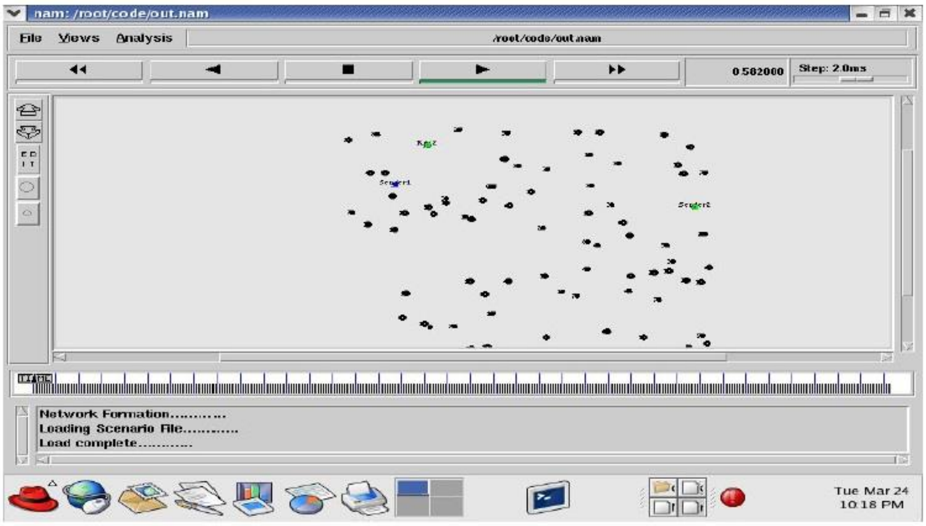


FIG 8.2 APPLYING MP AND ODL RULE

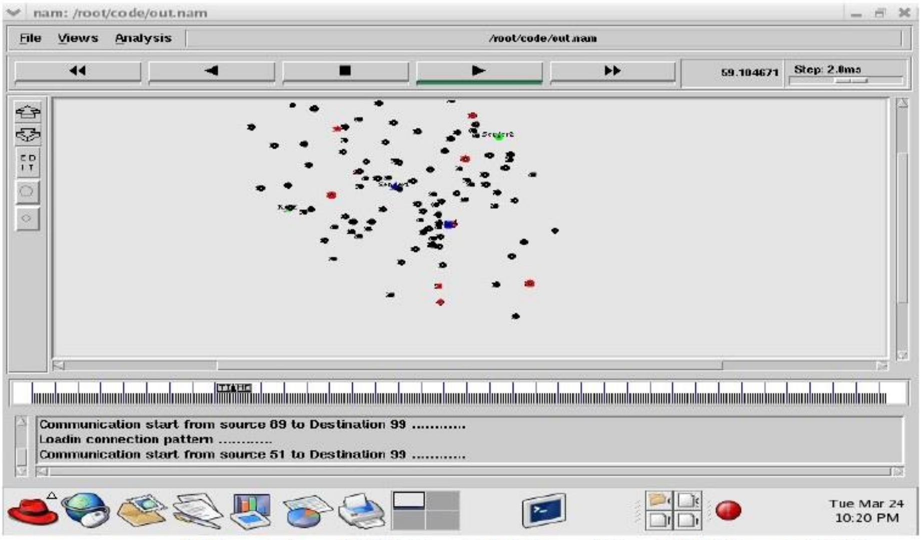
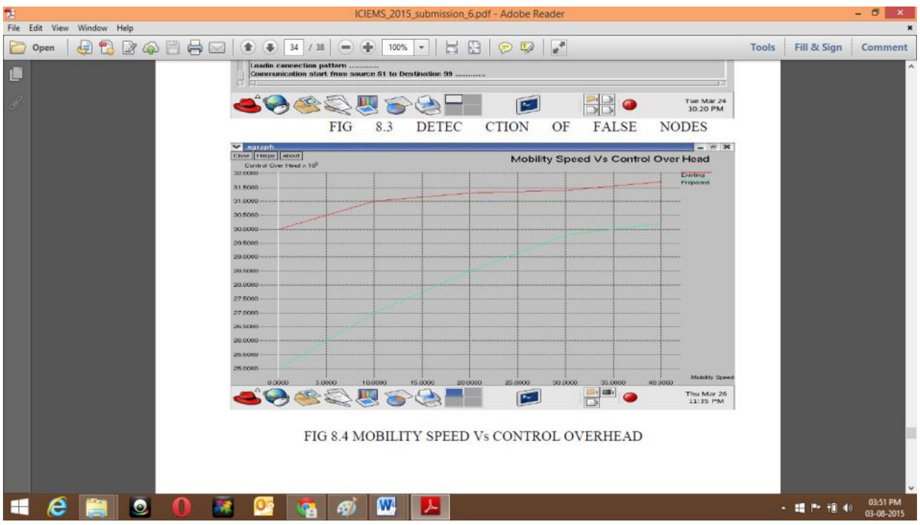


