PILOT: A Power-Aware, Location Driven Traffic Forwarding Algorithm for MANETs

Anandha Gopalan, Taieb Znati and Sameh Gobriel
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260, U.S.A
Email: {axgopala, znati, sameh}@cs.pitt.edu

***Abstract*—**
This paper presents PILOT, a new data-forwarding algorithm for ad-hoc networks, that takes into consideration the change in location of the destination over time. PILOT is also power-aware and the rate of increase of the *probability of forwarding* is closely tied to the current energy level at the node. We conclude the paper by providing an experimental study of the proposed traffic forwarding algorithm. The results show that in most of the cases, PILOT outperforms both AODV and LAR.

***Index Terms*—**
Ad-hoc networks, directional forwarding, power-aware.

I. Introduction

Advances in wireless technology and portable computing along with demands for greater user mobility have provided a major impetus toward development of an emerging class of self-organizing, rapidly deployable network architectures referred to as ad-hoc networks. Ad-hoc networks, which have proven to be useful in military applications, are expected to play an important role in future commercial settings where mobile access to a wired network is either ineffective or impossible. Approaches towards infrastructure-less network solutions from vendors like Apple's rendezvous protocol [4], IETF's zeroconf protocols [15] and Bluetooth [6] are very important steps in this direction.

Due to the absence of an infrastructure, multi-hop routing in ad-hoc networks is achieved by using the nodes in the network. Node mobility, coupled with the limitation of computational and communication resources, brings about a new set of challenges that needs to be addressed. Routing algorithms designed for ad-hoc networks must thus, take into consideration the time-varying dynamics of the network, node mobility and power-consumption. The tradeoffs between these design factors must also be carefully evaluated.

Technologies like GPS [5] allow the routing algorithms to take advantage of the knowledge of a node's physical location. This *location information* can be used to *optimize* a routing algorithm, by sending information in a direction that is *geographically* closer to the destination, rather than broadcasting it.

The focus of this paper is on the development of a power-aware, location-driven traffic forwarding algorithm for MANETs. The resource constrained environment of ad-hoc networks necessitates the need for the routing algorithm to be energy efficient. Node mobility must also be taken into consideration while designing the routing protocol.

The main contribution of this paper is: PILOT, a power-aware, location-driven traffic forwarding algorithm for MANETs. PILOT handles the mobility of nodes in the network by using the *expected* speed and direction of a mobile node to predict the current location of the node. PILOT is also power-aware since it ties the *probability of forwarding* the message at each forwarding node to the location information of the node along with the information gathered from the message to be forwarded, namely: energy-level of the forwarding node, direction of the destination, positional relevance of forwarding node to destination. Also, messages are not flooded in the network, rather they are sent in the direction of the destination.

PILOT was implemented in the Glomosim network simulator and evaluated under various network conditions. The results show that the proposed algorithm out performed both LAR and AODV in most of the cases, especially when the density of the network increases.

The rest of the paper is organized as follows: Section II details the work related to this paper, Section III details the characteristics of PILOT, Section IV details the simulations and the results and Section V concludes the paper and identifies areas of future work.

II. Related Work

The basic classifications of routing protocols in ad-hoc networks are: pro-active, re-active and hybrid.

Pro-active routing protocols like DSDV [13] establish and maintain routes periodically in the network. Routes are usually available before they are needed by the nodes. Route maintenance is of the highest priority with such protocols. In a highly mobile network, the cost associated with route maintenance can be very high.

Re-active routing protocols like DSR [8] and AODV [12] establish routes as and when needed by the nodes in the network. Route maintenance is very minimal and a route is usually maintained only for the lifetime of the connection. These protocols can introduce a high latency during the route discovery phase. PILOT is a re-active routing protocol and routes are established only when needed.

Hybrid routing protocols like ZRP [7] and the (α, t)-Cluster [1, 11] use a combination of pro-active and re-active routing protocols. They abstract the highly mobile elements of the network into *clusters* and use a pro-active scheme to route within a cluster, while routing between clusters is achieved by using a re-active scheme.

There is another class of routing protocols for ad-hoc networks that rely on the position of a node in space rather than on the topology of the network. These kind of protocols rely on the fact that the nodes in the network know their location (using a service similar to GPS [5]). This information is used to optimize the routing protocol by sending the routing information in a direction that is *closer* to the destination rather than broadcasting this information. Examples of location based routing protocols are: LAR [9] and DREAM [14]. Traditional location-based routing protocols do not account for the change in the position of the destination over time; this leads to a performance loss in the network if the destination changes its position over time. DREAM and LAR also suffer from an increase in network traffic due to the flooding of location updates.

PILOT differs from DREAM and LAR by not flooding the network with location updates, rather, messages are forwarded by intermediary nodes on a piece-meal basis (where the position of the destination is re-calculated and hence the direction of forwarding is changed to suit the direction of the destination). This ensures that our message forwarding algorithm is scalable when compared to DREAM and LAR.

III. PILOT Characteristics

A. Piece-meal, location-driven traffic forwarding Algorithm

This section describes in detail the traffic forwarding algorithm that takes into consideration the change in the position of the destination as the destination moves. Section III-B describes in detail the protocols used to make the traffic forwarding algorithm a power aware one, by tying the *probability of forwarding* to the *power* available at an intermediary node.

Let us consider a scenario, when a node S tries to contact another node D. Using the knowledge about the position of D and its *expected* direction and speed, node S tries to route the traffic to node D [1]. To reduce flooding in the network, the traffic is sent in a cone-shaped fashion towards D as shown in figure 1, (similar to [14], but here all the nodes need not know about the position of every other node in the network). The nodes in zone 1 have the highest priority to forward the traffic, while the nodes in zone 2 have a lower priority. If a node in zone 1 fails to forward the traffic or there is no node in zone 1, nodes in zone 2 forward the traffic towards D.

