Rapid Mobility via Type Indirection

Ben Y. Zhao, Ling Huang, Anthony D. Joseph and John Kubiatowicz
Computer Science Division, University of California, Berkeley
{ravenben, hling, adj, kubitron}@cs.berkeley.edu

Abstract

Economies of scale and advancements in wide-area wireless networking are leading to the availability of more small, networked mobile devices, placing higher stress on existing mobility infrastructures. This problem is exacerbated by the formation of *mobile crowds* that generate storms of location update traffic as they cross boundaries between base stations. In this paper, we present a novel aggregation technique we call *type indirection* that allows mobile crowds to roam as single mobile entities. We discuss our design in the context of *Warp*, a mobility infrastructure based on a peer-to-peer overlay, and show that its performance approaches that of Mobile IP with optimizations while significantly reducing the effect of handoff storms.

1 Introduction

Economies of scale and advancements in wide-area wireless networking are leading to the widespread availability and use of millions of wirelessly-enabled mobile computers, Personal Digital Assistants (PDAs), and other portable devices. The same trends are also resulting in the large-scale deployment of publically acessible wireless access points in both fixed (*e.g.*, hotel, coffee shop, etc.) and mobile (*e.g.*, train, subway, etc.) environments [2].

We consider two rapid mobility scenarios. The first is rapid individual mobility across network cells (*e.g.*, a mobile user on an inter-city bus travelling on a highway with cell sizes of half a mile). This scenario requires fast handoff handling to maintain connectivity. A second, more problematic scenario is a bullet train with hundreds of mobile users. With cell sizes of half a mile, there are frequent, huge bursts of cell crossings that will overwhelm most mobility and application-level protocols.

The challenge is to provide fast handoff across frequent cell crossings for a large number of users, potentially traveling in clusters (*mobile crowds*). Handled naively, the delay in processing handoffs will be exacerbated by the large volume of users moving in unison, creating congestion and adding scheduling and processing delays and disrupting the timely delivery of packets to the mobile hosts.

A similar problem exists in cellular networks. As mobile crowds travel across the network, cells can "borrow" frequencies from neighbors, but base stations are often overloaded by control traffic and as a result, drop calls [6]. In certain cases, specialized "mobile trunk" base stations can be colocated with mobile crowds to aggregate control traffic. Ideally, each provider would place such a base station on each bus or train segment, but the individual component and maintenance costs are prohibitive.

Previous works propose to minimize handoff delay using incremental route reestablishment and hierarchical foreign agents or switches, by organizing the wireless infrastructure as a static hierarchy or collection of clusters [3, 12, 7]. A proposal also exists for Mobile IP to adopt a simplified version of hierarchical handoff management [8]. These approaches specify separate mechanisms to handle handoffs at different levels of the hierarchy. Also, since they statically define aggregation boundaries in the infrastructure, foreign agents or switches are prone to overloading by spikes in handoff traffic, such as those generated by the movement of large mobile crowds.

To address these issues, we introduce *Warp*, a mobility infrastructure leveraging flexible points of indirection in a peer-to-peer overlay. Warp uses a mobile node's unique name to choose the members of a virtual hierarchy of indirection nodes. These nodes act as hierarchical foreign agents to support fast handover operations. Warp also supports hierarchical types, where mobile crowds can redirect traffic through single indirection points and aggregate handoffs as a single entity. For example, an access point on the train can then perform handoffs as a single node while forwarding traffic to local mobile nodes. Although our techniques are applicable to most decentralized object location and routing (DOLR) networks [4], we discuss Warp in the context of the Tapestry peer-to-peer overlay.

We begin with a brief overview of the Tapestry [14] protocol. In Section 3, we discuss basic mobility support, followed by a discussion in Section 4 of rapid mobility and hierarchical type

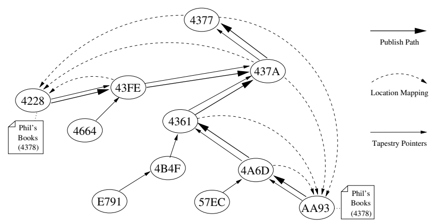

Figure 1: *Tapestry object publication.* Two copies of an object (`4378`) are published to its root node at `4377`.