FIG 8.3 DETEC CTION OF FALSE NODES



Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

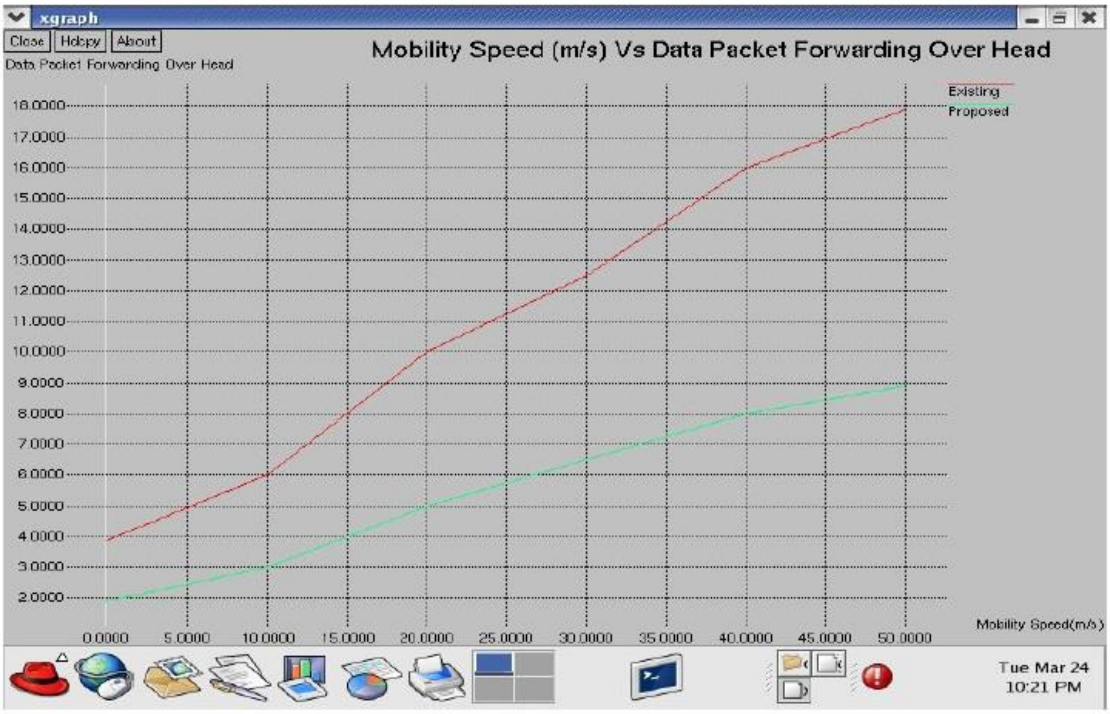


FIG 8.4 MOBILITY SPEED Vs DATA PACKET OVERHEAD