The nodes that receive the message sent by S calculate their priorities and based on this information, they either listen or forward the message. Initially, only nodes in zone 1 have the priority to forward the traffic, while nodes in zone 2 listen to the messages. In case, there are no nodes in zone 1, nodes in zone 2 will forward the message to D (this happens after a timeout). Each node that receives this routing information re-calculates the position of D using its *expected* speed ($V(t_0)$) and direction ($D(t_0)$) and thus re-calculates the *cone* based on this information before sending the messages. This forwarding of messages happens on a *piece-meal* basis. This routing algorithm is explained as a *pseudo code* in algorithm 1.

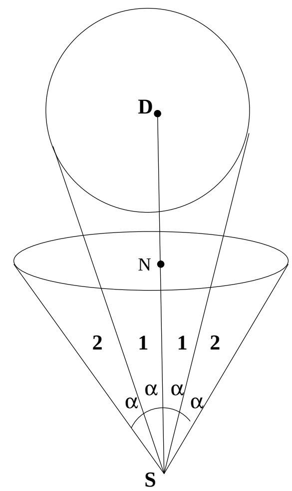

Fig. 1. Directional Routing

The components of the *vector* $[V(t_0), D(t_0), P_V(t), P_D(t)]$ in algorithm 1 are:

- $V(t_0)$: expected average starting speed.
- $D(t_0)$: expected average initial direction.
- $P_V(t)$: Predictor for speed after t units of time since departure.
- $P_D(t)$: Predictor for direction after t units of time since departure.

The important aspect of algorithm 1 is to calculate the priorities based on which the nodes decide whether to forward the message or not. The priority function P_n is the probability of forwarding (for each node n) and is dependent on α_n (the angle this node makes with the source) and d_n (the distance of this node from the source). Let the angle of the cone be α_c (from figure 1, this would be α). Consider the figure 1, node N is the best node in zone 1 to forward the message towards the destination and hence must have the highest priority. Our formula for calculating the priority must reflect this. We must choose a node within the cone that is the farthest away from the source and also be in the direction of the destination. To balance these two factors, weights are added to each factor in the formula [2]. Let R_c be the distance of N from S; this is the radius of the cone. The formula for calculating the priority is given by:

$$P_n = \begin{cases} w_1 * \frac{d_n}{R_c} + w_2 * \frac{(\alpha_c - \alpha_n)}{\alpha_c} & \\ 0 & d > R_c \ or \ \alpha_n > \alpha_c \end{cases} \quad (1)$$

It can be clearly seen that $P_n \leq 1 \ iff \ w_1 + w_2 \leq 1$.

The value of P_n is highest for N, since α_n is 0 and $d = R_c$.

To calculate the value of α_n, we use the *law of cosines*. Consider figure 2, where S is the source and D is the destination and I is the intermediary node. I_D is the distance from I to D and S_D is the distance from S to D. The angle α_n is given by the equation 2.

[1] [2] contains the information on how node S receives information about the *expected* speed and direction of D

[2] the weights can be determined using simulation or analytical analysis

Algorithm 1: Forwarding Messages
Input: Void
Output: Result
FORWARD-MSG(α_n, d_n, $M(D)$)
Let P_n be the forwarding probability of node n
Calculate P_n using α_n, d_n
While (!success)
    Generate a random number in $[0,1]$
    With P_n
        Calculate $V_L(D) = [V(t_0), D(t_0), P_V(T), P_D(T)]$, where $V_L(D)$ gives the expected location of node D
        Send limited-directed broadcast of $[V_L(D), M(D), L(N)]$ as shown in figure 3, where $M(D)$ is the message and $L(N)$ is the location of this node N
        success = *true*
    With $1 - P_n$
        Wait for the next time slot
        if Msg-Sent by another node before timeout
            drop request
            return Success
        else
            continue
return Success

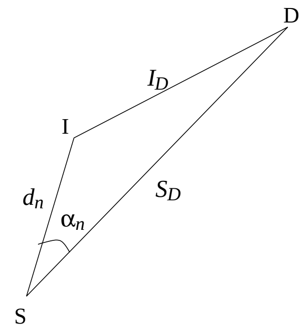

Fig. 2. Calculation of α_n

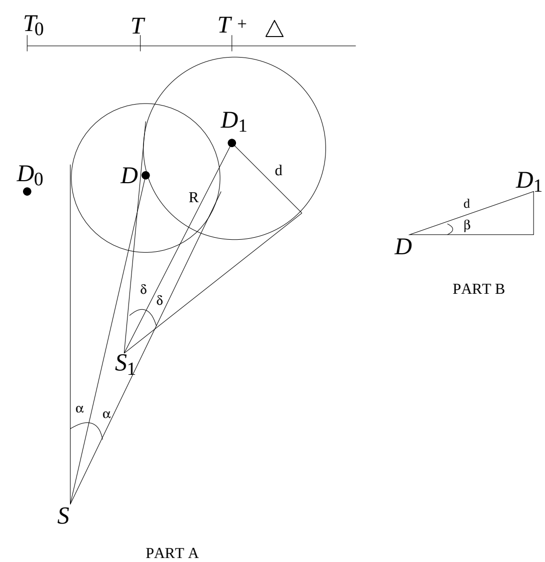

Fig. 3. Directional Routing as destination moves

$$\alpha_n = arccos[\frac{d_n^2 + S_D^2 - I_D^2}{2 * d_n * S_D}] \tag{2}$$

Another important piece of the algorithm is the *directional routing* that requires each node along the path to re-calculate the *cone* used to forward the message to the destination (shown in figure 3 (part A)). Consider the timeline in figure 3 (part A). At time T_0, node D is at position D_0, at time T, the node is at position D and at time $T + \triangle$, node D is at position D_1. Now, consider the situation, when source S wants to send a message to D. It calculates with a certain probability [3], a region where node D can reside. Node S now calculates the angle α and hence derives the *cone* and sends the message towards the destination. Node S_1 upon receipt of this message, re-calculates the angle δ and hence re-calculates a new cone (based on the probability of the new position of D) and sends the message. The most important aspect of this protocol is the calculation of the cone, which is based on two things: the *expected speed* and the *expected direction* of D.

