

11-1-2002

# Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks

Y. Charlie Hu

Saumitra M. Das

Himabindu Pucha

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

---

Hu, Y. Charlie ; Das, Saumitra M. ; and Pucha, Himabindu , "Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks" (2002). *ECE Technical Reports*. Paper 167.  
<http://docs.lib.purdue.edu/ecetr/167>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

# Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks

Y. Charlie Hu  
Saumitra M. Das  
Himabindu Pucha

TR-ECE 02-07  
November 1, 2002

School of Electrical and Computer Engineering  
1285 Electrical Engineering Building  
Purdue University  
West Lafayette, IN 47907-1285

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	DSR . . . . .	3
2.2	Pastry . . . . .	4
<b>3</b>	<b>DPSR Design</b>	<b>6</b>
3.1	Basic Design . . . . .	6
3.2	Dealing with Mobility: Route Maintenance . . . . .	7
3.3	Optimizations . . . . .	8
3.4	Discussions . . . . .	8
<b>4</b>	<b>Other Related Work</b>	<b>10</b>
<b>5</b>	<b>Status</b>	<b>11</b>
<b>6</b>	<b>Acknowledgment</b>	<b>12</b>
	<b>Bibliography</b>	<b>13</b>

## Abstract

*We argue that there exists a synergy between peer-to-peer (p2p) overlay networks for the Internet and mobile ad hoc networks (MANET) connecting mobile nodes communicating with each other via multi-hop wireless links – both share the key characteristics of self-organization and decentralization, and both need to solve the same fundamental problem, that is, how to provide connectivity in a completely decentralized environment. We propose Dynamic P2P Source Routing (DPSR), a new routing protocol for MANETs that exploits the synergy between P2P and MANET for increased scalability. By integrating DSR and a topology-aware structured P2P overlay routing protocol, DPSR limits the number of the source routes that each node has to discover and rediscover to  $O(\log N)$ , while retaining all the attributes of DSR for dealing with the specifics of ad hoc networks. This is in contrast to the potentially unlimited number of source routes each node has to maintain in DSR. Thus DPSR has potential to be more scalable than previous routing protocols for MANETs, such as DSR and AODV.*

*In addition to being a network layer multi-hop routing protocol, DPSR simultaneously implements a distributed hash table (DHT) in MANET; it implements the same functionalities as CAN, Chord, Pastry, and Tapestry, and such functionalities can be exposed to the applications built on top of it via a set of common p2p APIs.*

**Key Phrases:** Peer-to-peer computing, mobile ad hoc networks, multi-hop routing.

# 1. Introduction

A peer-to-peer (p2p) overlay network consists of a dynamically changing set of nodes connected via the Internet (i.e., IP). A mobile ad hoc network (MANET) consists of mobile nodes communicating with each other using multi-hop wireless links. P2p overlays and MANETs share the key characteristics of self-organization and decentralization. These common characteristics lead to further similarities between the two types of networks: (i) Dynamic network topology. Both have a flat and frequently changing topology, caused by node join and leave in p2p overlays and MANETs and additionally terminal mobility of the nodes in MANETs; and (ii) Hop-by-hop connection establishment. Per-hop connections in p2p are typically via TCP links with physically unlimited range, whereas per-hop connections in MANETs are via wireless links, limited by the radio transmission range. Unstructured p2p networks such as Gnutella share additional similarities with MANETs such as (iii) flooding-based routing protocols and (iv) limited scalability due to bandwidth consuming traffic from flooding.

The common characteristics shared by p2p overlays and MANETs also dictate that both networks are faced with the same fundamental challenge, that is, how to provide connectivity in a completely decentralized environment. Thus, there exists a synergy between these two types of networks in terms of the design goals and principles of their routing protocols; both P2P overlays and MANETs routing protocols have to deal with dynamic network topologies due to membership changes or mobility.

We argue that a promising research direction in networking is to exploit the synergy between p2p overlay and MANET routing protocols to design better routing protocols for MANETs. As a supporting example, in this paper, we apply recent advancement in p2p overlay networks, i.e., topology-aware structured p2p overlay routing protocols, to routing in MANETs, and propose a new routing protocol that promises to be more scalable than previous MANET routing protocols.

