Random Landmarking in Mobile, Topology-Aware Peer-to-Peer Networks

Rolf Winter, Thomas Zahn, Jochen Schiller
Institute of Computer Science
Freie Universität Berlin, Germany
{winter, zahn, schiller}@inf.fu-berlin.de

Abstract

DHTs can locate objects in a peer-to-peer network within an efficient amount of overlay hops. Since an overlay hop is likely to consist of multiple physical hops, the ratio between the number of physical hops induced by the overlay routing process and the number of physical hops on a direct physical path is often significantly lopsided. Recently, some approaches have been suggested to optimize that ratio by building topology-aware peer-to-peer overlays. However, none of them were explicitly designed to handle node mobility.

We present an approach that optimizes the overlay vs. direct physical path ratio and maintains it even in the presence of node mobility. Thus, it is well suited for highly dynamic networks, such as ad-hoc networks.

1. Introduction

Much of the research effort on peer-to-peer (P2P) computing has recently been devoted to distributed hash tables (DHT) and related issues to overcome the scalability problems of first-generation P2P systems [1]. The main advantage of DHTs is that they provide a guaranteed bound on the number of overlay routing hops that have to be taken to locate an object (i.e. a given key) on the overlay network. For [3, 4, 5] this bound is O(log N), where N is the number of nodes participating in the overlay network. However, they do *not* offer a guarantee on the number of physical hops taken during a lookup process, as a single overlay hop is likely to involve multiple physical routing hops. Consider the overlay routing example given in figure 1. Overlay node S initiates a lookup that will eventually be routed to overlay node T. In this example, three intermediate overlay routing steps are involved until the request reaches its final destination, clearly traveling a highly suboptimal physical route when compared to the direct physical path. Due to this discrepancy, the ratio between the number of physical hops induced by the overlay routing process and the number of physical hops on a direct physical path is often markedly lopsided, which can reduce the practical value of the O(log N) overlay bound.

The main contribution of this paper is the design and analysis of a new approach, *Random Landmarking*, that optimizes the overlay vs. direct physical path ratio and maintains it even in the presence of node mobility. Thus, it is well suited for highly dynamic networks, such as ad-hoc networks.

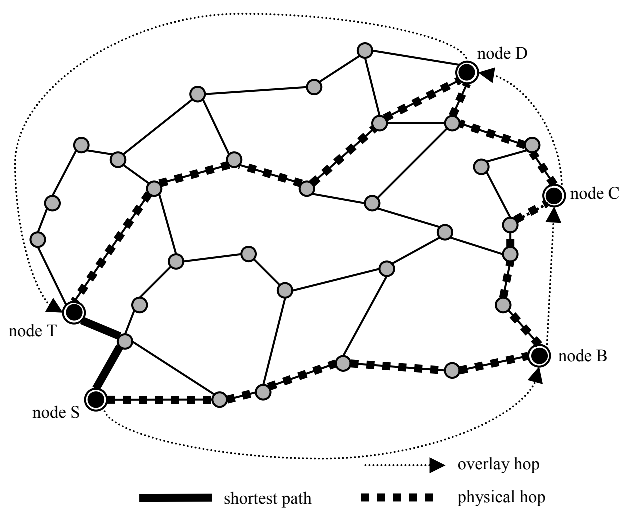

Figure 1: Overlay routing.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Section 3, we present Random Landmarking in detail. Section 4 analyzes and evaluates experimental results. Section 5 concludes this paper and gives a brief outlook on our future work.

2. Related Work

A significant amount of work has been dedicated to the development of P2P overlay networks, but so far only few approaches explicitly focus on making overlay networks reflect the locality properties of the underlying physical networks. Most importantly, none of the related approaches consider node mobility.

One of the general concepts used to close the gap between physical and overlay node proximity is landmark clustering. Ratnasamy et al. [6] use landmark clustering in an approach to build a topology-aware CAN [2] overlay network. Prior to joining the overlay network, a joining node has to measure its distance (e.g., RTT, hop count, or any other appropriate metric) to a fixed set of landmark nodes and assigns itself a point in CAN's virtual coordinate space according to its landmark distances. The intuition behind this idea is that nodes that have similar distances to all landmark nodes, are also quite likely to be close to each other topologically. However, a fixed set of landmarks renders this approach unsuitable for mobile networks. The most significant downside of this approach, however, is that it can lead to an extremely uneven overlay ID distribution. This means that a small set of nodes could be responsible for a very large part of the ID space, essentially turning them into hot spots. Xu et al. [7] have verified this in their study presenting a fine-tuned approach.