Consider part B of figure 3, this shows the movement of the destination from position D to D_1. Assume the expected direction of travel to be β with respect to the $x-axis$ and the expected speed of travel to be v. Let D be the point (x_1, y_1) and D_1 be the point (x_2, y_2) in the cartesian co-ordinate system. We know the position D and we need to find out the new location D_1 in terms of D. This is accomplished using the following equations (derived using simple laws of motion and trigonometry):

$$d = v * (T - (T + \triangle)) \Longrightarrow d = v * \triangle \tag{3}$$

$$x_2 = x_1 + d * cos\beta \tag{4}$$

$$y_2 = y_1 + d * sin\beta \tag{5}$$

Consider the scenario when node S_1 receives a message from S that is intended for the destination, now S_1 needs to calculate the angle for the cone. Now that we have the position D_1 and the value for d, we need to calculate δ based upon these values. Let S_1 be the point (sx_1, sy_1). The following sets of equations help us derive δ:

Using the values of x_2 and y_2 from equations 4 and 5 respectively, we get:

$$R = \sqrt{(x_2 - sx_1)^2 + (y_2 - sy_1)^2} \tag{6}$$

Using the value of R from equation 6 and the value of d from equation 3, we get:

$$\delta = arcsin(\frac{d}{R}) \tag{7}$$

This algorithm is used by all the nodes to build the cone to forward the traffic towards the destination. The advantage with this scheme is that this scheme adapts to the mobility of the node, the lesser a node moves, the smaller the cone, and the greater a node moves, the larger the cone.

B. Power-Aware forwarding algorithm

As discussed in section III-A, the probability P_n, of a node to forward a message to the destination depends both on the distance from the sender node and the angle of deviation from the center line of the cone. However, P_n defines only the initial forwarding probability at the instant when a message arrives at the relay node. If the relay node overhears that the route-request message has been forwarded by another node, it drops the message, if not, the probability of forwarding the message should build up as time elapses. The forwarding probability has to reach 1 by S time slots, where S is a design parameter that is directly related to the expected number of hops and the message-reply timeout.

For choosing the next relay node to forward the message to the destination, the best candidate is the node that is located on the cone center line and is farthest from the source (maximum P_n). However, the current energy level of the relay node is also important. If an energy-poor node is picked as the next relay node, its energy will be depleted in transmitting the packets and this node will soon die. From the network lifetime point of view, the low energy nodes are the most important and also the most critical. These nodes have used their energy either because they have a lot of data to send or because they are located in the confluence of many routes. Leaving these critical nodes to deplete their energy may cause a network partition and some sources might be unable to reach other destinations, or at least there is an energy and bandwidth overhead associated with re-routing the packets after discovering a broken link (dead nodes). Bottom line is that we want to construct the best available route from the source to the destination while maximizing the network lifetime.

As a result, in our design, the current energy level of a node affects the probability of choosing this node as the next relay host. This is done by making the rate of increase of P_n, a function of the node's energy level. The higher the energy level, the higher the probability increase rate and vice versa. For example, consider the case when a message arrives at 2 nodes, node A, a high-energy node and node B that has a lower energy level but has a higher initial probability of forwarding (P_n). Though node B has an initial higher probability, as time goes by, the probability of forwarding for node A increases at a faster rate; as a result of which node A may forward the request message sooner than node B and, consequently, it may be chosen as the next relay node instead of node B.

The *probability of forwarding* function ($\Gamma(p,t)$), thus depends on the energy p and the time slot t at a node and has the following properties:

$$\Gamma(p,1) = P_n \tag{8}$$
$$\Gamma(p,t+1) \ > \ \Gamma(p,t) \tag{9}$$
$$\Gamma(p+1,t) \ > \ \Gamma(p,t) \tag{10}$$
$$\Gamma(p,S) \ = \ 1 \tag{11}$$

Since the forwarding probability P_n has to reach 1 by the end of S slots, we have to derive the forwarding probability as a strictly increasing function that starts from P_n and reaches 1 as $t \to S$. Also, the function must produce a family of curves depending on the power at a node. The higher the power at the node, the faster the curve reaches 1. Equation 12 has the requisite property and produces a family of curves that start at 0 and reach 1 depending on the value of v.

$$F(x) = 1-(1-x)^v \tag{12}$$

To illustrate the property of this function, we plot this function for varying values of v, the result of which is shown in figure 4.

