USENIX Association

Proceedings of HotOS IX: The 9th Workshop on Hot Topics in Operating Systems

Lihue, Hawaii, USA
May 18–21, 2003

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

Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha
Purdue University
West Lafayette, IN 47907
{ychu, smdas, hpucha}@purdue.edu

Abstract

We argue that there exists a synergy between peer-to-peer (p2p) overlay networks for the Internet and mobile ad hoc networks (MANETs) 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 decentralized, dynamic environment. We propose Dynamic P2P Source Routing (DPSR), a new routing protocol for MANETs that exploits the synergy between p2p and MANETs for increased scalability. By integrating Dynamic Source Routing (DSR) and a proximity-aware structured p2p overlay routing protocol, DPSR limits the number of the source routes that each node has to discover and rediscover to $O(logN)$, while retaining all the attributes of DSR for dealing with the specifics of ad hoc networks. This is in contrast to the maximum of N source routes each node has to maintain in DSR. Thus DPSR has the 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 MANETs; it implements the same functionalities as CAN, Chord, Pastry, and Tapestry, which can be exposed to the applications built on top of it via a set of common p2p APIs.

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: (1) 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 (2) Both use 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.

The common characteristics shared by p2p overlays and MANETs also dictate that both networks are faced with the same fundamental challenge, that is, to provide connectivity in a decentralized, dynamic 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 and MANET 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 a recent advancement in p2p overlay networks, i.e., proximity-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. The obvious idea of overlaying a p2p network (protocol) on top of a multi-hop routing protocol can be inefficient, as it is difficult to exploit the interactions between the two protocols. Instead, our proposed new routing protocol for MANETs, Dynamic P2P Source Routing protocol (DPSR), seamlessly integrates functions performed by p2p overlay routing protocols operating in a logical namespace and by MANET routing protocols operating in a physical namespace. Specifically, DPSR integrates Dynamic Source Routing (DSR) [8] and Pastry [16], a proximity-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 $O(logN)$, 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 maximum of N 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, Pastry. In the following, we give a brief overview of DSR and Pastry.

2.1 DSR

DSR [8] 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*.

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 packet which is flooded across the network in a controlled manner. In addition to the address of the initiator of the request and the target of the request, each ROUTE REQUEST packet contains a route record, which records the sequence of hops taken by the ROUTE REQUEST packet as it propagates through the network. ROUTE REQUEST packets use sequence numbers to prevent duplication. The request is answered by a ROUTE REPLY packet from the destination node. To reduce the cost of route discovery, each node maintains a cache of source routes that have been learned or overheard, which it uses aggressively to limit the propagation range of ROUTE REQUESTS.

When a route is in use, 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 via the link-level acknowledgment frame defined by IEEE 802.11, or by a passive acknowledgment, i.e., after a host sends a packet to the next hop, it overhears whether the next hop forwards the packet further along the path. If the route breaks due to a link failure, the detecting host sends a ROUTE ERROR packet to the source which upon receiving it, removes all routes in the host's cache that use the hop in error.

Optimizations suggested for DSR for reducing the overhead of route discovery include: (1) overheard and forwarded routing information are cached to reduce the frequency of route discovery invocations; (2) cached routes are used to generate replies to ROUTE REQUESTS to limit the propagation of ROUTE REQUESTS; and (3) ROUTE REPLY storms caused by nodes replying from their caches are prevented by delaying each reply by a period proportional to the number of hops to the destination. This also increases the probability that the source receives the shortest route first.

Optimizations suggested for DSR for improving the effectiveness of route maintenance include: (1) every node helps to maintain shorter routes by sending a gratuitous ROUTE REPLY if it knows of a shorter route to the destination than the one used in an overheard packet; (2) each node always attempts to salvage a data packet that has caused a ROUTE ERROR; (3) ROUTE ERROR packets received by a source node are piggybacked on its next route request to ensure increased spreading of information about stale routes; and (4) ROUTE ERROR packets that are forwarded or eavesdropped on are used to invalidate locally cached routes that contain the hop in error.

Comparison studies of DSR with other proposed routing protocols for MANETs [3, 6] have shown that DSR exhibits good performance at all mobility rates.

2.2 Pastry

Pastry [16] is one of several proximity-aware structured p2p routing protocols [4]. Although it is chosen for the design of DPSR in this paper, other structured p2p protocols such as CAN [15], Chord [17], and Tapestry [18] could potentially be used as well.