The primary challenge with using a p2p routing protocol in MANETs is the fact that p2p overlays in the wired Internet rely on the IP routing infrastructure to perform hop-by-hop routing between neighboring nodes in the overlays, whereas such an infrastructure does not exist in MANETs, i.e., in the link layer. The obvious idea of overlaying a p2p network on top of a multi-hop routing protocol is conceptually superfluous (and thus inefficient), as either of the p2p and the multi-hop routing protocols is already peer-to-peer in nature, and has mechanisms and incurs overheads dealing with self-organization and decentralization. Instead, our proposed new routing protocol for MANETs, Dynamic P2P Source Routing protocol (DPSR), seamlessly integrates functions performed traditionally by p2p overlay routing protocols operating in a logical namespace and by MANET routing protocols op-

erating with a physical namespace. Specifically, DPSR integrates DSR (Dynamic Source Routing protocol) [10, 11] and a topology-aware structured p2p overlay routing protocol. The key idea of the integration is to bring the structured p2p routing protocol to the network layer of MANETs via a one-to-one mapping between the IP addresses of the mobile nodes and their nodeIds in the namespace, and replacing each routing table entry which used to store a (nodeId, IP address) pair with a (nodeId, source route) pair. With this integration, DPSR limits the number of the source routes that each node has to discover and rediscover to  $O(\log N)$ , while retaining all the attributes of DSR for dealing with the specifics of ad hoc networks, i.e., due to wireless transmissions. Compared to the potentially unlimited number of source routes each node has to maintain in DSR, the bounded number of source routes managed by each node in DPSR has the potential to make DPSR much more scalable than previous routing protocols for MANETs, such as DSR and AODV.

## 2. Background

DPSR is based on the DSR protocol for MANETs and a structured p2p overlay routing protocol. In the following, we give a brief overview of DSR and Pastry, one of the structured p2p overlay routing protocols.

### 2.1 DSR

DSR [10, 11] is a representative multi-hop routing protocol for ad hoc networks. It is based on the concept of source routing in contrast to hop-by-hop routing. It includes two mechanisms, *route discovery* and *route maintenance*, which work together to allow the discovery and maintenance of source routes in ad hoc networks.

Route discovery is the process by which a source node discovers a route to a destination for which it does not already have a route in its cache. The process broadcasts a route request (RREQ) packet that is flooded across the network in a controlled manner. In addition to the address of the original initiator of the request and the target of the request, each RREQ packet contains a route record, which records the sequence of hops taken by the RREQ packet as it propagates through the network. RREQ packets use sequence numbers to prevent duplication. The request is answered by a route reply (RREP) packet from either the destination node or any intermediate node that has a cached route to the destination. To reduce the cost of the route discovery, each node maintains a cache of source routes that have been learned or overheard, which it uses aggressively to limit the frequency of propagation of route requests.

The route maintenance procedure monitors the operation of the route and informs the sender of any routing errors. A host detects transmission of corrupted or lost packets by means of passive acknowledgments, i.e., after a host sends the packet to the next hop, it overhears whether the next hop further forwards the packet. If the route breaks due to a link failure, the detecting host sends a route error (RERR) packet to the source which upon receiving removes all routes in the host's cache that use the hop in error.

Optimizations suggested for DSR include (1) modification of route cache based on forwarded source routes, (2) inclusion of a delay period proportional to the number of hops to the destination to increase the probability that the source receives the shortest route first, (3) proactive prevention of cycles in the path by source route comparisons, (4) piggy-backing of data with the RREQ to prevent latency, (5) eavesdropping on RERR packets being sent to other hosts to delete stale routes, and (6) using an exponential back-off to limit the rate at which new route discoveries are initiated. This is to prevent a large number of unproductive requests in case of partitioned networks (i.e., destination may be unreachable).

