# PeerNet: A Peer-to-Peer Framework for Large-Scale Service and Application Deployment in MANETs

Anandha Gopalan and Taieb Znati

Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, U.S.A Email: {axgopala, znati}@cs.pitt.edu

Abstract—Ad-hoc networks are an emerging technology with enormous potential. Providing support for large-scale service and application deployment in these networks, however is crucial to make them a viable alternative. The lack of infrastructure, coupled with the time-varying characteristics of ad-hoc networks, brings about a new set of challenges to the design and deployment of applications on a large-scale. This paper addresses these challenges and presents PeerNet, a unified, overlay-based service architecture to support large-scale service and application deployment in mobile, ad-hoc environments. We discuss the main functionalities of PeerNet, describe the algorithms for resource registration and discovery, and present PILOT, a novel power-aware, location-driven traffic forwarding algorithm to enable node interaction in this architecture. We conclude the paper by providing a sensitivity analysis of the proposed framework and a comparative study of PILOT. The results show that PeerNet is scalable and robust, even when the mobility of the nodes in the network is high.

## I. INTRODUCTION

Advances in wireless technology and portable computing along with demands for greater user mobility have provided a major impetus towards development of an emerging class of self-organizing, rapidly deployable network architectures referred to as ad-hoc networks. Ad-hoc networks, which have proven 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. Despite their advantages, however, the large-scale deployment of services and applications over these networks has been lagging. This is mostly due to the lack of an efficient and scalable architecture to support the basic functionalities necessary to enable a computing model.

Several challenges must be addressed in order to develop an effective service architecture to support the deployment of applications in a scalable manner. These challenges are related to the development of several capabilities necessary to support a service architecture. These capabilities include: Resource registration and discovery, Mobile node location, and Traffic forwarding.

Node mobility, coupled with the limitation of computational and communication resources, brings about a new set of challenges that need to be addressed in order to enable an efficient, robust and scalable architecture for service deployment in ad-hoc networks. In addition to the resource information, the mobility information of a mobile node must also be stored in order to facilitate interaction between nodes. This information, however, changes dynamically, as the node moves from one location to another. Efficient mechanisms must, therefore be in place to update this information as nodes move.

The major contributions of this paper are: *PeerNet*, a novel P2P framework that allows for large-scale service and application deployment in MANETs and *PILOT*, a power-aware, location-driven traffic forwarding algorithm to support peer interaction and service provision.

PeerNet is scalable, robust and does not impose any location restriction on the resources. The basic tenet of PeerNet revolves around the concept of *zones*, *virtual residence* and *mobility profile*. Physically, a zone represents a geographical area in the network. Conceptually, however, a zone represents a "reference point" for a node to bootstrap resource discovery and enable peer interaction. Resources in the network are mapped to zones and these zones act as a liaison between the peers that own resources and peers that request resources. The zones are organized as a virtual DHT (Distributed Hash Table)-based structure that enables resource location through distributed indexing. The basic *design principle* of this scheme is to use *geographical* mapping for the DHT as opposed to *node* mapping since nodes are mobile.

The *virtual residence* of a node is the physical area where the node is most likely to be located. This is used as a *congregation* point by nodes to contact other nodes. In the case when a node moves from its virtual residence, it leaves behind its mobility information with a select set of neighboring *proxy* nodes. This information constitutes the *mobility profile* and consists of the expected direction and speed of travel and is used by other nodes to predict the current location of the mobile node.

The second contribution of this paper is PILOT, an algorithm for message forwarding that forwards traffic in a location-directed manner. To limit flooding in the network, PILOT forwards traffic in a truncated cone-shaped manner towards the destination. The intermediary node to forward traffic is chosen by using a priority-based scheme that imposes a priority on the neighboring nodes in a way, such that nodes which are more in line with the direction of the destination have higher probability to forward the message. This reduces the delay that traffic suffers on its way towards the destination. This priority is also closely tied to the residual energy-level of the intermediary node to maximize network lifetime.