In a Pastry network, each 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.

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 $O(logN)$ 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 $(2^b \quad 1)$ entries each. Each of the $(2^b \quad 1)$ 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 $(2^b \quad 1)$ possible values other than the $(n + 1)$th digit in the present node's nodeId. In addition to a routing table, each node maintains a leaf set, consisting of $L/2$ nodes with numerically closest larger nodeIds, and $L/2$ 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 forwards a message to a node whose nodeId shares with the message key a prefix that is at least one digit (or b bits) longer than the 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.

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 ith row of the routing table from the ith node encountered along the route from A to Z. Finally, X announces its presence to the initial members of its leaf set and routing table, which in turn update their own leaf sets and routing tables.

3 Key Concepts

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. [1] Simulations performed in ad hoc network protocol studies such as [3, 6] have been limited to networks of up to 100 nodes and a small number of network connections (source-destination pairs). 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 medium access of wireless communication (e.g., for a 100 node network using DSR, the ratio of routing overhead to data packets for moderate to high mobility ranges from 2:1 to 10:1). Specifically, the size of the 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 that the memory required to store such routes is not a scalability concern. Rather, it is the overhead required to discover and rediscover these many routes that limits the scalability of DSR.

Destination	Source Route
$< nodeId_x >$	$< S_i....S_x >$

Table 1. A DPSR routing table or leaf set entry.

In contrast, in structured p2p overlay networks such as Pastry, each node maintains $O(\log N)$ routing state, independent of the number of different destinations that node has to send messages to. This suggests that a promising approach to improving the scalability of DSR is to limit the size of the routing state each node has to maintain by leveraging efficient structured p2p routing protocols. In the rest of the paper, we propose DPSR as one way of integrating DSR and Pastry.

4 DPSR Design

Like DSR, DPSR is proposed as a network layer protocol. Message destinations and nodes are addressed using IP addresses. DPSR adds a level of indirection to multi-hop routing in MANETs by assigning nodeIds from a circular name space to nodes in the MANET. A prefix-based routing scheme similar to Pastry is then employed to route data packets in the name space. Prefix-based routing has been shown to provide low delay stretch and other useful proximity properties as demonstrated by Pastry [4, 5].

4.1 Basic Design

NodeId assignment DPSR assigns unique nodeIds to nodes in a MANET as is done in Pastry. NodeIds are generated as the secure hashing (SHA-1) of the IP addresses of the hosts. Since the number of nodes in MANETs is small, i.e., in the order of hundreds, this ensures that with very high probability the nodeIds are unique.

Node state 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 leaf set and routing table entry. Since there is no underlying routing infrastructure in MANETs, each entry in a DPSR leaf set or a routing table stores the route to reach the designated nodeId, as shown in Table 1. As in Pastry, each routing table entry for a given node is chosen such that it is physically closer to that node than other choices for that routing table entry. This is achieved via a similar node joining process as in Pastry.

Routing Routing in the basic DPSR design is the same as in Pastry: a message key is first generated by hashing the message's destination IP address, and the message is routed using Pastry's prefix-based routing procedure. In DPSR, since both message keys and nodeIds are hashed from IP addresses, an exact match between a message

[1] We note that position-based routing protocols which rely on global position systems can be more scalable than topology-based protocols such as DSR.

key and the destination node's nodeId is expected. In other words, a message will be delivered to the destination node whose nodeId matches the message key, if that destination node is reachable via the wireless links. The only difference between DPSR and Pastry routing is that each hop in the DPSR network is a multi-hop source route, whereas each hop in the Pastry network is a multi-hop Internet route.

Node join The DPSR node joining process is similar to that of Pastry. The only difference is in constructing the contents of the routing table and leaf set entries: each entry in a DPSR routing table or a leaf set stores the source route to a DPSR node, while an entry in Pastry simply stores the IP address of a Pastry node. In both cases, network proximity is taken into consideration when choosing the best node for each routing table entry.

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 route has either crashed, or has moved out of the range of its adjacent nodes along the route. In either case, a route discovery for that node is invoked on-demand. If the route discovery still does not find a new route to the unreachable node, that node, if present in the leaf set, is replaced in a similar way as in Pastry.

4.2 Optimizations

The basic design of DPSR inherits all of the optimizations on route discovery and maintenance used by the DSR protocol (see Section 2.1). A number of additional optimizations are unique to the DPSR routing structures and operations.