Waldvogel and Rinaldi [8] propose an overlay network (Mithos) that focuses on reducing routing table sizes. Mithos also tries to establish overlay locality from physical network proximity. A new node is assigned an overlay ID based on the IDs of its (physical) neighbors. They employ virtual springs to make the ID fit into the neighborhood range. In order to avoid local minima, substantial probing has to be undertaken. Unfortunately, only very small overlay networks (200 – 1000 nodes) are used for simulations and the impact of network degression is not considered.

Pastry [4] uses certain heuristics to exploit physical network proximity in its overlay routing tables. In a thorough analysis [9], Castro et al. examine the impact of various network parameters and node degression on Pastry's locality properties. Unlike the other approaches presented, Pastry does not construct its overlay structure from the underlying physical network topology. Instead, Pastry distributes its nodes evenly in the overlay ID space regardless of the actual physical topology. One way in which Pastry tries to exploit physical proximity is that a new node should bootstrap itself using a node close-by. During the join process, it then tries to choose among the candidate nodes for a particular routing table entry a node that is "close" to itself. During its lifetime, a node periodically performs routing table maintenance and improvement by asking other nodes for "better" routing table entries. Obviously, those are mere heuristics and, therefore, Pastry does not guarantee optimal routing table states.

3. Random Landmarking

Random Landmarking (RLM) tries to actively exploit physical proximity in the creation of overlay networks. Its main focus is on achieving good locality properties in the overlay network by inducing as little construction and maintenance overhead as possible while still maintaining an even overlay ID distribution. This will translate into an optimized overlay vs. physical routing distance ratio. Maintaining an even overlay ID distribution is especially important in ad-hoc networks with extremely heterogeneous devices where devices with scarce resources should not become hotspots.

Our implementation of RLM is based on a Pastry [4] overlay network. Pastry is a very well-known DHT that provides built-in locality heuristics. We chose Pastry because these heuristics – as mentioned above – have been thoroughly analyzed [9]. This analysis makes a good background against which to compare our experimental results. However, we believe that RLM's mechanisms are DHT-independent and could, thus, be ported to other DHTs.

RLM differs primarily from Pastry's approach by the way in which overlay IDs are assigned. Pastry's overlay construction basically works in a top-down fashion, i.e. Pastry randomly assigns overlay IDs regardless of the underlying topology. It, then, tries to make the physical proximity fit into the overlay routing state through the join process and table maintenance. In contrast, with RLM the overlay network is constructed in a bottom-up fashion, i.e. the overlay is built considering locality information from the underlying network. Before a node joins the overlay, it gathers information concerning its physical neighborhood and uses it to assign itself an appropriate overlay ID.

Conventional landmarking, as introduced in [6, 7], suffers from the limitation that it assumes a set of fixed, stationary landmark nodes. All overlay nodes are expected to know the landmark nodes and to measure their respective distances to those landmarks. This, obviously, reintroduces the client-server concept into the bootstrap process. Especially in mobile networks there are usually no sets of fixed nodes available, which renders this approach infeasible. Therefore, we introduce Random Landmarking (RLM) into the overlay construction process.

RLM utilizes the overlay lookup capabilities to locate overlay nodes responsible for a fixed set of landmark keys (overlay IDs). These nodes serve as temporary landmarks for a joining node. Landmark keys are chosen in a way that they divide the overlay ID space into equal portions. For example, in a network with an ID base of 16, an appropriate set of landmark keys would be: 000..00, 100..00, 200..00, …, F00..00. The joining node then measures the distances to those temporary landmarks and assigns itself an ID based on its landmark ordering. The advantage of this approach is that "landmark nodes" can fail and others will simply step in as overlay routing mechanisms of the underlying DHT will automatically redirect future key lookups to those nodes now responsible for the landmark keys. After having measured its landmark distances, the joining node adopts an ID prefix of a certain length from the landmark node closest[1] to itself. The ID remainder can be assigned randomly or can be based on an algorithm that further takes into account the physical neighborhood.