PeerNet and PILOT were implemented in Glomosim and tested under various network scenarios. A sensitivity analysis was performed for PeerNet and the results showed that the framework was robust and scalable. PILOT was compared to LAR and AODV and the results showed that PILOT out-performed both LAR and AODV in most of the cases.

The rest of the paper is organized as follows: Section II details the work related to this paper while Sections III and IV detail the network characteristics used in PeerNet, the different components of PeerNet and the algorithms used. Section V details the simulations and the ensuing results and Section VI concludes the paper.

## II. RELATED WORK

LAR [1] and DREAM [2] are location-based routing protocols that rely on the fact that nodes know their location and use this information to optimize the routing protocol by sending the routing information in the direction of the destination rather than broadcasting it. We differ from [1] and [2] by not flooding the network with location updates; rather, messages are forwarded by intermediary nodes on a piece-meal basis, where the direction of forwarding is changed to suit the direction of the destination. This leads to better scalability.

[3], [4], [5] use the concept of home regions. Each node in the network is mapped to an area in the network that is designated as its home region. A mobile node updates its location information by sending updates to its home region. In our scheme, a mobile node does not keep updating its *virtual residence*, rather it leaves a trail behind that can be used by other nodes to locate it.

TWINS [6] provides an architecture for addressing and locating nodes in large networks. Twins uses a DHT-based architecture for location management in selforganizing networks. Our approach differs from TWINS, in that it does not use multiple address spaces. This overhead is avoided by directly mapping the DHT onto the physical structure of the network. Furthermore, the proposed work seamlessly incorporates a power-aware data forwarding scheme into the architecture.

CAN [7] provides a distributed, Internet-scale hash table. The network is divided into zones according to a virtual co-ordinate system, where each node is responsible for a zone in the network. Given (key,value) pairs, CAN maps the key to a point within a zone using a uniform hash function and stores the (key,value) pair at the node that owns the zone.

Geographic hash table (GHT) [8] is a distributed data centric algorithm for sensor networks. GHT works by hashing keys into geographical co-ordinates and stores the (key,value) pair at the sensor node geographically closest to the hash of its key.

The Grid location service (GLS) [9] provides distributed location information service in mobile ad-hoc networks. GLS combined with geographic forwarding can be used to achieve routing in the network. A node X "recruits" a node that is "closest" to its own ID in the ID space to act as its location server.

Similar to CAN, GHT and GLS our approach also divides the network into zones, but this division is based on the physical topology. Furthermore, the proposed work seamlessly incorporates a power-aware data forwarding scheme into the architecture.

Ekta [10] integrates DHTs into MANETs. This provides an efficient architecture for constructing distributed applications and services in MANETs. Our architecture differs from this scheme, since we do not use Pastry [11] for the purpose of our DHT. The Virtual-DHT is constructed in a manner that allows it to take advantage of the location information provided in the network. PeerNet also takes node mobility into consideration.

The schemes described in [12], [13] focus on resource discovery in MANETs. Similar to these works, PeerNet provides support for resource discovery in MANETS, but uses the concept of virtual residences to bootstrap the resource discovery process.

## **III. NETWORK CHARACTERISTICS**

The main network characteristics of PeerNet are *zones* and *virtual residences*. A zone in PeerNet represents a physical area to which resources are mapped and it acts as a "reference point" between the peers that own resources and peers that request resources. A virtual

residence of a node in PeerNet is the physical area within a zone where the node is most likely to be located and is used as a congregation point to initiate peer interaction.

## A. Zones

Consider an ad-hoc network covering a specific geographical area, denoted by  $\Lambda$ . Logically, the network area is perceived to be a collection of zones,  $Z_i$   $(1 \le i \le N)$ , such as  $\Lambda = \bigcup_i Z_i$  Each zone,  $Z_i$ , is characterized by its coverage,  $C(Z_i)$ , such as  $C(Z_i) = \int \int_{\mathcal{B}} dx \, dy$ , where  $\mathcal{B}$  is the boundary of  $Z_i$ . Resources in the network are characterized by their unique identifiers. Furthermore, each resource, R, is mapped into a zone, Z in  $\Lambda$ . This mapping is achieved using a dual-valued hash function H(), such as  $H(R) = \langle x, y \rangle$  and  $\langle x, y \rangle \in C(Z)$ . Mobile nodes within C(Z) are responsible for maintaining and managing information related to R. This responsibility includes resolving requests regarding the current location of the node which owns the resource along with any other relevant attributes.