Use of indirectly received source routes There are three ways a DPSR node can discover source routes: (i) via explicit route discovery; (ii) via overhearing routes in messages sent between neighboring nodes; (iii) via forwarded source routes. In the basic operations of DPSR, a node always chooses the shortest route explicitly discovered for each entry. As an optimization, for every route indirectly received, i.e., via (ii) and (iii), a node checks whether the route is a better candidate than the current corresponding entry in the leaf set and the routing table. If so, the new route replaces the old entry. This optimization thus constantly discovers fresh and low proximity routes for the leaf set and routing table entries.

Routing table and leaf set as route caches In addition to the "prefix-based view" of the routing table, or the "neighbor-node view" of the leaf set, the two routing structures can be viewed as two caches of source routes, similar to the route cache in DSR. This allows the use of *implicit source routes* to destinations, as in the DSR protocol. An implicit source route is a source route embedded in a normal source route. For example, an explicit source route $A \rightarrow B \rightarrow C \rightarrow D$ contains two implicit routes, $A \rightarrow B$ and $A \rightarrow B \rightarrow C$. The implicit source routes can be exploited to optimize the DPSR routing procedure. To send a data packet, it first searches all implicit source routes in the routing table and the leaf set for an exact match between the message key and the destination nodeIds. If this initial search returns a source route, DPSR uses it directly. Otherwise, the original DPSR lookup algorithm, same as Pastry's, is executed to return the source route to the next DPSR hop. In addition, these implicit routes can be used to populate newly created leaf set and routing table entries, for example, when a new node joins.

3-D routing table and 2-D leaf set To further extend the above idea of using leaf sets and routing tables as route caches, leaf sets can be extended to 2-D and routing tables extended to 3-D, i.e., each entry in a leaf set and a routing table contains a vector of N_r routes, where N_r is a configuration parameter. For each explicit and implicit route in each directly or indirectly received route, a node checks whether there is an exact match between the route's destination nodeId and some entry in the leaf set. If so, the route is inserted into the leaf set entry. In addition, the route is also inserted into the unique entry in the routing table, i.e., the one that matches the longest prefix with the nodeId of the route's destination. Obviously, the optimal choice of the number of backup routes N_r depends on the tradeoff between the availability of routes and their freshness. To maintain the freshness of the cached routes, an approximate FIFO replacement policy similar to that used in the DSR cache is used.

The above 3-D routing table and 2-D leaf set have two benefits: (i) They effectively increase the sizes of "route caches" by a factor of N_r, increasing the probability of finding an implicit route in routing data packets. (ii) They potentially reduce the need for route maintenance. If the shortest route, based on the hop count, in a leaf set or routing table entry is broken, instead of performing a route discovery to the destination node, the node can switch to use the shortest route among the backups in the same entry.

4.3 Discussions

4.3.1 Design Alternatives to the Pastry operations

A unique characteristic of MANETs, shared medium access, suggests several design alternatives to the original Pastry protocol operations. In a shared medium, packet delivery can be unreliable due to collisions in transmission. On the other hand, overhearing of packets transmitted by neighboring nodes can be used for routing state maintenance.

The Pastry joining algorithm requires the transmission of many critical messages each of which when lost, would cause restarting the entire joining process. The algorithm assumes a low probability of packet loss and a low cost of message transmission in the network. Both of these assumptions do not hold for wireless ad hoc networks. Additionally, the join algorithm (directly inherited from Pastry) in it's final stage requires the joining node to discover routes to all members of its leaf set and routing table, each of which may require a flooding, for a total of $L + O(logN)$ floodings. This suggests a potentially more efficient joining process in which the joining node simply floods the entire network once, and a selected subset of nodes, e.g., the potential candidates for the leaf set and the routing table entries, send replies back to the flooding node.

The Pastry routing table maintenance algorithm is designed to preserve the locality of routing table entries in the presence of network dynamics. The algorithm involves periodic communication with nodes in a subset of routing table entries and a subsequent comparison of the proximity of the exchanged routing table entries with the node's own. However, in MANETs, such periodic communication violates the on-demand nature of DSR and thus may incur high overhead. Proximity probing is a very high overhead exercise in MANETs if the route to the probed node needs to be discovered. The nature of the shared medium access of MANETs provides an efficient alternative. A node can use overhearing of routes to maintain locality of its routing table entries. In fact, the nature of the overhearing process guarantees that the routes overheard are from the physically nearby nodes. Continuous updates to routing table entries using overhearing can make DPSR resilient to degradation of routing table quality due to mobility. The cost of this operation is just the power consumed to operate the network access device in promiscuous mode.