RLM has the following effects. First of all, it leads to physically close nodes forming regions with common ID prefixes, which means these nodes are also likely to be numerically close to each other in the overlay ID space, as can be seen in figure 2. Since the last overlay routing step is the numerically closest, another effect of RLM is that the last overlay routing step also tends to be the physically closest, whereas with Pastry the opposite is the case [4, 9].

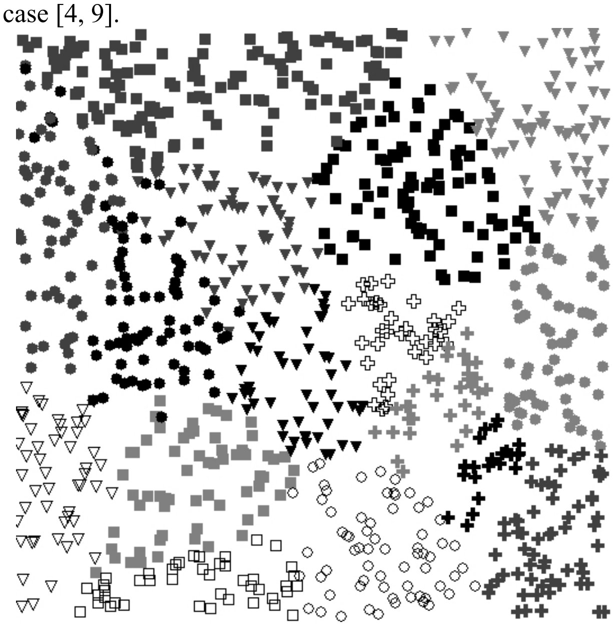

Figure 2: Spatial prefix distribution as generated by RLM. Equal symbols and shades represent equal prefixes.

4. Experimental Results

In this chapter, we present RLM's performance results that we obtained using the discrete event simulator Omnet++ [10]. In order to put RLM's results into perspective, we compared them to a Pastry reference implementation that is in strict conformance with the Pastry papers [4, 9]. We chose to run both our Pastry and RLM overlays in ad-hoc scenarios employing an AODV [11] physical routing layer.

4.1. Static Networks

The first set of simulations were run in order to verify the correctness of our Pastry reference implementation and to, thereby, create a background against which to compare RLM's results. For our simulations, we considered static networks with randomly distributed plane topologies of 1,000, 2,000, 5,000, and 10,000 participating nodes. All participating nodes form an underlying ad-hoc network. The average node connectivity is about 14, i.e. on average each node is within the transmission range of 14 other nodes. Furthermore, each physical node also participates in the overlay network. During a simulation run, 20,000 random key lookups are initiated by randomly picked overlay nodes.

We examined various Pastry bootstrap mechanisms. As a lower bound, we implemented an artificial bootstrap procedure where we used global knowledge to fill all overlay routing tables. This means that for each routing table entry the physically closest candidate is *always* known and chosen. However, global knowledge is an absolutely unrealistic assumption and, thus, this was only used as Pastry's theoretical best state in our scenarios.

We also examined a bootstrap mechanism that uses Pastry's standard join procedure. According to [9], after a new node has bootstrapped itself, it sends the n^{th} row of its routing table to each entry in that row. These entries, then, update their own routing tables. This optimization serves both to propagate information about newly joined nodes and to avoid cascading routing table inefficiencies. Obviously, it also induces a hefty network overhead.

To study what a Pastry network without any such overhead performs like, we also implemented a bootstrap mechanism that does not try to optimize the routing tables after node arrivals. With this mechanism, Pastry's locality properties have to rely on the mere heuristic embedded in its join process. This approach can be viewed as the upper bound on Pastry's performance in our scenarios.

In a next step, we considered two different approaches: RLM without bootstrap optimization and RLM with bootstrap optimization. Furthermore, we used 16 landmark keys.

