PeerNet: A Peer-to-Peer Framework for 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 immense 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 new 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 MANETs. 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. We conclude the paper by comparing PILOT to LAR and AODV for a network of mobile nodes.**

I. Introduction

Advances in wireless technology 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 the advantages provided by ad-hoc networks, 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 for ad-hoc networks. These capabilities include: Resource registration, Resource discovery, Mobile Node location and Traffic forwarding.

Accommodating the mobility of nodes in the network clearly requires mechanisms and information which go beyond the information required for a typical service architecture in wired networks. In addition to the resource interface, nodes must also register their location information and their mobility information in order to facilitate interaction with other peers "anytime, anywhere". This information, however, changes dynamically, as the peers move from one location to another. Efficient mechanisms must, therefore, be in place to update this information as peers move. The goal of this paper is to address the fundamental design issues of a service infrastructure for ad-hoc networks and provide a comprehensive solution which takes into consideration node mobility and resource constraints.

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

The proposed architecture is scalable and robust and does not impose any location restrictions on the resources. The basic tenet of PeerNet revolves around the concepts of *zones*, *virtual residence* and *mobility profile*. Physically, a zone represents a geographical area in a hierarchically structured network. Conceptually however, a zone represents a "reference point" for a node to bootstrap resource discovery and peer interaction. The zones are organized as a virtual DHT (Distributed Hash Table)-based structure that enables resource location through distributed indexing. The novelty of this approach is that no specific table content is managed by the nodes. The proposed approach uses a virtual structure that is tightly coupled to the physical structure of the network to locate nodes where resource information is stored.

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 away from its virtual residence, it leaves behind its mobility information with a selected set of neighboring *proxy* nodes. This mobility information constitutes the *mobility profile* of the node; it consists primarily of the expected direction and speed of travel and is used by other nodes to predict the current location of the mobile node. The *main* advantage of this approach is that each node can choose to provide its *own* mobility prediction model, which it deems to be most appropriate to its current activity, rather than using a network-wide mobility model which may not be applicable to specific itineraries and situations.

The second contribution of this paper is an algorithm for information dissemination. This algorithm, called PILOT forwards traffic in a location-directed manner. To limit flooding in the network, PILOT uses the knowledge about the location of the source and the direction of the destination to forward 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. Consider the case of two nodes, similar with respect to their position from the source and the destination; the node with higher energy will have the higher probability to forward the message towards the destination.

The rest of this paper is organized as follows: Section II details the related work in this area while Section III details the network characteristics used in PeerNet, Section IV details the different components of PeerNet and the algorithms used. This section also includes a description of PILOT (Section IV-C), the message forwarding protocol used in PeerNet. Section V details the simulations and the ensuing results while Section VI concludes the paper and identifies areas of future work.

II. Related Work

Service discovery for ad-hoc networks is still a very new area of research. There have been some protocols for service location and discovery that have been developed for LANs, namely: Service Location Protocol (SLP) [1] and Simple Service Discovery Protocol (SSDP) [2]. SLP relies on agents to search for and locate services in the network; an *user agent* is used on behalf of users to search for services, while a *service agent* advertises services on behalf of a server and finally a *directory agent* collects the advertisements sent out by the *server agent*. SSDP uses a specific protocol and port number to search for and locate services in the network. HTTP UDP is used on the reserved local multicast address *239.255.255.250* along with the *SSDP port* while searching for services. Both SLP and SSDP cannot be directly used for MANETs due to their reliance on an existing network structure.

Ad-hoc routing protocols, in general can be classified into three categories: pro-active [3], re-active [4], [5] and hybrid [6], [7]. There is a new 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 protocols rely on the fact that the nodes in the network know their location (using a service similar to GPS [8]). This is used to optimize the routing protocol by sending the routing information in a direction that is *closer* to the destination rather than broadcasting it. Examples of location-based routing protocols are: LAR [9] and DREAM [10]. We differ 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 leads to our message forwarding algorithm being scalable when compared to DREAM and LAR.