4.3.2 Scalability

In MANETs, two notions of scalability are of interest: (1) up till what network sizes a reasonable data packet delivery ratio can be maintained, and (2) for a fixed-size network, how large the routing overhead is for a fixed packet delivery ratio. The lower the routing overhead, the more network bandwidth is available for sending data packets. The two notions, however, are inter-related. If a network is not as congested in delivering a fixed volume of data packets, it can be scaled up further.

We qualitatively compare the routing overhead of DPSR with DSR. As described in Section 2.1, using DSR, depending on the number of distinct destinations a node sends messages to, each node needs to maintain up to N routes in a MANET of N nodes. In contrast, using DPSR, the number of routes each node needs to maintain is limited to $O(logN)$, independent of the number of different destinations that node has to send messages to. The exact tradeoff between DPSR and DSR is more complicated since both discover and rediscover routes on-demand. But to the first order of approximation, DPSR is expected to incur less overhead than DSR when each node communicates with on average over $O(logN)$ other nodes (one-to-many).[2] The reduced routing overhead allows more data packets which compete for accessing the shared medium to be delivered successfully.

Since DPSR routes packets through several overlay hops, whereas DSR takes the direct path, if queuing delay is discounted, the routing delay using DPSR is expected to be longer than using DSR. However, the delay stretch, defined as the delay going through the overlay hops divided by the delay via a single DSR source route, is expected to be within a factor of two. With a random uniform distribution of nodeIds and node locations in the 2-D proximity space, the lengths of consecutive overlay hops in routing a message in DPSR increase exponentially by a factor of $2^{b/2}$, since the number of nodes matching each additional digit decreases by a factor of 2^b. Since the last hop dominates, and the earlier hops are directionless, i.e., they are equally likely to move towards or away from the destination node, the expected delay stretch is bounded by $\frac{1}{1 \quad 1/2^{b/2}}$, and thus is small than 2 for $b > 1$. Furthermore, this delay can be reduced whenever implicit routes are found and used to deliver data packets directly to their destination nodes.

In summary, under the first notion of scalability, if there are many nodes that communicate with multiple other nodes, we expect DPSR to scale up to a larger network size than DSR from lower routing overhead. As the network size increases, however, the scalability of both DSR and DPSR will be limited by the lengths of the source routes, because the longer the source routes, the more likely they will break. Under the second notion of scalability, DPSR is expected to deliver more packets than DSR for one-to-many communication patterns and perform comparably when a node communicates on average with a few other nodes.

[2] For example, many p2p applications exhibit such traffic patterns.

5 Other Related Work

PeerNet [7] 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 [14], DSDV [13], and TORA [12] 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 requires 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 [10] and DREAM [1], "graph-based" such as RGD [2], and "geographic-based" such as GPSR [9]. Among these, geographic forwarding approaches route packets based on only local decisions, and thus have less overhead and are more scalable. GLS [11] is a scalable distributed location service that can be combined with geographic forwarding to construct large ad hoc networks.

Acknowledgment

We thank Peter Druschel, Dave Johnson, and the anonymous reviewers for their helpful comments. This work was supported in part by an NSF CAREER award (ACI-0238379) and a Honda Initiation Grant 2002.

References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proc. ACM MOBICOM'98*.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proc. DialM'99*.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. ACM MOBICOM'98*.

[4] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Exploiting network proximity in distributed hash tables. In *Proc. FuDiCo'02*.

[5] 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.

[6] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. IEEE INFOCOM'00*.

[7] J. Eriksson, M. Faloutsos, and S. Krishnamurthy. Peernet: Pushing peer-to-peer down the stack. In *IPTPS'03*.

[8] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic, 1996.

[9] B. Karp and H. Kung. GPSR: Greedy perimeter stateless terminode routing. In *Proc. ACM MOBICOM'00*.

[10] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proc. ACM MOBICOM'98*.

[11] 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 *Proc. ACM MOBICOM'00*.

[12] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. IEEE INFOCOM'97*.

[13] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. ACM SIGCOMM'94*.

[14] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proc. WMCSA'99*.

[15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proc. ACM SIGCOMM'01*.

[16] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. MIDDLEWARE'01*.

[17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. ACM SIGCOMM'01*.

[18] 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.