In order to balance the distribution of resource information among the network zones, the hash function, H(), must be uniform [14]. Furthermore, it is desirable that the computational cost and collision of H() be minimal. To meet these requirements, the function H()is defined as follows:

$$H(R) = \begin{cases} H^x(R) = \lfloor L * (R * A - \lfloor R * A \rfloor) \rfloor \\ H^y(R) = \lfloor M * (R * A - \lfloor R * A \rfloor) \rfloor, \end{cases}$$
(1)

where: R is a *m*-bit resource id, 0 < A < 1 is a constant,  $\Lambda$  is a network area of size LxM and  $H^x()$  and  $H^y()$  are uniform hash functions. It is shown in [14] that a good choice for A is:  $(\sqrt{5} - 1)/(2)$ .

## B. Virtual Residences (VR)s

The *virtual residence* of a node refers to the current physical area within a zone where the node is most likely to be located. The *virtual residence* also changes over time depending on the mobility of the node. Hence, a virtual residence of a node refers to its current physical location and also reflects the user behavior. A user provides this information to the network while registering, thus facilitating location of the user.

A mobile node A, upon departure from its virtual residence enroute to another VR leaves behind information with select *proxy* nodes. This is used by other nodes to probabilistically determine the location of A to initiate interaction with A. This information is stored in the form of a vector that contains the expected direction and speed of travel of A and is called the *mobility profile* of A.

#### **IV. SYSTEM ARCHITECTURE AND SERVICES**

## A. Resource Registration

A peer A, that owns a collection of resources must register its resources with the network in order for other nodes in the network to locate its resources. A peer registers its resources with the network by first hashing the *resource id* to obtain a *hash value*. This hash value maps to the Cartesian co-ordinate of a point (P) within the coverage of a zone, (Z) in the network. A sends a message along with its *virtual residence* (VR), resource id and other attributes related to the resource to C(Z). The set of mobile nodes within the neighborhood of point P in C(Z) register this information. In the case, when there are no nodes C(Z), node A tries to register again, after a timeout. Algorithm 1 details the process by which a mobile node, A registers information relevant to resource  $R_i$ , with the network (H is the hash function).

Algorithm 1 Resource Registration		
<b>RESOURCE-REGISTER</b> $(R_i, H)$	Res	
(1) Calculate $H(R_i) = \langle x_i, y_i \rangle$	(1)	
(2) Let $P = \langle x_i, y_i \rangle, P \in \mathcal{C}(Z)$	(2)	
(3) Send a message to $\mathcal{C}(Z)$ contain-	(3)	
ing $[VR(A), R_i,$ other relevant at-		
tributes of $R_i$ ]		
(4) Nodes $\in \mathcal{C}(Z)$ in the neighborhood	(4)	
of P register $R_i$ 's information		
(5) return	(5)	

#### B. Resource Discovery

A peer A, wishing to locate the resource(s) of interest in the network, first calculates the *hash value* by hashing the *resource id*. This hash value maps to the co-ordinate of a point (P) within the coverage of a zone (Z) in the network. A sends a request to C(Z) for information about the resource. The mobile nodes responsible for holding the resource information reply with a list of peers that *own* this resource. Algorithm 2 details the process by which node A, discovers information about resource  $R_i$ .