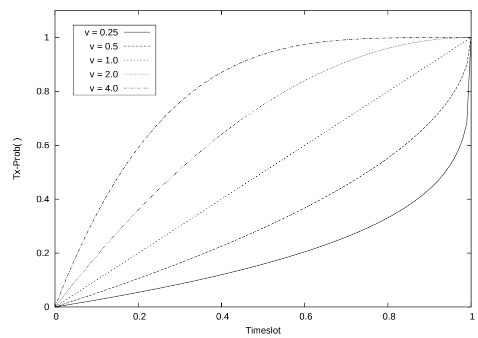

Fig. 4. F(x) with variable v

The function given in equation 12 has the requisite property by giving us a family of curves that start at 0 and reach 1 depending on the value of v. However, equation 12 does not satisfy equations 8 and 11. Hence, we add the following constraints and substitute $x = \frac{t}{S}$ (t = time slot and S = max time slot).

$$F(\frac{t}{S}) = \begin{cases} P_n & t=1 \\ 1 & t=S \end{cases} \tag{13}$$

Equation 12 is in the format $a*x + b$, where $x = \frac{t}{S}$ (t = time slot and S = max time slot). While obeying the constraints in equation 13, we can solve Equation 12 (for details, refer to appendix A) to get the overall probability of forwarding $p(f)$, which is given in equation 14.

$$p(f) \ = \ 1-\left(\frac{S \ * \ (1-P_n)^{\frac{1}{v}} * (\frac{t}{S}-1)}{(1-S)}\right)^v \tag{14}$$

where P_n is the initial forwarding probability as defined by equation 1, t is the elapsed time slots since the instant of route-request arrival.

Using equation 14, we can plot the function as shown in Figure 5 by varying the value of v from 0.25 to 4. The initial probability is set to 0.2. We can observe from the figure that equation 14 observes the properties stated in equations 8 - 11. It also has the property of the probability increasing faster for a higher value of v and vice versa.

The parameter v in equation 14 is a function of the current energy level of the node. When the node has a low energy level, v should be low, which forces the forwarding probability to increase at a slow rate. On the other hand, when the node has a

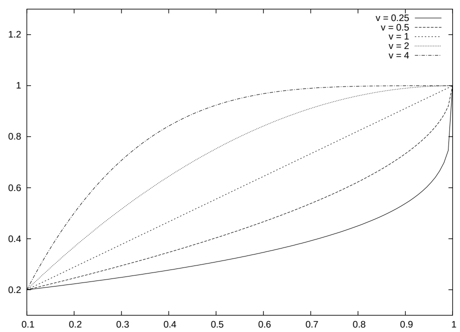

Fig. 5. The probability of forwarding while changing the value of v

high energy level, v must be high, thus forcing the forwarding probability to increase at a higher rate. As a result, probabilistically, the likelihood of picking a high-energy node for the relay host role increases as time goes by. It should be noted that, for mid-range energy level the increase of the forwarding probability is almost linear with time. The parameter v can be computed from the node's current relative energy level as follows:

$$v_i = 2^{4\cdot(R_i - 0.5)} \tag{15}$$

where R_i is the relative energy level (normalized to full energy) of node i.

It should be mentioned that, although equation 14 is a computationally expensive function to evaluate, the wireless node does not need to compute its value online. A matrix $\pi(K,S)$, which defines the function value at each S value and for different K energy levels can be computed offline and then used by the wireless node. The values of both K and S are to be determined at design time. Table I presents an instance of this matrix when we have $K = 5$ and $s = 5$.

TABLE I
$p(f)$ WITH $K = 5$ AND $S = 5$

	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
$R=0.2$	0.093	0.199	0.329	0.503	1
$R=0.4$	0.156	0.321	0.501	0.705	1
$R=0.6$	0.255	0.492	0.701	0.881	1
$R=0.8$	0.401	0.691	0.878	0.975	1
$R=1$	0.59	0.871	0.974	0.998	1

Using the knowledge of the energy at a node and the matrix in table I, Algorithm 1 can now be re-written as Algorithm 2.

Algorithm 2: Power-Aware Forwarding
Input: Void
Output: Result
FORWARD-MSG(α_n, d_n, $M(D)$)
(1) Let P_n be the forwarding probability of node n
(2) Calculate P_n using α_n, d_n
(3) While (!success)
(4)   Generate a random number in $[0,1]$
(5)   With P_n
(6)   forward message
(7)   success = *true*
(8)   With $1 - P_n$
(9)   Wait for the next time slot
(10)   update P_n for that time slot according to the matrix reflecting the current power of the node (table I)
(11)   **if** Msg-Sent by another node before timeout
(12)   drop request
(13)   **return** Success
(14)   **else**
(15)   **continue**
(16) **return** Success

IV. SIMULATION AND RESULTS

This section explains in detail the simulation environment used and the ensuing results.

The protocol was implemented using the Glomosim network simulator [10] on Linux and was tested by providing different network scenarios. The first set of tests were conducted as part of the sensitivity analysis of the protocol. The second set of tests compared the performance of PILOT to LAR [9] and AODV [12]. We chose to compare this protocol to LAR and AODV because, LAR is also an on-demand location-based routing protocol and has been used as a benchmark for comparison of location-based routing protocols and AODV is an on-demand routing protocol of a different nature.

A basic sensitivity analysis of the protocol provides us with some ideas about the performance of the protocol under different network conditions. For this reason, we simulated four different types of networks differentiated by the channel characteristics and the mobility of the nodes. The channel characteristics used were those that were available in Glomosim: *TWO-RAY* (where the receiving antenna sees two signals, a direct path signal and a signal reflected off the ground) and *FREE-SPACE* (where radio wave propagation is in the absence of any reflections or multipath). The mobility models available in Glomosim that were used during the experiments were: *NO-MOBILITY*, where nodes are static and *RANDOM-WAYPOINT* mobility, where a node randomly chooses a destination and move towards that destination. The speed of the node is chosen randomly between an upper limit (10 m/s) and a lower limit (1 m/s). Upon reaching the destination, the node pauses for a *pause time* of 30s, before becoming mobile again. For the comparative analysis with LAR and AODV, we compare the throughput for a network of mobile nodes under various network conditions.