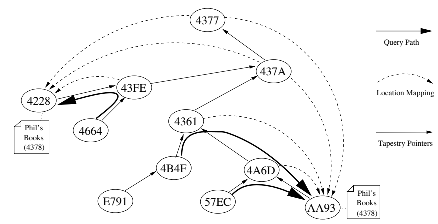

Figure 2: *Tapestry route to object.* Nodes send messages to object `4378`.

mobility. We present simulation results in Section 5, and finish with related work and our conclusions in Section 6.

2 Tapestry Overview

We provide a brief overview of Tapestry [14], a scalable structured peer-to-peer (P2P) infrastructure that routes messages to nodes and objects.

2.1 Routing Layer

Object and node IDs are chosen uniformly at random from the namespace of fixed-length bit sequences with a common base (e.g. Hex). Each node uses local routing tables to route messages incrementally to the destination ID digit by matching prefixes of increasing length (e.g., `4***` $\Longrightarrow$ `45**` $\Longrightarrow$ `459*` $\Longrightarrow$ `4598` where `*`'s represent wildcards). A node N has a routing table with multiple levels, where the n^{th} level stores nodes matching at least $n-1$ digits to N. The i^{th} entry in the j^{th} level is the location of the node *closest in network latency* that begins with $prefix_{j-1}(N)+i$.

To forward on a message from its n^{th} hop router, a node examines its $n+1^{th}$ level routing table and forwards the message to the link corresponding to the $n+1^{th}$ digit in the destination ID. This routing substrate provides efficient location-independent routing within a logarithmic number of hops and using compact routing tables.

2.2 Data Location

A server S makes a local object O available to others by routing a "publish" message to the object's "root node," the live node O's identifier maps to. At each hop along the path, a location mapping from O to S is stored. Figure 1 illustrates object publication, where two replicas of an object are published. A client routes queries toward the root node (see Figure 2), querying each hop on the way, and routing towards S when it finds the O to S location mapping. For nearby objects, queries quickly intersect the path taken by publish messages, resulting in low

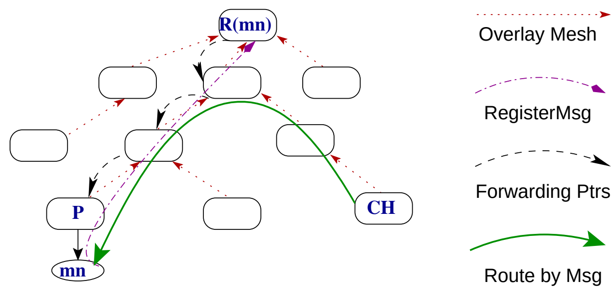

Figure 3: *Communicating with a mobile host.* Mobile node `mn` registers with proxy `P`, and correspondent host `CH` sends a message to `mn`.

latency routing to objects [14].

3 Mobility Support

We now discuss how mobility support can be layered on top of a structured peer-to-peer overlay. In this paper, we refer to *mobile nodes* (MN) that interact with *correspondent hosts* (CH).

3.1 Basic Mobility Support

A mobile node roaming outside of its home network connects to a local proxy node as its temporary care-of-addresses. Mobile nodes are client-only nodes that do not route or store data for the overlay. We assume that infrastructure nodes are nodes with relatively fixed positions, giving them the perspective of a relatively stable infrastructure. Nodes join and leave the infrastructure using Tapestry's dynamic membership algorithms [5].