Figure 3 shows the average ratio between the number of physical hops induced by an overlay lookup and the direct physical routing path between the source node and the target node for the bootstrap mechanisms mentioned above. As expected, Pastry's results correlate directly with the original results in [4, 9]. When global knowledge is applied during the bootstrap process, Pastry can achieve a ratio of around 1.33 on average. In the more practical case of Pastry's original bootstrap strategy, an average ratio of 1.45 is achieved. With no optimization, the ratio rises to around 2.47 as the number of participating nodes increases.

As can be observed, RLM achieves better or equal ratios in all tested networks *without* any optimization than Pastry does *with* its optimization. If RLM also utilizes the same bootstrap optimization as Pastry, it gains a ratio of 1.19, which is significantly lower than the best possible

[1] Conceivable metrics include hop count, RTT etc.

ratio that Pastry can only score when artificially bootstrapped.

Figure 4 depicts the total number of messages that have to be exchanged among all nodes during the bootstrap process in order to build up and optimize the overlay routing tables. Obviously, these different message efforts cause the varying ratios between bootstrapping with and without optimization as displayed in figure 3.

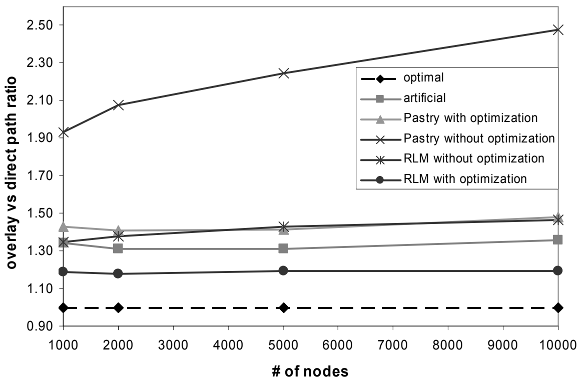

Figure 3: Overlay vs. direct path ratio of the various bootstrap mechanisms.

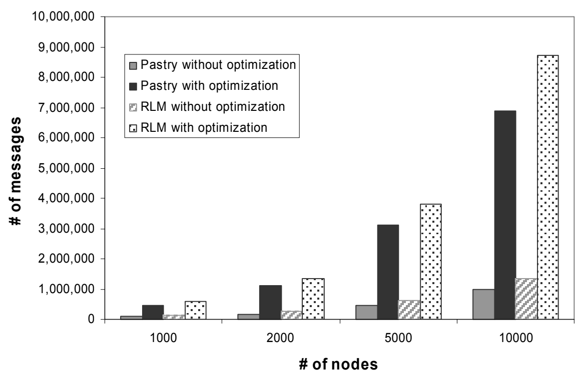

Figure 4: Total number of messages exchanged during bootstrap.

As can be seen, Pastry's bootstrap optimization introduces a significant overhead. With optimization, 6 to 7 times more messages have to be exchanged compared to Pastry's bootstrap without optimization. These messages include join requests and forwards, distance measurements, and messages containing routing table state information. Figure 4 also indicates that RLM without optimization can achieve the same ratio as Pastry does with optimization while inducing *significantly* less message overhead. Additionally, the simulation also showed that RLM's overlay ID distribution remains nearly optimal.

4.2. Mobility

Mobile networks represent the biggest challenge when building topology-aware overlay networks because the underlying physical network changes constantly. In order to evaluate the performance of RLM in such networks, we conducted several simulations comparing it to Pastry. For RLM, we implemented an ID reassignment strategy to deal with mobility. Pastry has no explicit mechanisms to deal with rapid topological changes in its underlying physical network. The only way to adapt its routing tables to reflect a modified physical underlay is to periodically run routing table maintenance tasks. These tasks are not run explicitly to detect mobility-induced changes, but instead are performed to compensate for any effects causing routing table deterioration.

In certain intervals, each Pastry node randomly selects other nodes from its routing table and probes them for better entries, as described in [9]. RLM uses a different strategy that deals explicitly with mobility-induced topology changes. Every node periodically re-measures its distance to the current landmark nodes. If its ID prefix is still congruent with the prefix of its closest landmark node, it will increase its re-measure interval by some factor. Otherwise, it will re-assign itself a new overlay ID based on the same strategy as used during its bootstrap and will rejoin the network with its new ID. Due to the extremely dynamic nature of mobile networks, a node uses the standard Pastry bootstrap optimization as explained in Section 3 after it has rejoined the network under its new ID. This serves to propagate its new ID faster.