Comparison studies of DSR with other proposed routing protocols for MANET [4, 7] have shown that DSR exhibits good performance at all mobility rates. Routing overhead due to source routing is high in DSR but its performance in a high mobility scenario offsets this overhead.

**Scalability** Although DSR is one of the leading MANET routing protocols, ad hoc networks constructed using DSR are still far from scalable when compared to the “fixed” Internet.<sup>1</sup> Simulations performed in ad hoc network protocol studies such as [4, 7] have been limited to networks of around 100 nodes. The fundamental reason for the limited scalability of such protocols is that any ad hoc network routing protocol has to pay a high overhead dealing with the dynamic network topology and the shared media access of wireless communication. In particular, the size of the source route cache in a DSR node is proportional to the number of distinct destination nodes to which it has to send messages, and thus is potentially as high as  $N$ , the size of the network. Note the memory required to store such source routes is not a scalability concern. Rather, it is the overhead required to discover and rediscover these many source routes that is limiting the scalability of DSR.

The DPSR protocol proposed in this paper aims at improving the scalability of current state-of-the-art protocols by exploiting the synergy between p2p and MANET routing protocols. The key idea is to reduce the protocol overhead by limiting the number of source routes each node has to maintain.

## 2.2 Pastry

Pastry [20] is a scalable, fault resilient and self-organizing peer-to-peer substrate. Each Pastry node has a unique, uniform, randomly assigned `nodeId` in a circular 128-bit identifier space. Given a message and an associated 128-bit key, Pastry reliably routes the message to the live node whose `nodeId` is numerically closest to the key. Moreover, each Pastry node keeps track of its neighboring nodes in the namespace and notifies applications of changes in the set. We briefly review Pastry’s routing protocol below.

In a Pastry network consisting of  $N$  nodes, a message can be routed to any node in less than  $\log_{2^b} N$  steps on average ( $b$  is a configuration parameter with typical value 4), and each node stores only  $O(\log N)$  entries, where each entry maps a `nodeId` to the associated node’s IP address. Specifically, a Pastry node’s routing table is organized into  $\lceil \log_{2^b} N \rceil$  rows with  $(2^b - 1)$  entries each. Each of the  $(2^b - 1)$  entries at row  $n$  of the routing table refers to a node whose `nodeId` shares the first  $n$  digits with the present node’s `nodeId`, but whose  $(n + 1)$ th digit has one of the  $(2^b - 1)$  possible values other than the  $(n + 1)$ th digit in the present node’s id. In addition to a routing table, each node maintains a leaf set, consisting  $L/2$  node with numerically closest larger `nodeIds`, and the  $L/2$  nodes with numerically closest smaller `nodeIds`, relative to the present node’s `nodeId`.  $L$  is an even integer parameter with typical value 16.

In each routing step, the current node normally forwards the message to a node whose `nodeId` shares with the key a prefix that is at least one digit (or  $b$  bits) longer than the

---

<sup>1</sup>We note that position-based routing protocols which rely on global position systems can be more scalable than topology-based protocols such as DSR. See Section 4 for more detail.

prefix that the key shares with the current nodeId. If no such node is found in the routing table, the message is forwarded to a node whose nodeId shares a prefix with the key as long as the current node, but is numerically closer to the key than the current nodeId. Such a node must exist in the leaf set unless the nodeId of the current node or its immediate neighbor is numerically closest to the key, or  $L/2$  adjacent nodes in the leaf set have failed concurrently.

**Node join** An arriving node with a newly chosen nodeId  $X$  initializes its state by contacting a nearby node  $A$  (according to the proximity metric) and asking  $A$  to route a special message with  $X$  as the key. This message is routed to the existing node  $Z$  whose nodeId is numerically closest to  $X$ .  $X$  then obtains the leaf set from  $Z$ , and the  $i$ th row of the routing table from the  $i$ th node encountered along the route from  $A$  to  $Z$ . One can show that with this scheme,  $X$  can correctly initialize its state and notify nodes that need to know of its arrival.

Pastry is one of several topology-aware structured p2p routing protocols [5]. Although it is chosen for the design of DPSR in this paper, other structured p2p protocols such as CAN [19], Chord [21], and Tapestry [22] could potentially be used as well.