CAN [11] (Content Addressable Network) 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 virtual zone. Given (key,value) pairs, CAN maps the key to a point P in the co-ordinate system using a uniform hash function. The corresponding (key,value) pair is stored at the node that owns the zone containing P. We differ from CAN by having more nodes in a zone to hold object information. Also, a zone is not split when a new node arrives and the overhead is avoided. Mobility is also incorporated by using the *mobility profile management base*.

The Landmark routing hierarchy [12] provides a set of algorithms for routing in large, dynamic networks. Nodes in this hierarchy have a permanent node ID and a *landmark* address that is used for routing. The *landmark* address consists of a list of the IDs of nodes along the path from this node to a well known *landmark* node. Location service is provided in the landmark hierarchy by mapping node IDs to addresses. A node X chooses its address server by hashing its node id. The node whose value matches or is closest to the hash value is chosen as X's address server. Our scheme is similar to the *landmark* scheme in terms of using the address server for a resource, but we do not rely on any one specific node to hold this information. We elect a group of nodes from within a *zone* to maintain this information.

The Grid Location Service (GLS) [13] 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. PeerNet differs from GLS and the Landmark scheme by using a group of nodes (that are selected from within a zone) to act as the object location server. The information is stored in a manner such that only a fraction of the fragments are necessary to *re-construct* it. This increases the robustness of PeerNet, since the information is still available, even after the departure or failure of some nodes in the zone.

[14], [15], [16] use the concept of home regions. Each node is mapped to an area (using a hash function) in the network that is designated as its home region. The home region holds the location information about the mobile nodes which map to this location. A node updates its location information by sending updates to its home region. In our scheme, a node does not keep updating its *virtual home*, rather it leaves a trail behind that can be used by other nodes to locate it.

Ekta [17] integrates distributed hash tables into MANETs and provides an architecture for constructing distributed applications and services. PeerNet differs by not using Pastry [18] as the DHT. The DHT is constructed in a manner that allows it to take advantage of the location information provided. PeerNet also takes node mobility into consideration.

[19], [20] provide basis for resource discovery in MANETs. Our main contribution when compared to these works is the use of a DHT-based system for providing resource discovery in MANETs.

III. Network Characteristics

Consider an ad-hoc network covering a specific geographical area, denoted by Λ. We perceive this area to be divided into zones such that $\Lambda = \bigcup_i Z_i$, where Z_i is a zone. A zone, Z_i can for example, be defined based on geographical proximity to facilitate communication between the nodes. A zone is characterized by its geographical co-ordinates, which constitutes the unique identifier of the zone.

A zone is responsible for hosting a specific set of services that are required to support interaction between peers. These services include: Registry Service: Responsible for registering information about the resources that map to the zone and Location Service: Used by nodes in the zone to acquire their geographical co-ordinates. These services are provided by mobile nodes within a zone that are selected based on their mobility and power information.

The *zone*s in the network form a virtual DHT-based distributed information base that holds resource information. Each resource in the network is characterized by a *unique* resource id. The virtual DHT maps the resource ids into zones in the network, where resource information is stored. This is achieved by *hashing* the resource id (using a system-wide hash function) to get a *hash value* that maps to the physical x and y co-ordinates of a point that falls within a zone in the network. For example, let the resource id be R_i and the hash function be H; $H(R_i) = [x_i, y_i]$. The set of nodes within the zone containing $[x_i, y_i]$ assume the responsibility of maintaining information about the resource R_i. These nodes also resolve the request for information about R_i into the virtual residence of the mobile node that owns the resource.

The novelty of this approach is that no specific DHT-routing table is managed by the nodes. The virtual DHT-structure is tightly coupled to the physical structure of the network. While bootstrapping, a node only needs to know the hash function that is used. To ensure that there are no *hot spot*s in the network due to the hashing of *id*s, the hash function is chosen to be a *uniform* hash function [21].