Node Registration As with mobile IP, a mobile node `MN` registers itself with a nearby proxy node `P`[1]. When a proxy receives a registration from `MN`, it uses the DOLR interface [4] to publish `MN` as an endpoint. We call this reduction from a node to an object *type indirection.* At each node along the path from proxy to `MN`'s root node, a local pointer to the last node on

[1] Registrations are encrypted with a node's private key. Node IDs are hashes of public keys and verified by certificates issued by a central certificate authority

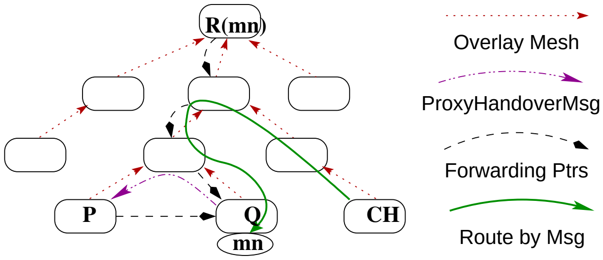

Figure 4: *Updating a location binding via ProxyHandoverMsg*. Correspondent host `CH` sends a message to mobile node `mn` after `mn` moves from proxy `P` to `Q`.

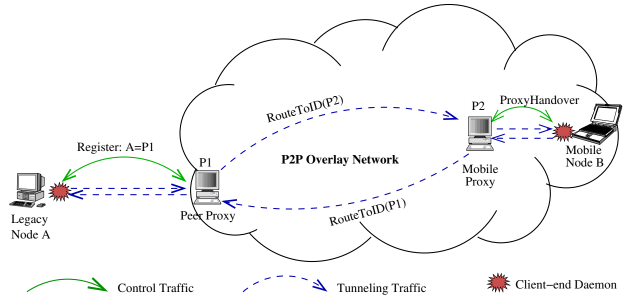

Figure 5: *Tunneling legacy application traffic through client-end daemons and overlay proxies.*. A legacy node `A` communicates with mobile node `B`.

the path is stored. The result is a multi-hop *forwarding path* from `MN`'s root to its proxy.

When a correspondent host `CH` sends a message to `MN`, Tapestry routes the message towards `MN`'s root. When the message intersects the forwarding path, it follows the path of pointers to the proxy and `MN`. Figure 3 shows a node `CH` routing a message to `MN`. Note that hops in structured overlays such as Tapestry generally increase in physical length (# of IP hops) closer to the destination. Messages avoid the longer hops to the root by intersecting the forwarding path. This is key to reducing routing stretch for communication with closeby `CH`'s.

Unlike other approaches to traffic redirection [11], Tapestry uses the overlay to transport both control and data traffic. By using points inside the network to redirect traffic, we eliminate the need to communicate with the endpoints when routes change. In the case of Warp, it means that as nodes move, proxy handover messages modify the forwarding path between proxies without incurring a roundtrip back to the home agent or correspondent host.

Mobile nodes listen for periodic broadcasts from nearby proxies for discovery, similar to techniques used by Mobile IP. Fast-moving nodes can proactively solicit proxy nodes via expanding ring search multicast to reduce discovery latency.

Proxy Handover Mobile node `MN` performs a proxy handover from `P` to `Q` by sending a ProxyHandoverMsg to `Q`, <`MN`, `P`, `Q`> signed with its secret key. `Q` sets up a forwarding route to `MN`, and requests that `P` sets up a forwarding pointer to `Q`. `Q` then routes the ProxyHandoverMsg towards `MN`'s root node, and builds a forwarding path to itself. The message is forwarded until it intersects `P`'s forwarding path. Note the path taken by the handover message is roughly proportional to the distance between `P` and `Q`. This is a key distinction from basic Mobile IP, and is analogous to a version of hierarchical handoff [3] with dynamically constructed, topologically-aware hierarchies.

When the message intersects a node `A` that is on the forwarding path to `MN`, it redirects the forwarding pointers to point to the new path. `A` then forwards the message downwards to `P`. Each node along the way schedules its forwarding pointer for deletion and forwards the message towards `P`[2]. When the message reaches `P`, `P` schedules the forwarding pointer to `Q` for deletion. Once all deletions are completed, handover is complete. The process is shown in Figure 4.