## C. Mobile Node Location and Peer Interaction

A mechanism is needed by which node mobility can be incorporated into PeerNet. Consider the situation when a node A leaves its current *virtual residence*. Node A has some knowledge about its intended destination and its direction and speed of travel. Node A leaves behind information in the form of a *mobility profile* with select *proxy* nodes that act as the *MPMB* (Mobility Profile

## Algorithm 2 Resource Discovery

- RESOURCE-DISCOVER $(R_i, H)$
- (1) Calculate  $H(R_i) = \langle x_i, y_i \rangle$
- (2) Let  $P = \langle x_i, y_i \rangle, P \in \mathcal{C}(Z)$
- (3) Send a query to C(Z) containing  $R_i$
- (4) Nodes ∈ C(Z) in the neighborhood of P reply with a list of peers that own R<sub>i</sub>
- (5) return

Management Base). The mobility profile of A is a metric of the form  $[t_0, V(t_0), D(t_0), P_V(t), P_D(t)]$ , where:  $t_0 =$ starting time,  $V(t_0) =$  expected starting speed,  $D(t_0) =$ expected initial direction,  $P_v(t) =$  predictor for speed after t units of time since A's departure, and  $P_d(t) =$ predictor for direction after t units of time since A's departure. Nodes which wish to contact A can predict the new location of A based on its mobility profile [15].

The mobile node A needs to find the set of proxy nodes that form the *MPMB* in its zone to leave behind its mobility profile with. To recruit proxy nodes, node Asends out a broadcast message within its zone and waits for replies from the other nodes. The mobility profile is encoded in a manner such that k out of N (number of replies) fragments are enough to re-construct the original profile. This is to ensure that the mobility profile is still available even when a few proxy nodes become mobile. To allow more flexible node mobility and accommodate random mobility, a node also sends back corrections with regards to its mobility profile to the MPMB.

## D. PILOT: A Power-Aware, Location Driven Traffic Forwarding Algorithm

This section describes in detail the traffic forwarding algorithm that uses the *mobility profile* of a mobile node.

Let us consider a scenario when a node S tries to contact another node D. Using the *mobility profile* of D, node S tries to route the traffic to D. To reduce flooding in the network, the traffic is limited to a truncated, coneshaped region whose central-axis is directed towards the direction of D, as shown in Fig. 1. Nodes in region 1 have the highest probability to forward the traffic. If no nodes are currently available in region 1, the transmission area is expanded to include region 2, after a timeout. This strategy imposes a priority on neighboring nodes in way such that nodes which are more in line with the direction of the destination have higher probability to forward the message, thereby reducing the traffic delay. As the message progresses toward its destination, the node with the highest probability for forwarding the message calculates a new cone and re-iterates the process. This is shown in Algorithm 3 ( $V_L(D)$  = expected location of D, L(N) = location of N, M(D) = message).



Fig. 1. Directional Routing

#### Algorithm 3 Forwarding Messages

For	RWARD-MSG $(\alpha_n, d_n, M(D))$
(1)	Calculate $P_n$ using $\alpha_n$ , $d_n$
(2)	While (!success)
(3)	Generate a number $P$ in $[0, 1]$
(4)	if $0 \leq P \leq P_n$
(5)	Calculate $V_L(D) =$
	$[V(t_0), D(t_0), P_V(T), P_D(T)]$
(6)	Send limited-directed bcast
	of $[V_L(D), M(D), L(N)]$
(7)	success = true
(8)	else
(9)	Wait for the next time slot
(10)	if Msg-Sent by another node
	before timeout
(11)	drop request; return
(12)	else
(13)	continue
(14)	return

An important aspect of algorithm 3 is to calculate  $P_n$ , the probability of forwarding (for an intermediary node, N). This is dependent on  $\alpha_n$  (angle N makes with S) and  $d_n$  (distance of N from S). Let the angle of the truncated cone be  $\alpha_c$ . Based on Fig. 1, it is clear that, if all nodes had equal energy reserves, node E is the best node in zone 1 to forward the message and hence must have the highest probability. We must choose a node within the cone that is the farthest away from the source and is also in the direction of the destination. Furthermore, the probability function must also provide flexibility to balance each factor in the formula. Let R be the transmission range of S. The formula for calculating  $P_n$  is given by:

$$P_n = \begin{cases} w_1 * \frac{d_n}{R} + w_2 * \frac{(\alpha_c - \alpha_n)}{\alpha_c} & w_1 + w_2 \leq 1\\ 0 & d > R \text{ or } \alpha_n > \alpha_c \end{cases}$$
(2)



Fig. 2. Directional Routing as destination moves

Another aspect of PILOT is the *directional routing* that requires each node along the path to re-calculate the *cone* used to forward the message to the destination. Consider a situation, when source S wants to send a message to D. Based on the *mobility profile* of D, S now calculates the angle  $\alpha$  (shown in Fig. 2) and derives the *cone* and sends the message towards D. Node S<sub>1</sub> upon receipt of this message, calculates the angle  $\delta$  (based on the new position of D) and sends the message.