In all experiments, the throughput was measured with respect to the parameters of the network. The nodes in the network were randomly placed in a grid of size 3000x3000m. Traffic generated was CBR traffic with two different sources and two different destinations, to ensure some congestion in the network. Traffic statistics are collected at the destination by measuring the total time taken (in $nano-seconds$) for the packets to reach the destination. The total number of packets sent during the simulations were 20 packets from each source. For the comparative analysis, the parameters varied were: number of nodes, node *pause time*, transmission range and average speed. Each experiment was run 10 times and the results used were averaged over these experiments. Table II shows a summary of the parameters used during the simulation and Table III shows the parameters used for the *RANDOM-WAYPOINT* mobility model.

TABLE II

SUMMARY OF SIMULATION PARAMETERS

Name	Value
No. of nodes	100 - 500
Channel Characteristic	TWO-RAY, FREE-SPACE
Transmission Range	100 - 500m
Grid Size	3000x3000m
Type of traffic	CBR
Size of packet	512 bytes
Number of packets	20
Mobility Patterns	NO-MOBILITY, RANDOM-WAYPOINT
Length of simulation	240s

TABLE III

PARAMETERS FOR THE *RANDOM-WAYPOINT* MOBILITY MODEL

Name	Value
Node Speed (upper limit)	1m/s, 5m/s, 10m/s, 20m/s
Node Speed (lower limit)	1m/s, 5m/s, 20m/s
Node Pause Time	0s, 15s, 30s

A. Sensitivity Analysis

In this section, we analyze the results of the experiments performed to do a sensitivity analysis of PILOT. In the experiments, the throughput was measured with respect to the density of the network (number of nodes in the network). The number of nodes was varied from 100 - 500. The network thus simulated varies from a sparsely populated network to a densely populated network.

The first set of experiments depicted in figures 6 and 7 respectively, were performed by varying the channel characteristics, while keeping the nodes in the network as static.

From Figure 6, we conclude that the throughput increases as the density of the network grows. This can be attributed to the fact that as the network size increases, there is a higher probability of a node being available in the path towards the destination and hence this node can forward the packet towards the destination.

From Figure 7, we find out that the throughput is very high even for a network that is not very dense. The transmission range for the node is higher when using the channel, *FREE-SPACE* when compared to using the channel *TWO-RAY*. This increase in transmission radius leads to a lower hop-count for

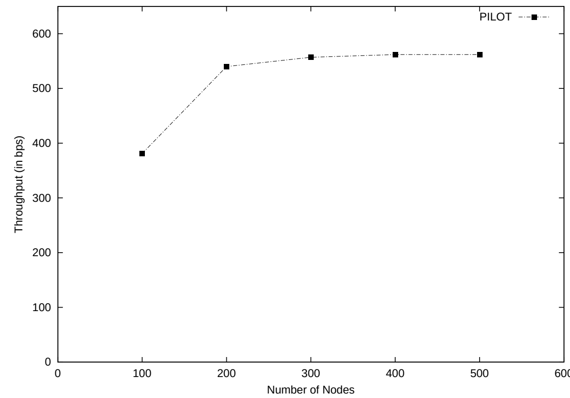

Fig. 6. Channel Characteristic: Two-Ray (Using PILOT)

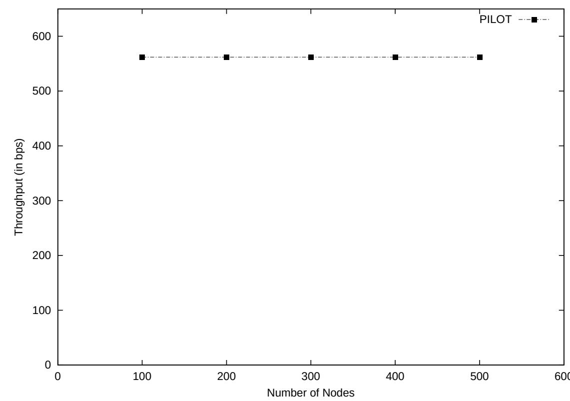

Fig. 7. Channel Characteristic: Free-Space (Using PILOT)

a packet that is transmitted from the source to the destination, thus leading to a substantially higher throughput.

Figures 8 and 9 show the result of the experiments when mobility is introduced into the network. The nodes follow the *RANDOM-WAYPOINT* mobility pattern and there is no assumption made as to which nodes are static and which nodes are mobile in the network.

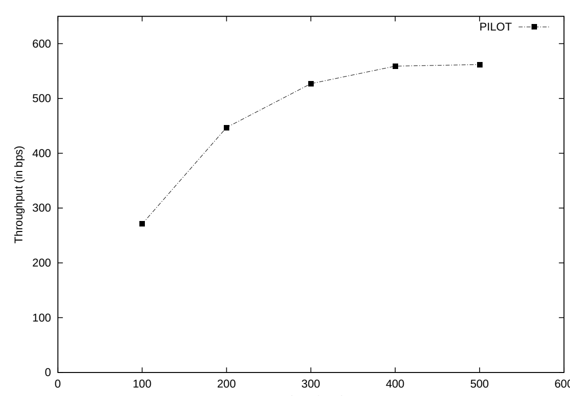

Fig. 8. Mobility with Channel Characteristic: Two-Ray (Using PILOT)

Figure 8 shows the case when the channel characteristic used is *TWO-RAY*. We observe that the throughput is slightly lower in this case when compared to the static case. This is to be expected due to the mobility in the network. The throughput though, increases as the density of the network increases, with the value finally reaching the value of the throughput achieved in the static case.