If the proxies do not overlap in coverage area, then `MN` will have a window of time after it leaves coverage of `P` and before it completes handover to `Q`. In this scenario, `P` performs a limited amount of buffering for `MN`, and then forwards the buffer to `Q` when a forwarding pointer is established [1].

Location Services for Mobile Objects We also support the routing of messages to objects residing on mobile nodes. An object named `O` residing on mobile node `MN` is published in the overlay with the location mapping from `O` to `MN`. A message for `O` routes towards `O`'s root until it finds the location mapping. Recognizing `MN`'s ID as a mobile address[3], the overlay routes the message for `O` as a normal message addressed to the mobile node `MN`. The message routes to `MN`'s proxy, `MN`, then `O`.

3.2 Supporting Legacy Applications

Warp supports communication between mobile nodes and legacy (non-overlay) nodes using a mechanism similar to that introduced by ROAM [16]. Mobile nodes are assigned unique DNS names with a specialized suffix, such as `.tap`. The mobile node

[2] A delay in deleting forwarding pointers is required to handle potential reorderings of messages between nodes by the underlying transport layer.

[3] All mobile node IDs share a specialized tag appended to their normal ID

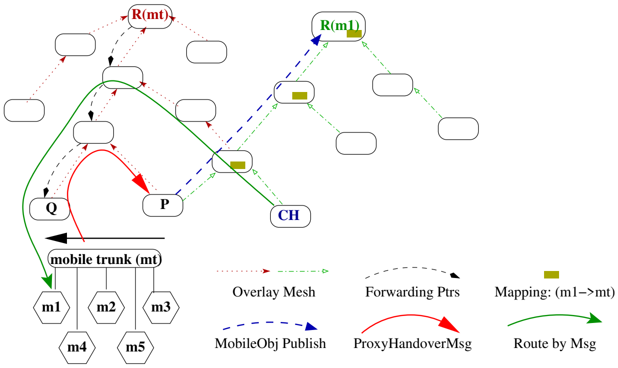

Figure 6: *Mobile crowds*. Five members (`m1..5`) of a crowd connected to a mobile trunk (`mt`). A message routes to `m1` as the crowd moves from proxy `P` to `Q`.

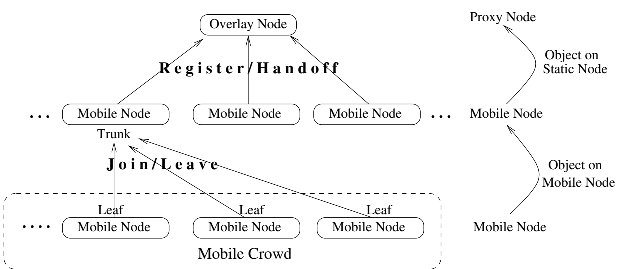

Figure 7: A figure summarizing levels of *type indirection*. The arrows on right illustrate relative relationships between types.

stores a mapping from a hash of its DNS name to its overlay ID into the overlay.

Figure 5 shows an example of the connection setup. Legacy node `A` wants to establish a connection to mobile node `B`. The local daemon redirects the DNS lookup request, retrieves the mobile node's stored ID using a hash of `B`, and forwards traffic through the overlay address to `B`'s overlay ID.

4 Supporting Rapid Mobility

Recall that in our approach, routing to mobile nodes uses indirection to translate a mobile ID into an overlay identifier. Routing to a mobile object goes through two levels of this *type indirection*, from object ID to mobile node ID to proxy ID. Here we discuss chaining multiple levels of type indirection to aggregate mobile crowds as single entities, reducing handoff message storms to single handoff messages.

4.1 Mobile Crowds

A *mobile crowd* forms where large groups of mobile users travel together. Examples include a large number of train passengers with wireless laptops and PDAs or tourists wirelessly accessing information on historic sites on a group tour. Such groups cause large bursts of handoff messages as they move in unison across cell boundries.