Consider part B of Fig. 2. Assume the expected direction of travel is  $\beta$  with respect to the *x*-axis and the expected speed of travel is v. Let  $D = \langle x_1, y_1 \rangle$ ,  $D_1 = \langle x_2, y_2 \rangle$  and  $S_1 = \langle sx_1, sy_1 \rangle$ .  $\delta$  can now be calculated as follows:

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

$$x_2 = x_1 + d * \cos\beta; \ y_2 = y_1 + d * \sin\beta$$
 (4)

$$R = \sqrt{(x_2 - sx_1)^2 + (y_2 - sy_1)^2} \tag{5}$$

$$\delta = \arcsin(\frac{d}{B}) \tag{6}$$

*Power-Aware Forwarding:* As discussed in section IV-D, 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 message has not been forwarded the probability of forwarding the message should increase as time elapses. The forwarding probability has to reach 1 by S time slots, where S is a design parameter.

For choosing the next relay node to forward the message to the destination, the best candidate is the node with the maximum  $P_n$ . However, the current energy level of the relay node is also important. Letting nodes deplete their energy and die may cause a network partition. We want to construct the best available route from the source to the destination while maximizing 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 faster the probability increase rate and vice versa. The *probability of forwarding* function ( $\Gamma(e, t)$ ), thus depends on the energy e and the time slot t at a node and has the following properties:

$$\Gamma(e,1) = P_n; \ \Gamma(e,S) = 1 \tag{7}$$

$$\Gamma(e+1,t) > \Gamma(e,t); \ \Gamma(e,t+1) > \Gamma(e,t)$$
(8)

Since the forwarding probability has to reach 1 by S slots, we have to derive it 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. Equation 9 has the requisite property and produces a family of curves that start at 0 and reach 1 depending on the value of e.

$$F(x) = 1 - (1 - x)^{e}, 0 < x \le 1$$
(9)

Equation 9 however, does not satisfy constraint 7. Hence, we add the following constraints and substitute  $x = \frac{t}{S}$  (t = time slot and S = max time slot).

$$F(\frac{t}{S}) = \begin{cases} P_n & t = 1\\ 1 & t = S \end{cases}$$
(10)

While obeying the constraints in equation 10, we can solve Equation 9 to get the overall probability of forwarding  $\Gamma(e, t)$ , which is given in equation 11.

$$\Gamma(e,t) = 1 - \left(\frac{S * (1 - P_n)^{\frac{1}{e}} * (\frac{t}{S} - 1)}{(1 - S)}\right)^e \quad (11)$$

where  $P_n$  = initial forwarding probability as defined by equation 2, t = elapsed time slots since the instant of message arrival. Using equation 11, we can plot  $\Gamma(e, t)$ as shown in Fig. 3 by varying the value of e from 0.25 to 4.  $P_n$  is set to 0.2.

The parameter e in equation 11 is a function of the current energy level of the node. When the node has a low energy level, e should be low, which forces the



Fig. 3. Probability of forwarding while changing the value of e

forwarding probability to increase at a slow rate. Also, when the node has a high energy level, e must be high, thus forcing the forwarding probability to increase at a higher rate. It should be noted that, for mid-range energy level the increase of the forwarding probability is almost linear with time. The parameter e can be computed from the node's current relative energy level as follows:

$$e_n = 2^{4 \cdot (E_n - 0.5)} \tag{12}$$

where  $E_n$  is the relative energy level of node n.

It should be mentioned that, although equation 11 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 and for K different energy levels and different probabilities can be computed offline and then used by the wireless node. A node chooses to use the table whose probability is closest to its own  $P_n$ . The values of both K and S can be determined at design time.