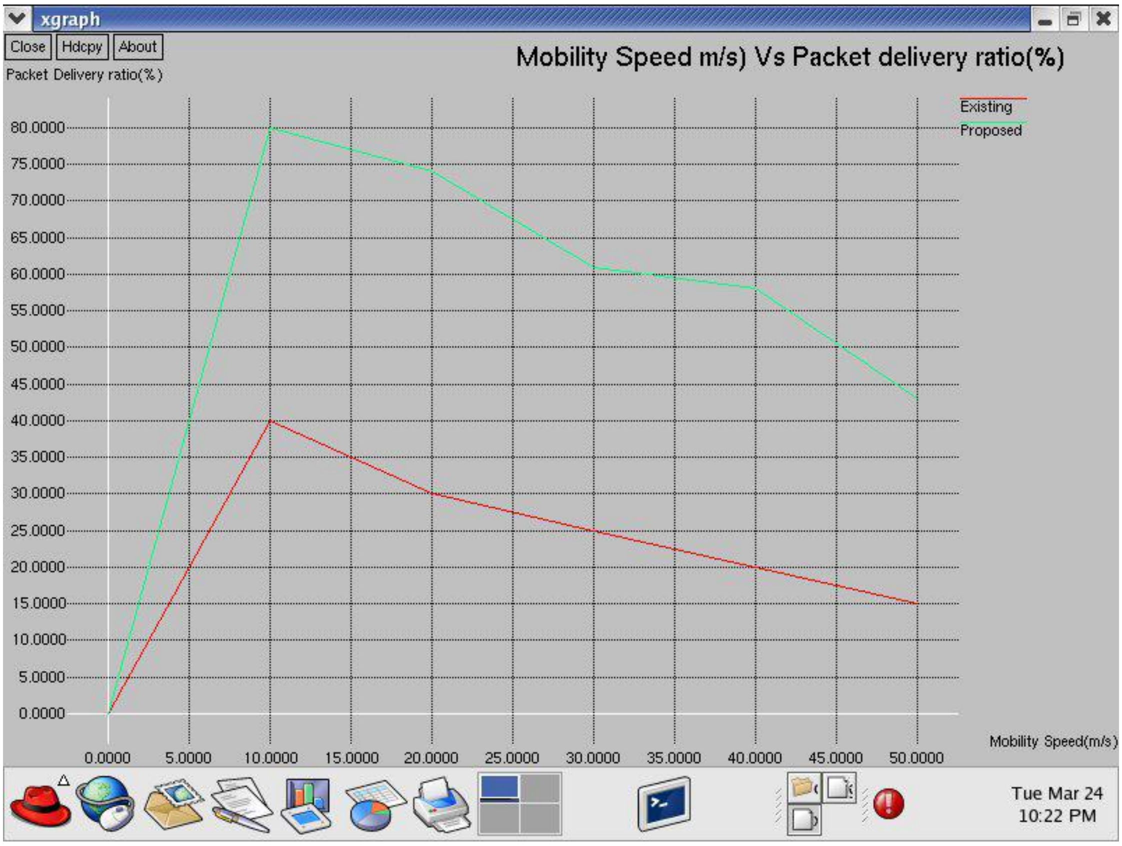


FIG 8.4 MOBILITY SPEED Vs PACKET DELIVERY RATIO

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.