To minimize the resulting delay and congestion at nearby basestations, we choose a mobile node as the *mobile trunk*, and use it as a secondary proxy for others in the mobile crowd. The trunk advertises each member of the crowd (a *mobile leaf*), as a locally available object. Messages to a mobile leaf routes first to the trunk, then to the leaf. As the crowd moves across cell boundaries, only the trunk needs to update its location with a single handover.

Figure 6 shows an example. When a mobile node joins a mobile trunk in the crowd, the trunk publishes the <`m1`,`mt`> "location mapping." A message addressed to `m1` routes towards `m1`'s root. When it finds a location mapping, the message is redirected towards node `mt`. It encounters the mapping from `mt` to its proxy `Q`, routes to `Q`, `mt`, then `m1`.

4.2 Discussion

Type indirection reduces handoff messages from one message per node to one message per crowd. For more flexibility, a crowd can choose an unique crowd ID. Any mobile trunk would register with the proxy using the crowd ID instead of its own node ID. This allows multiple trunks to function simultaneously to guard against trunk failures or departures. Furthermore, since the trunk can suffer degraded performance, the responsibility can rotate across crowd members at periodic intervals to provide fairness.

We can further chain together type indirections for more interesting scenarios. For example, multiple bluetooth-enabled devices on a passenger may form a personal mobile crowd. These devices connect to a local mobile trunk, which joins a mobile trunk on the tour bus, which itself acts as a mobile node traveling through the network. Figure 7 shows different types of mobility, and how we leverage type indirection.

5 Measurements and Evaluation

In this section, we evaluate our infrastructure design via simulation. Our performance metric is *routing stretch*, the ratio of routing latency on an overlay to the routing latency of IP. We use the shortest path latency as the IP layer latency. Note that our results do not account for computational overhead at nodes. We believe that processing time will be dominated by network latencies. More comprehensive measurement results are available [15].

We use a packet-level simulator running on transit stub topologies [13] of 5,000 nodes. Each topology has 6 transit domains of 10 nodes each; each transit

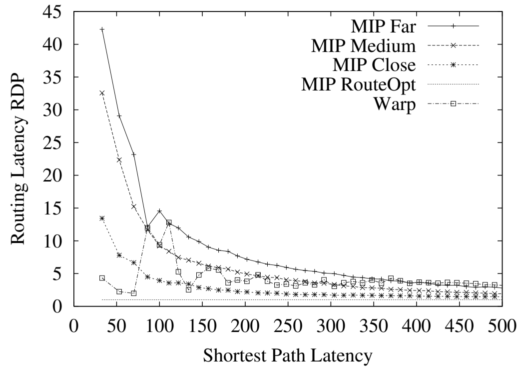

Figure 8: *Routing stretch*. Routing latency via Warp and Mobile IP measured as a ratio of shortest path IP latency. Mobile IP with pop-up mode achieves a stretch of 1.

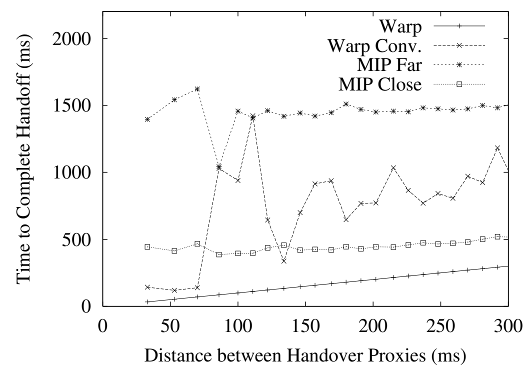

Figure 9: *Handoff latency* as a function of density of adjacent proxies or base stations. For Mobile IP, we measure both when the MN is close and far from home. Warp converge is the time to full routing state convergence.

node has 7 stub domains with an average of 12 nodes each. Our simulator measures network latency, but does not simulate network effects such as congestion, routing policies, or retransmission at lower layers. To reduce variance, we take measurements on 9 different 5,000 node transit stub topologies, each with 3 random overlay assignments.