In case there are no nodes in the zone to hold resource information, the node wishing to register the resource, calculates a second hash value (using a second hash function) that corresponds to the Cartesian co-ordinates of another point in the network. The node again tries to register the resource in this new zone. This process continues until the node has either registered its resource, or it has tried κ number of times (κ being a system parameter) and failed. This leads to our scheme being κ-fault tolerant. Each hash function (H_i) in the set tried by the node must be an *uniform* hash function.

The *virtual residence* of a node refers to the current physical location within a zone where the node is most likely to be located. A user provides this information to the network while registering, thus facilitating location of the user. A node is uniquely characterized by its *identifier* and its *virtual residence*. During resource registration, a node registers its *virtual residence* along with its *long-term schedule*, that consists of a list of *virtual residences* that it is likely to inhabit over time. Registering the *long-term schedule* has the added advantage that a node does not need to re-register every time it moves to a new virtual residence. In the case when node does deviates from its *schedule*, it sends a correction to the zone it registered with to reflect the change in its schedule.

A mobile node A, upon departure from its virtual residence leaves behind information that is used by other nodes to probabilistically determine the location of A to initiate interaction with A. This scheme includes building the *mobility profile management base* and consists of a set of *proxy* nodes that are responsible for holding A's mobility information. The mobility information is stored in the form of a vector that contains the expected direction and speed of A and is called the *mobility profile* of A. Nodes which wish to contact node A can predict the new location of A based on its *mobility profile* and the `elapsed` time since this information was provided.

IV. System Architecture and Services

A. Resource Registration

A peer A, that manages 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 a zone (Z) in the network. A sends a message along with its *virtual residence (VR)*, *long-term schedule (LTS)*, resource id and other attributes related to the resource to Z. The set of mobile nodes within Z register this information. Algorithm 1 details the process by which a mobile node, A registers information relevant to resource R_i, with the network in PeerNet (H is the hash function). Node A uses the traffic forwarding algorithm described in Section IV-C to contact the zone where the resource is to be registered.

Algorithm 1 Resource Registration

Input: R_i, H
Output: Result
RESOURCE-REGISTER(R_i, H)
(1) Calculate $H(R_i) = [x_i, y_i]$
(2) Let $[x_i, y_i]$ be the Cartesian co-ordinates of a point within a zone, Z
(3) Use directional routing to send a message to Z containing [$VR(A)$, $LTS(A)$, R_i, other relevant attributes of resource, if necessary]
(4) Nodes in Z register the information about the resource associated with R_i
(5) **return**

B. Resource Discovery

Resource discovery uses a procedure similar to resource registration. 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 Cartesian co-ordinate of a point (P) within a zone (Z) in the network. A sends a request to the co-responding zone for information about the resource. The mobile nodes responsible for holding the resource information reply with a list of peers that *own* this resource. Node A can then choose to select a particular peer and this decision may be made based upon factors like past history, distance to the current *virtual residence* of the peer (can be inferred from the *long-term schedule* of the peer) and the stability of the peer. Once node A decides on a particular peer, it can use the knowledge about the VR of the peer and the traffic forwarding algorithm described in Section IV-C to interact with the desired peer. Algorithm 2 details the process by which a node A, discovers information about a resource R_i in PeerNet.

Algorithm 2 Resource Discovery

Input: R_i, H
Output: Result
RESOURCE-DISCOVER(R_i, H)
(1) Calculate $H(R_i) = [x_i, y_i]$
(2) Let $[x_i, y_i]$ be the Cartesian co-ordinates of a point within a zone, Z
(3) Use directional routing to send a query to Z containing R_i
(4) Nodes in Z reply with a list of peers that *own* the resource associated with R_i
(5) **return**

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

This section gives a brief description of the traffic forwarding algorithm used in PeerNet. Consider the scenario when a source S attempts to route traffic to a destination D and D is not present in its virtual residence. Using the *mobility profile* of D, S locates D and routes traffic to it.