Both Pastry and RLM were evaluated with the following network settings. Due to the added simulation complexity, a 2,000 node physical network served as the underlay. Each test run lasted 24 simulated hours. During those 24h, the usual 20,000 random lookups were issued. The nodes in the network moved according to the random waypoint mobility model [12] with a speed of 0.6m/s and a pause time of 30s.

For our Pastry experiments, we evaluated two different routing table maintenance intervals, as well as Pastry networks with no maintenance at all. Before mobility set in, all Pastry networks were artificially bootstrapped to start out with a ratio of 1.31. As figure 5 shows, if no routing table maintenance is performed, Pastry's ratio quickly deteriorates to a level of 2.35 that it would also roughly achieve in static networks without any optimization. Therefore, we considered next a maintenance interval of 1 minute so that each node runs the routing table maintenance task as explained above every minute. As can be seen, Pastry is unable to maintain a stable factor over time. Its ratio deteriorates nearly linearly until it reaches 1.97 after 24 hours. Clearly, this ratio does not peak here but would further rise. For this reason we conducted a second set of experiments, lowering the interval to 30s thereby increasing the maintenance effort markedly. The results indicate that even with this increased effort Pastry still fails to reach a stable ratio level after 24 hours with a ratio of 1.94 that is only negligibly lower than before.

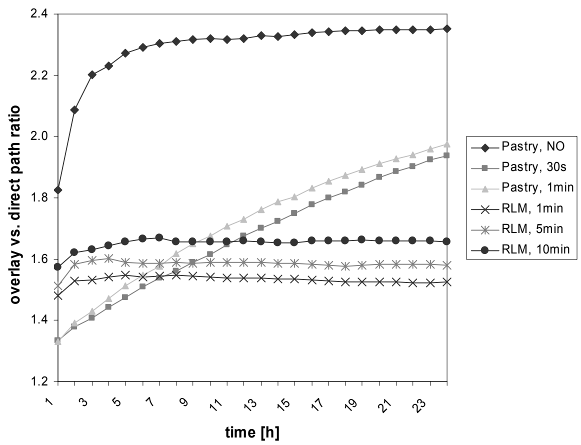

Figure 5: Overlay vs. direct path ratio change over time with Pastry and RLM.

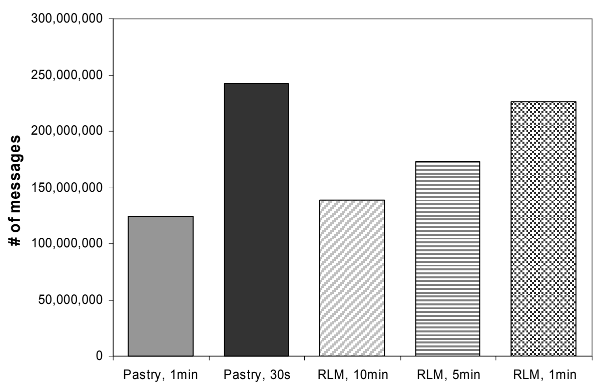

Figure 6: Number of overlay messages exchanged during an average 24h simulation.

With the same mobility parameters, we next evaluated RLM. Before mobility set in, all RLM networks were bootstrapped without optimization yielding an initial ratio of 1.37 slightly above Pastry's initial artificial ratio. Figure 5 shows that with a re-measure interval of 1 minute, RLM quickly reaches a stable ratio level of 1.54. If the re-measure interval is increased to 5 minutes, RLM achieves a stable ratio level of 1.58. Even if the nodes re-measure their landmark distances only every 10 minutes, RLM still maintains a stable ratio level of 1.65.