#### V. SIMULATION AND RESULTS

PeerNet was implemented in Glomosim [16] and was tested using different network scenarios. The first set of tests were conducted as part of the sensitivity analysis of the framework. The second set of tests compared the performance of PILOT to LAR [1] and AODV [17].

For the sensitivity analysis, the hit rate (amount of traffic reaching the destination), and the response time for resource registration were measured with respect to the transmission range and density of the nodes in the network. For the comparative analysis, the throughput was measured with respect to the transmission range, density and average speed of the nodes.

The number of nodes was varied from 100 to 500 and these nodes were placed in a network grid of size 2800x2800m. This grid was further divided into zones of size 400x400m. The mobility model used during the experiments was: *Random Trip* model [18]. Resource

registration was simulated by choosing a node at random and having that node register 10 "resources". For evaluating the "hit rate", we calculate the % of messages that reach the destination when the source sends a total of 10 messages; a message being sent every 10s. For the comparative analysis, traffic generated was CBR traffic with two different sources and two different destinations. The total number of messages sent during the simulations were 20 messages from each source.

1) Sensitivity Analysis: In this section, we analyze the results of the experiments performed for the sensitivity analysis of PeerNet. All experiments were conducted for low mobility (5 m/s) and high mobility (25 m/s) nodes.

The first set of experiments, depicted in figures 4 and 5 respectively, were performed to evaluate the hit-rate, while varying transmission range (number of nodes = 250) and node density (transmission range = 250)



From Fig. 4, we conclude that in all the cases the hit rate increases and reaches 100% as the transmission range grows. This can be attributed to the fact that as the transmission range grows, more nodes can be reached and hence more nodes are available to forward traffic towards the destination. The hit rate is always lower for the high mobility case as when compared to the low mobility case. This is due to the fact that the node mobility has a direct effect on zone membership.



From Fig. 5, we conclude that in all cases the hit rate increases and reaches 100% as the density of the network

grows. The hit rate is always lower for the high mobility case when compared to the low mobility case. This can be attributed to the fact that node mobility has an impact on the number of nodes in a zone that are available to hold resource information We can further observe that due to this the hit rate for a sparse network is lower.

The second set of experiments, depicted in figures 6 and 7 respectively, were performed to evaluate the response time for resource registration, while varying transmission range and node density.



Fig. 6. Effect of Transmission Range

From Fig. 6, we conclude that the response time decreases as transmission range increases. This is due to an increase in the availability of nodes due to increased transmission radius. We can also observe that the response time for a high mobility network is higher (though not by much) than in the case of a low mobility network. This is due to the fact that mobility causes frequent changes to zone membership.



Fig. 7. Effect of Node Density

From Fig. 7, we conclude that for a slightly dense to a very dense network, the response time remains almost a constant. For a sparsely populated network (100 nodes), the response time is significantly higher. This is due to the paucity of nodes in the zones in the network. We can also observe that the response time for a high mobility network is higher (though not by much) than in the case of a low mobility network. This is because node mobility causes frequent changes to the zone membership.

2) Comparative Analysis: In this section, we compare the performance of PILOT to LAR and AODV, in terms of the throughput achieved for a network consisting of mobile nodes by varying different network parameters.

The first experiment, depicted in Fig. 8 was performed to measure the impact of transmission range. The number of nodes was set to 250 and the transmission range was varied from 100 - 500m.



From Fig. 8, it can be seen that PILOT performs better than AODV, but not as well as LAR for the transmission ranges of 100 and 200. Upon increasing the transmission range ( $\geq$  300), PILOT out-performs both LAR and AODV. This is because of a 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 message towards the destination.

The second experiment, depicted in Fig. 9 was preformed to measure the impact of node density. The transmission range was set to 250 and the number of nodes was varied from 100 - 500.



From Fig. 9, we notice that PILOT performs better than both LAR and AODV. An increase in the network density leads to more nodes being available to forward the traffic towards the destination.

The final experiment, depicted in Fig. 10 was performed to measure the impact of average speed. The transmission range and number of nodes were fixed. The average speed of the nodes was varied from 10 - 50 m/s.