## 3. DPSR Design

Like DSR, DPSR is proposed as a network layer protocol. Message destinations and nodes are addressed using IP addresses. DPSR seamlessly integrates DSR and Pastry at the network layer for MANET routing. The key aspects of DPSR are (1) the use of a modified Pastry overlay routing in the network layer, (2) retention of DSR’s source routing mechanism, (3) inclusion of DSR’s on-demand behavior into originally strict Pastry routing semantics, and (4) inheritance of optimizations from both p2p (i.e., Pastry) and MANET (i.e., DSR) protocols.

### 3.1 Basic Design

**Mapping** DPSR assigns unique nodeIds to nodes in a MANET as is done in Pastry. To provide the same IP routing API as in DSR, nodeIds are generated by hashing the IP addresses of the hosts, e.g., using SHA-1 [8], thus obtaining a unique nodeId for each node in the network.

**Routing structure** The structures of the routing table and the leaf set stored in each DPSR node are similar to those in Pastry. The only difference lies in the content of each routing table entry. In Pastry, the routing table entry for a nodeId  $x$  contains the IP address of  $x$  which is used to route messages to node  $x$  using the IP routing infrastructure in the Internet. Since there is no such routing infrastructure in MANETs, each entry in the DPSR routing table stores the source route to reach the designated nodeId, as shown in Table 3.1. For simplicity, we also store the IP address of the destination node. As in Pastry, the routing table entry for any node  $k$  is chosen such that it is physically closer than the other choices for that routing table entry. This is achieved via a similar node join process as in Pastry, as explained below.

**Node join** The DPSR node join process is similar to that of Pastry. The differences are in constructing the contents (i.e., source routes) of the routing table and leaf set entries. (i) After the routing table has been initialized, i.e., with proper nodeIds, the source routes are invalid as they start with other nodes. The new source routes for the routing table entries can be constructed *on-demand*, using DSR’s route discovery mechanism. The idea is that the node needs not to evaluate the source route to a node  $k$  in the routing table until it is required to do so, e.g., when a message passing through it contains  $k$  as the destination nodeId, or when the node itself generates messages destined for node  $k$ . This *on-demand* or lazy discovery of source routes for routing table entries should in principle reduce the routing overhead in DPSR as it does in DSR. (ii) The source routes for entries

Destination	IP Address	Source Route
$\langle nodeId_x \rangle$	$\langle IP_x \rangle$	$\langle S_i \dots S_x \rangle$

**Table 3.1. DPSR routing table entry**

in new node’s leaf set  $L$  are discovered eagerly. This is necessary so that all nodes in the leaf set are informed about the new node arrival; up-to-date leaf sets in Pastry (and DPSR) guarantee correct routing. Note that this route discovery is done only at the node joining time. Route discovery for subsequent changes to the nodeIds in the leaf set is performed on-demand.

**Routing** The routing in DPSR is similar to that in Pastry; a message key is generated by hashing the destination IP address, and the message is routed to the node whose nodeId is numerically closest to the message key. In DPSR, since both message keys and nodeIds are hashed from IP addresses, an exact match between a message key and an destination node’s nodeId is guaranteed, i.e., a message delivery is guaranteed, if the destination node is reachable via the wireless links. The only different between DPSR and Pastry routing is that each hop in the nodeId space in DPSR is a multi-hop source route, whereas each hop in the nodeId space in Pastry is a multi-hop Internet route.

**Node failure or out of reach** Node failure is again handled similarly as in Pastry. In Pastry, if a node is not reachable, it is presumed to have failed. To replace a failed node in the leaf set, its neighbor in the nodeId space contacts the live node with the largest index on the side of the failed node, and asks that node for its leaf set. This set only partly overlaps with the present node’s leaf set. Among these new nodes, the appropriate ones are then chosen and inserted into the leaf set. In DPSR, a node could become unreachable via a source route for two reasons: it or other node(s) along the source route have either crashed, or have moved out of the range of its (their) adjacent nodes in the source route. In either case, a route rediscovery for that node is invoked. If the route rediscovery still does not find a new source route to the unreachable node, the node is replaced in the leaf set in a similar way as in Pastry. After the replacement, source routes for the new node in the leaf set are again discovered eagerly.