To limit flooding in the network, PILOT uses the knowledge about the location of the source (S) and the direction of the destination (D) to forward traffic in a truncated cone-shaped manner towards the destination (shown in Fig. 1). Nodes in zone 1 have the highest priority to forward the traffic, while the nodes in zone 2 have a lower priority. If no nodes are currently available in zone 1, the transmission area is expanded to include zone 2, after a timeout. This strategy 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 energy-level of the intermediary node to maximize network lifetime. Consider two nodes, similar with respect to their position from S and D; the node with higher energy will have the higher probability to forward the message towards D. The probability of forwarding is given by the formula in equation 1.

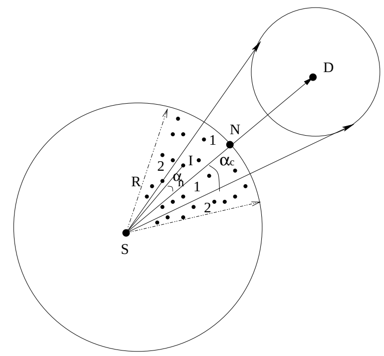

Fig. 1. Directional Routing

$$P_n = w_1 * d_n/R + w_2 * (\alpha_c - \alpha_n)/\alpha_c + w_3 * \tau(n) \quad (1)$$

where: α_c = angle of the truncated cone, α_n = angle of deviation of intermediary node (I) with S, d_n = distance of intermediary node from S, R = transmission range of S, $\tau(n)$ = residual power at the intermediary node, w_1, w_2 and w_3 are weights such that, $w_1 + w_2 + w_3 \leq 1$; $P_n = 0$, $d > R$ *or* $\alpha_n > \alpha_c$. Notice that the value of P_n is highest for a node N (as illustrated in Fig. 1) whose α_N = 0 and $d_N = R$.

To calculate P_n, an intermediary node needs to calculate α_n. An intermediary node only needs to know its relative position with respect to the source and hence, does not require the use of GPS [8]. Using a local co-ordinate system such as the one in [22], the intermediate node can calculate its relative position to S.

Consider the scenario when S sends a message to D using this approach. The nodes that receive the message sent by S calculate their probabilities and based on this information, they either listen or forward the message. Upon hearing a message, an eligible node uses the above probability to decide if it should forward the message. Furthermore, upon hearing a transmission within the zone, the remaining eligible nodes drop the message. As the message progresses toward its destination, the highest probability node responsible for forwarding the message calculates a new cone and re-iterates the process. This algorithm is shown as a *pseudo code* in algorithm 3.

V. SIMULATION AND RESULTS

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

The protocol was implemented in the Glomosim network simulator [23] on Linux and was tested by providing different network scenarios. The tests compared the performance of PILOT to LAR [9] and AODV [5]. We chose to compare this protocol to LAR and AODV because, LAR is also an

Algorithm 3 Forwarding Messages

FORWARD-MSG(α_n, $d_n, \tau(n)$)
(1) Calculate P_n using α_n, $d_n, \tau(n)$
(2) success = $false$
(3) While (!success)
(4) Generate a random number P in $[0, 1]$
(5) **if** $0 \le P \le P_n$
(6) Forward Msg
(7) success = $true$
(8) **else**
(9) Wait for the next time slot
(10) **if** Msg-Sent by another node before timeout
(11) drop request
(12) success = $true$
(13) **else**
(14) **continue**

on-demand location-based routing protocol and AODV is an on-demand routing protocol of a different nature.