Figure 6 shows the number of overlay messages exchanged during an average 24h simulation run. These messages include rejoin requests and forwards (only with RLM), distance measurements, and messages containing routing table state information. In the case of RLM, those numbers also include on-demand routing table maintenance to take care of "false positives". A false positive in this case is a routing table entry that refers to a node by its old overlay ID although the node has rejoined the network under a new overlay ID in the meantime. In order to rid our routing tables of such false positives, we employ a form of on-demand maintenance. When a node receives a request, it checks whether it still has the overlay ID that the sending node thinks it does. If not, it will acknowledge that to the requesting node, so that the false positive gets removed from the sending node's routing table. With an increasing request rate, that extra effort becomes negligible as false positives are removed quickly due to the high request frequency. The higher the request rate, the lower the probability will be for a single request to encounter false positives as previous requests are likely to have already cleaned up the routing table. If one was not interested in other effects, such as routing table locality and clustering, and only cared about lowering the overall message overhead, RLM per se requires high request rates in order for its lower path ratio to outweigh its construction overhead. Therefore, we believe that the extra effort induced by false positives is negligible.

Figure 6 shows that Pastry exchanges large volumes of overlay messages during the simulated 24h with both maintenance intervals (124 million and 241 million, respectively). Despite that, Pastry is not able to reach a stable ratio level after 24h as figure 5 showed. RLM, on the other hand, achieves a stable ratio level (1.65) significantly better than Pastry's with a comparable amount of messages exchanged when the re-measure interval is 10 minutes. If one is willing to accept a message total above Pastry's 1-minute interval total but still well below Pastry's 30s interval total, RLM's ratio can be lowered even further (see figures 5, 6). It has to be mentioned that in object storage overlays the rejoin of a node induces some additional overhead. Before a node assigns itself a new ID, it would have to pass on the references to the objects it is currently responsible for to its left and right neighbor in the ID space as they now become responsible for them. After rejoining the network with its new ID, the node would have to acquire the references to the objects it has now become responsible for from its new left and right ID space neighbor. Obviously, this overhead depends largely on the number of objects being stored on the network and the number of participating nodes.

5. Conclusion & Future Work

In this paper we have analyzed an approach designed to construct topology-aware overlay networks especially suited for mobile networks. Our approach employs an adapted form of landmarking and ID reassignment to establish and maintain good locality properties. Random Landmarking achieves a comparable ratio between the number of physical hops induced by the overlay routing process and the number of physical hops on a direct physical path to Pastry with *significantly* less message overhead. On the other hand, with comparable message overhead, RLM achieves a ratio markedly better than even Pastry's artificial optimum.

While Pastry's built-in heuristics to deal with physical proximity work well in static networks, our experiments have shown that they are ineffective in mobile networks. We have shown that RLM outperforms Pastry already in static networks, with this effect becoming even more prominent in mobile networks. Again, RLM needs comparable network traffic to achieve its better ratios.

In the future, we plan to examine RLM's performance in a variety of different network topologies and parameters. Of special interest, here, is the impact of the node mobility rate (velocity). It would be interesting to see in which combinations of topology and mobility rates it is still useful to try to maintain locality properties and in which scenarios the frequency of mobility-induced topology changes renders the efforts prohibitively expensive.

References

[1] J. Ritter. Why Gnutella Can't Scale. No, Really. http://www.darkridge.com/~jpr5/doc/gnutella.html

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proc. of ACM SIGCOMM*, Aug. 2001.

[3] 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. of ACM SIGCOMM*, Aug. 2001.

[4] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *International Conference on Distributed Systems Platforms (Middleware 2001)*. Nov. 2001.

[5] 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.

[6] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-Aware Overlay Construction and Server Selection. In *IEEE Infocom '2002.*

[7] Z. Xu, C. Tang, and Z. Zhang. Building Topology-Aware Overlays using Global Soft-State. In *ICDSC'2003,* May 2003.

[8] M. Waldvogel and R. Rinaldi. Efficient Topology-Aware Overlay Network. *HotNets 2002, SIGCOMM/CCR 2003.*

[9] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Exploiting Network Proximity in Peer-to-Peer Overlay Networks. *International Workshop on Future Directions in Distributed Computing (FuDiCo)*, June 2002.

[10] Omnet++. Discrete Event Simulation System. http://www.omnetpp.org

[11] Ad hoc On-Demand Distance Vector Routing. http://www.ietf.org

[12] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing, Imielinski and Korth, Eds. Kluwer Academic Publishers,* 1996, vol. 353.