## 3.2 Dealing with Mobility: Route Maintenance

A unique challenge faced by routing in MANETs is to deal with the mobility of participating nodes. Due to mobility, the network topology changes continuously, bringing certain nodes closer and pushing some other nodes away from any particular node. Although route discovery is used to repair source routes as nodes move, routing as per the original routing table will now prove inefficient as a physically closer node can now be found to replace the node which has moved further away. To prevent a deterioration of the routing table quality, DPSR adopts the *routing table maintenance* mechanism used in Pastry [6]<sup>1</sup> to

---

<sup>1</sup>Routing table maintenance in Pastry is used to prevent routing table deterioration due to node arrival, departure, and failure.

constantly update the routing table entries to improve their proximity to the current node. For each row of a node’s routing table, the node selects a random entry in the row, and requests from the associated node a copy of that node’s corresponding routing table row. Each entry in that row is then compared to the corresponding entry in the local routing table. If they differ, the node estimates the distance to both entries and installs the closer entry in its own routing table. The closeness of entries is estimated from the length of the source route path to the nodeIds referred to in those entries.

The idea of maintaining 3-D routing tables in Pastry can also be used in DPSR to battle mobility. Whenever a DPSR node replaces a node in a routing table entry because a closer candidate is found, the previous node is kept in a list of alternate nodes. This gives rise to a stack of entries for each entry in the routing table, and thereby a three-dimensional organization of the routing table. When the primary node in each entry is unreachable, one of the alternates is used until and unless a closer entry is found during the next periodic routing table maintenance.

### 3.3 Optimizations

DPSR inherits all the optimizations mentioned for DSR in Section 2.1. We plan to study additional optimizations that exploit the hybrid nature of DPSR. For example, one possible optimization can be derived from the inherent broadcast nature of the wireless medium; destination nodes which are one hop away (i.e., within the transmission range) from the source node can be detected by the source node and routed to without using the routing table.

### 3.4 Discussions

We qualitatively compare the scalability of DPSR with DSR. As described in Section 2.1, using DSR, depending on the amount of mobility and the number of distinct destinations a node is sending messages to, each node needs to maintain up to  $N$  source routes in a MANET of  $N$  nodes. In contrast, using DPSR, the number of source routes each node needs to maintain is limited to  $O(\log N)$ , independent of the number of different destinations that node has to send messages to. However, the tradeoffs between DPSR and DSR is more complicated since both discover and rediscover source routes on-demand. For example, in sending  $X$  messages from a fixed starting node, in DSR that node may need to rediscover  $Y$  source routes ( $Y < X$ ), whereas in DPSR, a total of  $Z$  source routes (starting from different nodes along the path in the namespace) may need to be rediscovered. Thus the tradeoffs between DPSR and DSR boils down to the relative values of  $X$ ,  $Y$ , and  $Z$ . We plan to perform extensive simulations to study these tradeoffs.

Note that routing in DSR takes the shortest path, whereas routing in DPSR takes several hops in the namespace. Thus if the overhead of route discoveries are discounted, the routing delay using DPSR is expected to be 40% longer than using DSR, analogous to the average overhead of a Pastry routing path compared to the direct underlying Internet path using a sphere model [6], which is similar to the random waypoint model [10] typically used in MANET protocol studies. This routing delay stretch suggests that DSR may outperform DPSR in relatively static ad hoc networks that have very low route discovery overhead.

We will experiment with varying network load, mobility, and network size in studying the tradeoffs between DPSR and DSR.

## 4. Other Related Work

PeerNet [9] is a p2p-based network layer similar to DPSR in that both aim at improving the scalability of routing protocols by bringing the p2p concept from the application layer down to the network layer. However, PeerNet focuses on dynamic networks with pockets of wireless connectivities interconnected with wired lines, whereas DPSR focuses on wireless ad hoc networks.