5.1 Routing Efficiency

We studied the relative routing performance of our system and Mobile IP under different roaming scenarios. Mobile IP performance is a function of the distance from MN to NODECH, and from MN to its HA. Our system allows free roaming without a home network, and latency is dependent on the distance between CH and MN. We compare our system against three Mobile IP scenarios, where the distance between MN and its HA is (1) $< \frac{1}{3} \cdot D$ (near), (2) $> \frac{2}{3} \cdot D$ (far), and (3) $> \frac{1}{3} \cdot D$ and $< \frac{2}{3} \cdot D$ (mid), where D is network diameter.

Figure 8 shows that for correspondents close to the mobile node (*i.e.*, MN near CH), basic Mobile IP generally performs quite poorly under scenarios 1 and 3 due to triangle routing. In contrast, Mobile Tapestry's RDP shows some initial variability for short routing paths, but generally performs well with low stretch, outperforming Mobile IP under scenarios 1 and 3. Note that Mobile IP with route optimization [9] achieves a routing stretch of 1. Its use in conjunction with smooth handoffs [10] would reduce the round trip overhead to send location updates to CH at every handoff.

5.2 Rapid Mobility

We evaluate Warp's support for rapid mobility by comparing latency to handle cell handovers relative to Mobile IP. Time is measured from the initial request for location binding update to when all forwarding routes are updated and consistent. Figure 9 show that when the mobile node roams far from its home network, it can take between 1-2 seconds for basic Mobile IP to converge after a handoff request. Note that this result is independent of the rate of movement, and is only a function of distance from the home network. In contrast, handoff latency in Warp is linear to the movement rate. Note that the redirection of traffic via convergence points in Tapestry is similar in function to hierarchical foreign agents in Mobile IP [8].

Note that the "jitter" or delay in traffic seen by the application during handoff is not identical to handoff latency. It is the time elapsed before a valid forwarding path is constructed to the new proxy. Warp sets up an immediate forwarding path between the proxies to allow seamless traffic forwarding while updating the full forwarding path, similar to the Mobile IP smooth handoffs scheme [10]. In cellular networks, the jitter, or latency between adjacent proxies, is often less than 50ms and within the tolerable range of most streaming media applications.

Finally, we examine the load placed on network routers by mobile crowds. Specifically, we count the expected number of handoff messages required as mobile crowds cross boundaries between base stations. We consider several scenarios: 1) naive mobility support with no aggregation, 2) using aggregation while assuming uniform distribution of crowd sizes from 1 to 50, 3) using aggregation with exponential distribution of crowd sizes with parameter $p = 0.1$, 4) using aggregation with a binomial distribution of crowd sizes centered around 20 with

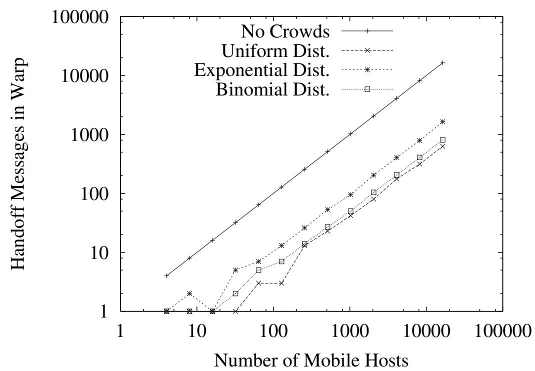

Figure 10: *Handoff load*. Reducing handoff messages of mobile crowds in Warp as a function of population size. Crowd sizes follow uniform, exponential, and binomial distributions.

parameter $p = 0.5$. Figure 10 shows the significant reduction in handoff messages. As the overall population increases, the net effect is a linear factor reduction in handoffs based on the mean crowd size. The result means that Warp can support larger and faster mobile crowds while using less bandwidth.

6 Related Work and Conclusion

The Internet Indirection Infrastructure project [11], supports a mobility framework (ROAM [16]) by storing generic triggers in the network infrastructure for traffic redirection. Each trigger maps a mobile node ID to its current IP address. A mobile node chooses an overlay node based on its mobile ID, and sends it trigger location updates to it while roaming.