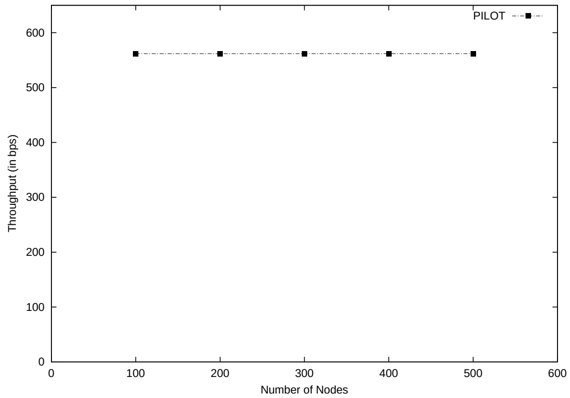

Fig. 9. Mobility with Channel Characteristic: Free-Space (Using PILOT)

From Figure 9, we can conclude that even when mobility is introduced into the network, the throughput is very high. In fact, the throughput matches the throughput achieved in the static case. This can be attributed to the channel characteristic *FREE-SPACE*, which has an increased transmission range, thus increasing the probability of finding a node that can forward the message towards the destination.

B. Comparative Analysis

In this section, we do a comparative study by comparing PILOT to LAR and AODV in terms of the throughput achieved for a network consisting of mobile nodes by varying various network parameters. The statistics collected for LAR and AODV were available as part of their implementations that are provided in Glomosim. For all experiments, the *RANDOM-WAYPOINT* mobility model was used.

The first set of experiments measure the impact of continuous mobility on the different routing protocols. For this experiment, the transmission range and the speed of the nodes are not varied. The transmission range is dependent on the channel characteristic, which in this case is *TWO-RAY*. The speed of the node is chosen randomly between an upper limit (10 m/s) and a lower limit (1 m/s). Upon reaching the destination, the node pauses for a *pause time* of either 0s or 30s, before becoming mobile again. The number of nodes in the network was varied from 100 - 500.

Figure 10 shows the result of the experiment when the *pause time* is 0s. This indicates that nodes are continuously moving in the network. We can see from the result that the throughput of all the network protocols increases as the network density increases. This is to be expected due to the availability of more nodes, and hence more routes in the network. For a network consisting of 100 nodes, PILOT performs almost as well as AODV, but not as well as LAR. This is due to a paucity of nodes in the network to forward the message towards the destination. Once the density of the network increases, which leads to more nodes available to forward the message towards the destination, PILOT outperforms both AODV and LAR.

From Figure 11, we can conclude that the throughput is higher in this case than the case when the *pause time* is 0s.

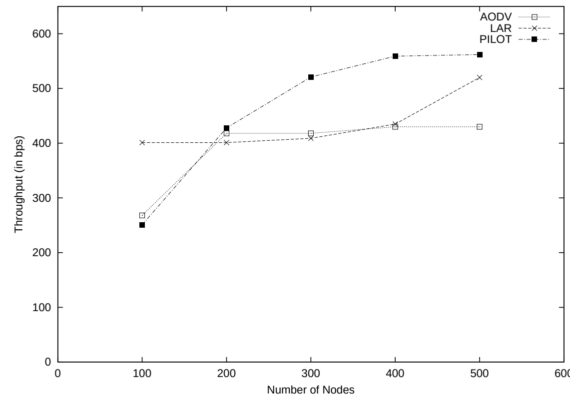

Fig. 10. No Pause Time

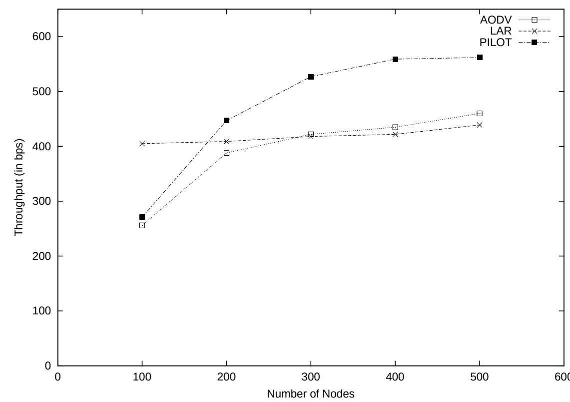

Fig. 11. Pause Time of 30s

The *pause time* used in this experiment is 30s. Due to a higher *pause time*, the nodes are more stable and hence the routes in the network remain for a longer period of time and hence the higher throughput. As in the previous case, PILOT performs better than AODV and LAR as the node density increases in the network.

The second set of experiments measure the impact of the transmission range on the different routing protocols. For this experiment, the number of nodes in the network is set to 250 and the nodes have a *pause time* of 15s. The experiments are conducted with the average speed of the node being 5m/s and 20m/s. The transmission range of the nodes in the network was varied from 100 - 500.

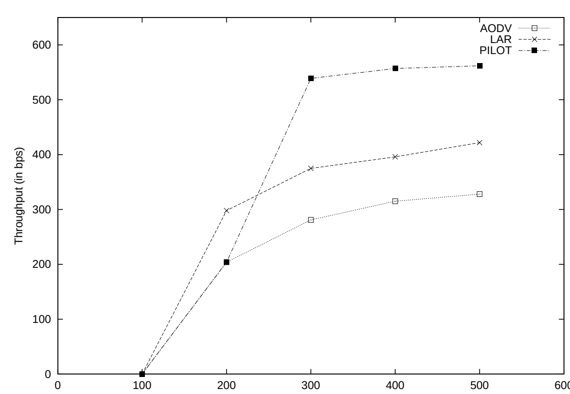

Fig. 12. Average Speed: 5m/s

From figure 12, we can notice that the throughput is very low for all the protocols when the transmission range is very low. This is to be expected, since the possibility of finding a node in the vicinity of the source to forward traffic to the destination is very low, given the low transmission range. We can concur from this experiment that PILOT performs as well as AODV for the transmission ranges of 100 and 200 but not as well as LAR, but for a network with transmission range greater than 300, PILOT performs better than both LAR and AODV.