The mobility model used during the experiments was the *Random Trip* model [24]. The most commonly used mobility model for wireless networks is the Random Waypoint model, which is easy to simulate but does not produce realistic scenarios [25]. We use the *Random Trip* mobility model because it is a generic mobility model that achieves realistic scenarios. For the comparative analysis with LAR and AODV, the throughput was measured while varying the node density, transmission range 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. The network simulated was thus varied from a sparsely populated network to a densely populated network. The network grid was further divided into zones of size 400x400m. To observe the effect of transmission range, the transmission range of the nodes was varied from 100 to 500m. To observe the effect of the average speed of the mobile nodes, the average speed was varied from 10 m/s to 100 m/s in steps of 10 m/s.

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 for packets to reach the destination. The total number of packets sent from each source was 20. Each experiment was run 5 times and results averaged.

The first experiment, depicted in Fig. 2 was preformed to measure the impact of node density on the throughput. For this experiment, the transmission range is not varied. The average speed of the nodes was set to 25 m/s. and the number of nodes in the network was varied from 100 - 500.

From Fig. 2, we notice that PILOT performs better than both LAR and AODV. Due to node mobility, routes that were discovered by LAR at the beginning of the simulation

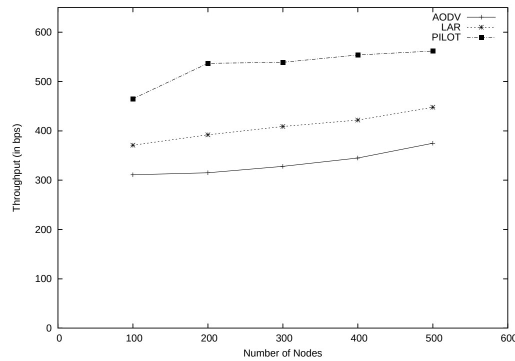

Fig. 2. Throughput vs Node Density

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 its maximum throughput. The denser the network becomes, the better PILOT performs due to the availability of more nodes that can forward the packet towards the destination.

The second experiment, depicted in Fig. 3 was performed to measure the impact of the transmission range of the nodes in the network on the throughput. The number of nodes for this experiment was not varied and is set to 250. The average speed of the nodes was set to 25 m/s and the transmission range was varied from 100 - 500m.

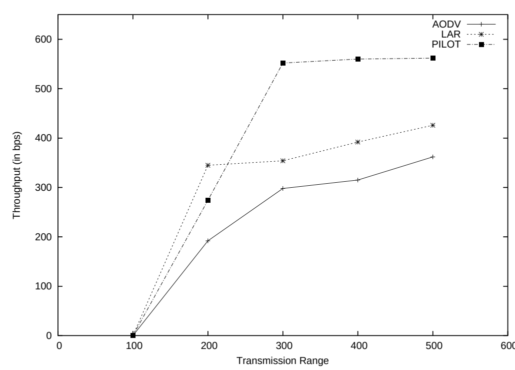

Fig. 3. Throughput vs Transmission Range

From Fig. 3, 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 packet towards the destination.

The final experiment, depicted in Fig. 4 was performed to measure the impact of average speed on the throughput. In this experiment, the transmission range and number of nodes are not varied. The number of nodes is set to 250. The average speed of the nodes was varied from 10 m/s to 100 m/s.

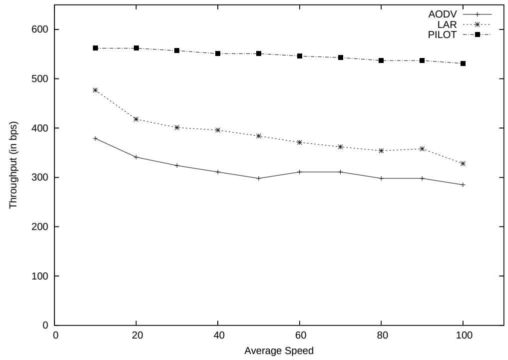

Fig. 4. Throughput vs Average Speed

Fig. 4 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 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 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 and Future Work

The major contributions of this paper are: (a) PeerNet, a peer-to-peer framework for service and application deployment in MANETs and (b) PILOT, a power-aware, location driven traffic forwarding algorithm for this framework.