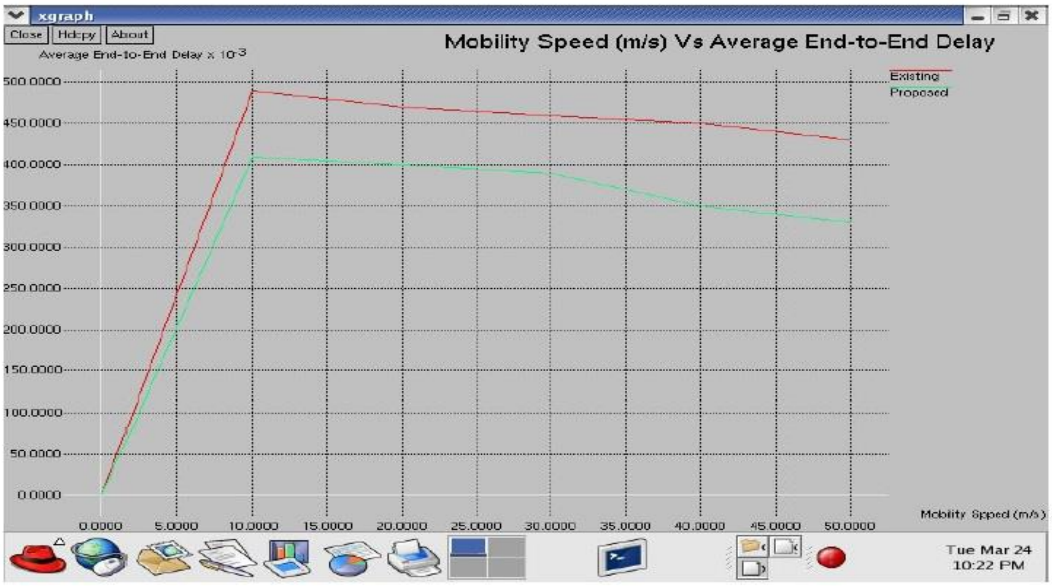


FIG 8.4 MOBILITY SPEED Vs END-TO-END DELAY

CONCLUSION:

Each update consumes node energy, wireless bandwidth, and increases the risk of packet collision at the medium access control (MAC) layer. Packet collisions cause packet loss which in turn affects the routing performance due to decreased accuracy in determining the correct local topology (a lost beacon broadcast is not retransmitted). A lost data packet does get retransmitted, but at the expense of increased end-to-end delay. Clearly, given the cost associated with transmitting beacons, it makes sense to adapt the frequency of beacon updates to the node mobility and the traffic conditions within the network, rather than employing a static periodic update policy. In our system, we use Adaptive Position Update strategy.Using this strategy, we can update the node position and velocity dynamically. The system use Periodic beaconing scheme, node can broadcast the beacon for fixed interval because this research based on proactive model.We studied the different recovery delays consecutive to a link failure and observed that this delay, under several topologies and mobility scenarios, was significant and incompatible with delay constrained applications.The simulation studies demonstrate that the proposed routing protocols are more robust and outperform the existing geographic routing protocol and conventional on demand routing protocols under various conditions including different motilities, node densities and traffic loads.

REFERENCES:

[1] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, Aug. 2001.
[2] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proc. ACM MobiCom, pp. 243-254, Aug. 2000.
[3] L. Blazevic, S. Giordano, and J.-Y. LeBoudec, "A Location Based Routing Method for Mobile Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 4, no. 2, pp. 97-110, Mar. 2005.
[4] Y. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," ACM/Baltzer Wireless Networks, vol. 6, no. 4, pp. 307-321, Sept. 2002.
[5] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks," Proc. IEEE INFOCOM, pp. 1678-1687, June 2002.
[6] D. Johnson, Y. Hu, and D. Maltz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, IETF RFC 4728, vol. 15, pp. 153- 181, Feb. 2007.
[7] C. Perkins, E. Belding-Royer, and S. Das, Ad Hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, July 2003.
[8] J. Li, J. Jannotti, D.S.J.D. Couto, D.R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," Proc. ACM MobiCom, pp. 120-130, Aug. 2000.
[9] Z.J. Haas and B. Liang, "Ad Hoc Mobility Management with Uniform Quorum Systems," IEEE/ACM Trans. Networking, vol. 7, no. 2, pp. 228-240, Apr. 1999.

Cite this article as: Siva Ranjani, Ranjani N, Sri Abarna, Ananthi K, Parkavvi. "EFFICIENT ROUTING AND FALSE NODE DETECTION IN MANET." *International Conference on Information Engineering, Management and Security (2015)*: 27-42. Print.