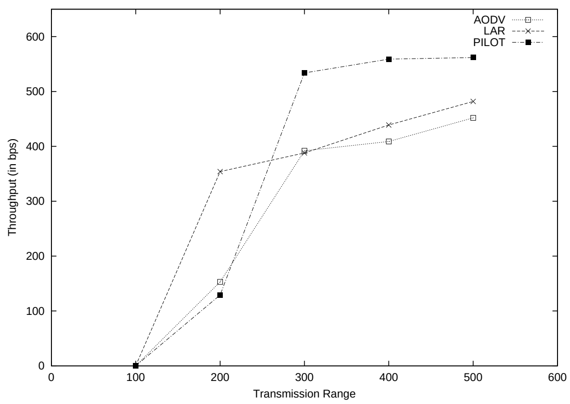

Fig. 13. Average Speed: 20m/s

Figure 13 shows that the throughput drops a little when compared to the previous case for all routing protocols. This is because of an increase in the average speed of the nodes, thus leading to more paths breaking in the network. PILOT is primarily a forwarding protocol and hence it does not incur the cost associated with forming and repairing routes. It can be seen that PILOT performs almost as well as AODV, but not as well as LAR for the transmission ranges of 100 and 200. Upon increasing the transmission range (> 300), PILOT outperforms both LAR and AODV. Lower overhead with respect to repairing routes and an increase in transmission range (which leads to an increase in the number of available nodes that can forward the packet towards the destination) leads to PILOT performing better than LAR and AODV.

The third set of experiments measure the impact of node density on the different routing protocols. For this experiment, the transmission range is not varied. The transmission range is dependent on the channel characteristic, which in this case is *TWO-RAY* (378m). The experiments are conducted with the average speed of the nodes varied between 5m/s and 20m/s. The number of nodes in the network was varied from 100 - 500.

From figure 14, we notice that PILOT performs better than both LAR and AODV. As nodes become mobile in the network, routes that were discovered by LAR at the beginning of the simulation may not be valid later and hence another route discovery must be performed. This overhead increases the latency to send packets from the source to the destination. PILOT is primarily a forwarding protocol and hence it does not incur the cost associated with forming and repairing routes. At 500 nodes (highly dense network), PILOT achieves the maximum throughput (the same achieved by PILOT in the static case with channel characteristics *TWO-RAY* and *FREE-SPACE*). The denser the network becomes, the better PILOT performs due to the availability of

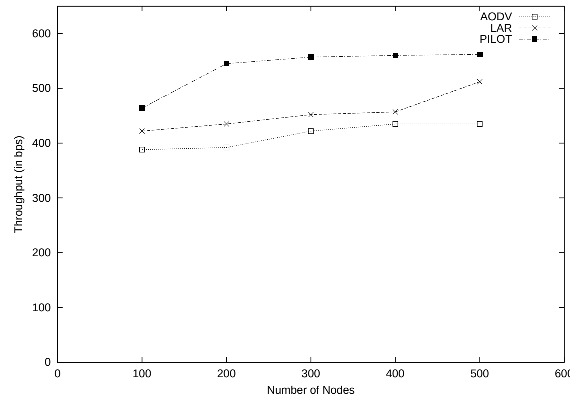

Fig. 14. Average Speed: 5m/s

more nodes in the network that can forward the packet towards the destination.

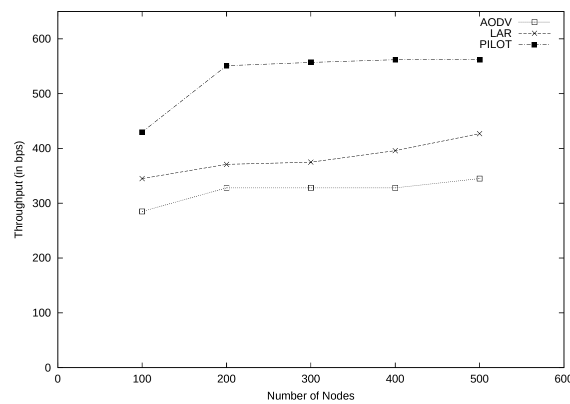

Fig. 15. Average Speed: 20m/s

Figure 15 shows that the throughput drops slightly when compared to the previous case. This is to be expected due to the higher average speed of the nodes in the network. We can notice that PILOT performs better than both LAR and AODV and at 500 nodes achieves its highest throughput. As mentioned earlier, the increase in the density of nodes in the network leads to more nodes being available for forwarding the traffic from the source to the destination.

The fourth experiment measures the impact of the average speed on the different routing protocols with respect to the throughput in the network. For this experiment, the transmission range and the number of nodes in the network are not varied. The transmission range is dependent on the channel characteristic, which in this case is *TWO-RAY*. The number of nodes in the network is set to 250. The average speed of the nodes are varied from 10m/s to 100m/s. The node *pause time* is set to 30s.