PeerNet is scalable, robust and efficient. It does not require nodes to maintain routing information. For initializing, a node only needs to know its zone's *location service*. Resource registration and discovery are achieved by hashing the *resource id* to obtain the physical co-ordinates of a *point* (P) in the network. The set of nodes within the zone containing P assume the responsibility of maintaining information about the resource whose id *hashes* to P.

PeerNet is scalable, since the hash function ensures that the information stored in the network is spread across the network. Node mobility is incorporated into PeerNet by using the *long-term schedule* and the *mobility profile* to predict the location of the mobile node.

PILOT is power-aware and the probability of forwarding is closely tied to the current energy level at the node. PILOT was implemented in the Glomosim network simulator and evaluated under different network scenarios. It was found that PILOT out-performed both LAR and AODV in most of the cases, especially when the density of the network is high.

There is potential for future work in this area. Security needs to be incorporated into this service-architecture at various levels, be it resource registration, resource discovery or traffic forwarding. Also, the resource and node location information stored in the network must be protected.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments. This work has been supported by the National Science Foundation (NSF) (under grant no: ANI-000073972).

References

[1] E. Guttmann and C. Perkins and J. Veizades and M. Day, "Service Location Protocol," 1999.
[2] Yaron Y. Goland, Ting Cai, Paul Leach, Ye Gu, "Simple Service Discovery Protocol/1.0," *Internet Draft*, Oct. 1999.
[3] C. R. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," in *ACM SIGCOMM*, Oct. 1994.
[4] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks," in *Mobile Computing*. Kluwer Academic Publishers, 1996, vol. 353.
[5] 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.
[6] 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.
[7] Z. J. Haas and M. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *Internet Draft*, Aug. 1998.
[8] E. E. D. Kaplan., *Understanding GPS: principles and applications*. Artech House, Boston, MA, 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] 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.
[11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A scalable content-addressable network," in *Proc. of ACM SIGCOMM*, 2001.
[12] Paul F. Tsuchiya, "The Landmark Hierarchy: A New Hierarchy for Routing in very large Networks," in *Proc. ACM SIGCOMM*, Aug. 1988.
[13] 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.
[14] I. Stojmenovic, "Home agent based location update and destination search schemes in ad hoc wireless networks," Computer Science, SITE, University of Ottawa, Tech. Rep. TR-99-10, Sep. 1999.
[15] J. P. Hubaux, J. Y. Le Boudec, G. T., and V. M., "Towards self-organizing mobile ad-hoc networks: the terminodes project," *IEEE Comm Mag*, vol. 39, no. 1, pp. 118 –124, Jan 2001.
[16] S. C. M. Woo and S. Singh, "Scalable Routing Protocol for Ad Hoc Networks," *Journal of Wireless Networks*, vol. 7, no. 5, Sep. 2001.
[17] 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 (WMCSA 2004)*, Dec. 2004.
[18] 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.
[19] 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.
[20] U. Kozat and L. Tassiulas, "Network Layer Support for Service Discovery in Mobile Ad Hoc Networks," in *Proc. of IEEE INFOCOM*, 2003.
[21] Donald E. Knuth, *Art of Computer Programming, Volume 3*. Addison-Wesley Professional, 1998.
[22] S. Capkun, M. Hamdi and J. P. Hubaux, "GPS-Free Positioning in Mobile ad-hoc Networks," in *HICSS*, 2001.
[23] L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia and M. Gerla, "Glomosim: A scalable network simulation environment," UCLA, Tech. Rep. 990027, May, 1999.
[24] J. Y. Le Boudec and M. Vojnovic, "Perfect Simulation and Stationarity of a Class of Mobility Models," in *Proc. of IEEE INFOCOM*, 2005.
[25] J. Yoon and M. Liu and B. Noble, "Random Waypoint Considered Harmful," in *Proc. of IEEE INFOCOM*, 2003.