Fig. 10. Impact of Average Speed

Fig. 10 shows that, as the average speed of the nodes increases, the throughput goes down in all cases. This can be attributed to the increased mobility in the network. 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 forward the traffic towards the destination. As the average speed increases, the overhead associated with maintaining and discovering routes in the network increases, thus decreasing the throughput for both AODV and LAR.

## VI. CONCLUSION

The major contributions of this paper are: (a) PeerNet, a novel P2P framework for large-scale service and application deployment in MANETs and (b) PILOT, a poweraware, location driven traffic forwarding algorithm for this framework.

PeerNet is scalable, robust and does not impose any location restriction on the resources. Resources in the network are mapped to zones (using a uniform hash function) in the network. These zones are organized as a DHT structure that is directly mapped onto the physical structure of the network. The basic design principle of PeerNet is to use *geographical* mapping for the DHT as opposed to *node* mapping since nodes are mobile. Node mobility is also incorporated into PeerNet by using the *mobility profile* of a node to predict its current location.

PILOT is 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 online. PILOT was simulated using Glomosim and evaluated by providing different network scenarios. It was found that PILOT out-performed both LAR and AODV in most of the cases.

#### REFERENCES

- Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad-Hoc Networks," in *Proc. ACM/IEEE MOBICOM*, Oct. 1998.
- [2] S. Basagni, I. Chlamtac, V. R. Syrotiuk and B. A. Woodward, "A Distance Routing Effect Algirithm for Mobility (DREAM)," in *Proc. ACM/IEEE Mobicom*, Oct 1998.
- [3] I. Stojmenovic, "Home agent based location update and destination search schemes in ad hoc wireless networks," SITE, University of Ottawa, Tech. Rep. TR-99-10, Sep. 1999.
- [4] J. P. Hubaux, J. Y. Le Boudec, G. T., and V. M., "Towards selforganizing mobile ad-hoc networks: the terminodes project," *IEEE Comm Mag*, vol. 39, no. 1, pp. 118 –124, Jan 2001.
- [5] S. C. M. Woo and S. Singh, "Scalable Routing Protocol for Ad Hoc Networks," *Journal of Wireless Networks*, vol. 7, no. 5, Sep. 2001.
- [6] A. Vaina, M. Amorim, Y. Viniotis, S. Fdida, and J. Rezende, "Twins: A dual addressing space representation for selforganizing networks," *IEEE Transactions on Parallel and Distributed Systems*, 2006.
- [7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. of ACM SIGCOMM*, 2001.
- [8] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage in SensorNets," in *Proceedings of the First* ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Sept. 2002.
- [9] J. Li and J. Jannotti and D. De Couto and D. Karger and R. Morris, "A Scalable Location Service for Geographic Ad-Hoc Routing," in *Proceedings of Mobicom*, Aug. 2000, pp. 120–130.
- [10] H. Pucha, S. M. Das and Y. Charlie Hu, "Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks," in *Proc. of the 6th IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 2004.
- [11] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Lecture Notes in Computer Science*, vol. 2218, pp. 329– 350, 2001.
- [12] Jivodar B. Tchakarov and Nitin H. Vaidya, "Efficient Content Location in Mobile Ad Hoc Networks," in *Proc. IEEE International Conference on Mobile Data Management (MDM 2004)*, Jan. 2004.
- [13] U. Kozat and L. Tassiulas, "Network Layer Support for Service Discovery in Mobile Ad Hoc Networks," in *Proc. of IEEE INFOCOM*, 2003.
- [14] Donald E. Knuth, Art of Computer Programming, Volume 3. Addison-Wesley Professional, 1998.
- [15] A. R. Aljadhai and T. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments." *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1915–1930, 2001.
- [16] L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia and M. Gerla, "Glomosim: A scalable network simulation environment," UCLA, Tech. Rep. 990027, May, 1999.
- [17] 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.
- [18] J. Y. Le Boudec and M. Vojnovic, "Perfect Simulation and Stationarity of a Class of Mobility Models," in *Proc. of IEEE INFOCOM*, 2005.