Figure 16 shows that, as the average speed of the node increases, the throughput goes down in all cases. This can be attributed to the increased mobility in the network due to an increase in the average speed. In this scenario, we can observe that PILOT performs better than both LAR and AODV. This is because, even with increased mobility, there is a high enough probability of finding a node that can be used to forward the traffic from the source towards the destination. As the average

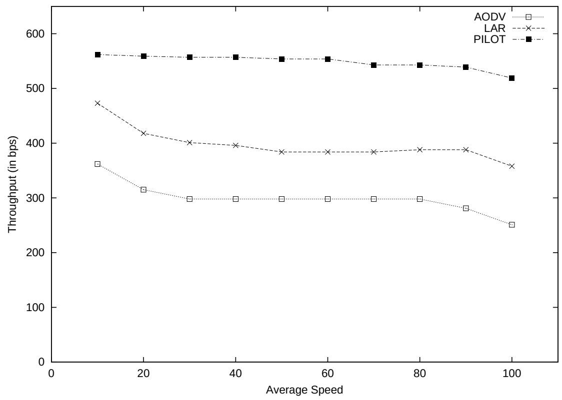

Fig. 16. Impact of Average Speed

speed increases, the overhead associated with maintaining and discovering routes in the network increases, thus decreasing the throughput. PILOT does not incur this overhead, and hence its throughput is higher than that of LAR and AODV.

V. Conclusion and future work

The major contribution of this paper is in providing a power-aware traffic forwarding algorithm for MANETs. The forwarding algorithm handles the mobility of nodes in the network by using the *expected* speed and direction of a mobile node to predict the location of the node. This algorithm is also power-aware and the rate of increase of the *probability of forwarding* is closely tied to the current energy level at the node. A power matrix is used to calculate the *probability of forwarding* for each node rather than calculating it on the fly, which would be computationally expensive. The proposed traffic forwarding algorithm was implemented in the Glomosim network simulator and evaluated by providing different network scenarios. A sensitivity analysis of the algorithm was performed, after which it was compared to LAR and AODV. It was found that PILOT out performed both LAR and AODV in most of the cases, especially when the density of the network increases.

There is a lot of potential for future work in this area. Security is an important aspect that must be incorporated into the traffic forwarding algorithm.

References

[1] A. Gopalan, S. Dwivedi, T. Znati and A. B. McDonald. On the implementation of the (α,t)-Cluster Protocol on Linux. In *Proc. 37th Annual Simulation Symposium*, Apr. 2004.

[2] A. Gopalan, T. Znati and P. K. Chrysanthis. Structuring Pervasive Services in Infrastructureless Networks. In *Proc. IEEE International Conference on Pervasive Computing (ICPS 2005), to appear*, 2005.

[3] A. R. Aljadhai and T. Znati. Predictive mobility support for QoS provisioning in mobile wireless environments. *IEEE Journal on Selected Areas in Communications*, 19(10):1915–1930, 2001.

[4] Apple Computer Inc. Rendezvou's Developer Page. *http://developer.apple.com/macosx/rendezvous/*, 2002.

[5] Ed. E. D. Kaplan. *Understanding GPS: principles and applications*. Artech House, Boston, MA, 1996.

[6] J. C. Haarsten. The Bluetooth Radio System. *IEEE Personal Communications Magazine, pp. 28-36*, February 2000.

[7] Z. J. Haas and M. Pearlman. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. *Internet Draft*, Aug. 1998.

[8] David B Johnson and David A Maltz. Dynamic Source Routing in Ad hoc Wireless Networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[9] Y. B. Ko and N. H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad-Hoc Networks. In *Proc. ACM/IEEE MOBICOM*, Oct. 1998.

[10] L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia and M. Gerla. Glomosim: A scalable network simulation environment. Technical Report 990027, UCLA, 13, 1999.

[11] A.B. McDonald and Taieb Znati. A Mobility Based Framework for Adaptive Clustering in Wireless Ad-Hoc Networks. *IEEE Journal on Selected Areas in Communications (J-Sac), Special Issue on Ad-Hoc Networks*, 17(8), Aug. 1999.

[12] C. Perkins and E. Royer. Ad Hoc On-Demand Distance Vector Routing. In *Proceedings 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, 1999.

[13] C. R. Perkins and P. Bhagwat. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM*, pages 234–244, Oct. 1994.

[14] S. Basagni, I. Chlamtac, V. R. Syrotiuk and B. A. Woodward. A Distance Routing Effect Algirithm for Mobility (DREAM). In *Proc. ACM/IEEE Mobicom*, pages 76–84, October 1998.

[15] A. Williams. Zero Configuration Networking. *Internet Draft*, September 2002.

Appendix

A.

We repeat equation 12 over here:

$$F(x) = 1 - (1-x)^{v} \tag{16}$$

Equation 16 is in the format $1 - (a \cdot x + b)$, where $x = \frac{t}{S}$ (t = time slot and S = max time slot).

Using $t = S$ in equation 16, we get:

$$a + b = 0 => a = -b \tag{17}$$

Using equation 17, we get:

$$\left(\frac{a}{S} - a\right)^{v} = 1 - P_n \tag{18}$$

$$=> \quad \frac{a - S \cdot a}{S} = (1 - P_n)^{\frac{1}{v}} \tag{19}$$

$$=> \quad a = \frac{S \cdot (1-P_n)^{\frac{1}{v}}}{1-S} \tag{20}$$

Using the value of a from equation 20, we get:

$$b = -\frac{S \cdot (1-P_n)^{\frac{1}{v}}}{1-S} \tag{21}$$

Using the values of a and b from equations 20 and 21 respectively, we get:

$$F\left(\frac{t}{S}\right) = \quad 1 - \left[\frac{S \cdot (1-P_n)^{\frac{1}{v}}}{1-S} \cdot \frac{t}{S} - \frac{S \cdot (1-P_n)^{\frac{1}{v}}}{1-S}\right]^{v} \tag{22}$$

$$= \quad 1 - \left(\frac{S \cdot (1-P_n)^{\frac{1}{v}} \cdot (\frac{t}{S} - 1)}{(1-S)}\right)^{v} \tag{23}$$