I3 triggers can be used to simulate a variety of mobility mechanisms, including hierarchical mobility and aggregation among mobile crowds. Whereas Tapestry uses the structured routing mesh to form the hierarchies necessary for traffic redirection, ROAM nodes would require input from the mobile nodes to construct them in an ad-hoc fashion.

Compared to previous proposals for hierarchical management to localize handoff processing [3, 8], our hierarchy based on the Tapestry routing mesh is self-organizing and self-repairing. Optimizations similar to our proxy forwarding have been proposed for Mobile IP [10]. While the performance of Warp is similar to Mobile IP with these optimizations, its main contribution is the use of type indirection to aggregate mobile crowds into a single mobile entity.

In summary, Warp is a framework that treats mobile nodes as objects residing on proxies. We propose the use of type indirection to aggregate mobile crowds as single mobile entities to reduce handoff messages. While Warp and the DOLR interface can be deployed on any peer to peer protocol that supports the Key-Based Routing API [4], overlays that utilize proximity neighbor selection will produce better routing performance.

References

[1] BALAKRISHNAN, H., ET AL. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks 1*, 4 (Dec 1995).

[2] BREWIN, B. Transportation companies moving to offer Wi-Fi service. In *ComputerWorld*. Aug 2003.

[3] CACERES, R., AND PADMANABHAN, V. N. Fast and scalable handoffs for wireless internetworks. In *Proceedings of MobiCom* (November 1996), ACM.

[4] DABEK, F., ZHAO, B., DRUSCHEL, P., KUBIATOWICZ, J., AND STOICA, I. Towards a common API for structured P2P overlays. In *Proc. of IPTPS* (Berkeley, CA, Feb 2003).

[5] HILDRUM, K., KUBIATOWICZ, J. D., RAO, S., AND ZHAO, B. Y. Distributed object location in a dynamic network. In *Proc. of SPAA* (Winnipeg, Canada, Aug 2002), ACM.

[6] KATZELA, I., AND NAGHSHINEH, M. Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey. *IEEE Personal Communications Magazine 3*, 3 (June 1996).

[7] KEETON, K., MAH, B. A., SESHAN, S., KATZ, R. H., AND FERRARI, D. Providing connection-oriented network services to mobile hosts. In *Proceedings of MLIC* (August 1993), USENIX.

[8] PERKINS, C. E. Mobile-IP local registration with hierarchical foreign agents. IETF Draft, Feb 1996.

[9] PERKINS, C. E., AND JOHNSON, D. B. Route optimization in Mobile IP. IETF draft., Nov 1997.

[10] PERKINS, C. E., AND WANG, K. Optimized smooth handoffs in Mobile IP. In *Proceedings of ISCC* (July 1999), IEEE.

[11] STOICA, I., ADKINS, D., ZHUANG, S., SHENKER, S., AND SURANA, S. Internet indirection infrastructure. In *Proc. of SIGCOMM* (Aug 2002), ACM.

[12] TOH, C.-K. The design and implementation of a hybrid handover protocol for multi-media wireless LANs. In *Proceedings of MobiCom* (1995), ACM.

[13] ZEGURA, E. W., CALVERT, K., AND BHATTACHARJEE, S. How to model an internetwork. In *Proc. of INFOCOM* (1996), IEEE.

[14] ZHAO, B. Y., ET AL. Tapestry: A global-scale overlay for rapid service deployment. *IEEE JSAC 22*, 1 (January 2003), 41–53.

[15] ZHAO, B. Y., JOSEPH, A. D., AND KUBIATOWICZ, J. D. Supporting rapid mobility via locality in an overlay network. Tech. Rep. CSD-02-1216, U. C. Berkeley, Nov 2002.

[16] ZHUANG, S. Q., ET AL. Host mobility using an internet indirection infrastructure. In *Proceedings of MobiSys* (May 2003).