In addition to DSR, AODV [17, 18], DSDV [16], and TORA [15] also belong to the category of *topology-based* multi-hop ad hoc routing protocols which assume no knowledge of the mobile nodes' positions. Such position information typically require the assistance of global positioning systems. In contrast, *position-based* protocols forward packets based on the physical positions of nodes. These include “flooding-based” such as LAR [13] and DREAM [1], “graph-based” such as RGD [2], and “geographic-based” such as GPSR [12]. Among these, geographic forwarding approaches route packets based on only local decisions, and thus have less overhead and are more scalable. GLS [14] is a scalable distributed location service that can be combined with geographic forwarding to construct large ad hoc networks.

## 5. Status

We are currently evaluating the DPSR protocol using ns-2 [3], which has been extended for studying multi-hop ad hoc networks [4]. We also plan to perform a detailed comparison of DPSR with other representative routing protocols for multi-hop ad hoc networks. ns-2 currently supports the following ad-hoc routing protocols: TORA, DSDV, DSR, and AODV. We will use the following metrics in the comparison: (i) Packet delivery fraction – the fraction of data packets successfully delivered to the destinations; (ii) Average delay caused by buffering during route discovery, queuing at the interface queue, retransmission at the MAC, and propagation and transfer times; (iii) Routing overhead in terms of the number of routing packets transmitted per data packet delivered to the destination; (iv) Route acquisition time – the time required to establish route(s) when requested, and (v) Path optimality – the ratio between the number of hops a packet takes to reach its destination and the length of the shortest path that physically existed through the network when the packet was originated.

## 6. Acknowledgment

This research was supported by a 2002 Honda Initiation Grant from Honda R&D Americas.

# Bibliography

- [1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proc. 4th Annual International Conference on Mobile Computing and Networking (Mobicom'98)*, October 1998.
- [2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM)*, August 1999.
- [3] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in network simulation. *IEEE Computer*, 33(5):59–67, May 2000.
- [4] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 98)*, October 1998.
- [5] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Exploiting network proximity in distributed hash tables. In O. Babaoglu, K. Birman, and K. Marzullo, editors, *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, pages 52–55, June 2002.
- [6] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Exploiting network proximity in peer-to-peer overlay networks. Technical report, Technical report MSR-TR-2002-82, 2002, 2002. <http://research.microsoft.com/~antr/PAST/localtion.ps>.
- [7] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, March 2000.
- [8] FIPS 180-1. Secure Hash Standard. Technical Report Publication 180-1, Federal Information Processing Standard (FIPS), NIST, US Department of Commerce, Washington D.C., April 1995.
- [9] S. K. Jakob Eriksson, Michalis Faloutsos. Peernet: Pushing peer-to-peer down the stack. In *IPTPS'03*, February 2003.
- [10] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic, 1996.
- [11] D. B. Johnson, D. A. Maltz, Y.-C. Hu, , and J. G. Jetcheva. The dynamic source routing protocol for mobile ad hoc wireless networks, November 2001.
- [12] B. Karp and H. Kung. GPSR: Greedy perimeter stateless terminode routing. In *Proc. ACM/IEEE MobiCom*, August 2000.
- [13] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proc. 4th Annual International Conference on Mobile Computing and Networking (Mobicom'98)*, October 1998.

- [14] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A Scalable Location Service for Geographic Ad Hoc Routing. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 120–130, Boston, Massachusetts, August 2000.
- [15] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM*, 1997.
- [16] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *Proceedings of ACM SIGCOMM*, August 1994.
- [17] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proc. of the 2nd IEEE workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.
- [18] C. Perkins, E. Royer, and S. Das. Ad hoc de demand distance vector (AODV) routing, June 1999.
- [19] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proceedings of ACM SIGCOMM*, August 2001.
- [20] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.
- [21] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proceedings of ACM SIGCOMM*, San Diego, California, August 2001.
- [22] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An Infrastructure for Fault-Resilient Wide-area Location and Routing. Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001.