SUPPORTING GROUPWARE IN MOBILE NETWORKS

Nadav Lavi
Israel Cidon
Idit Keidar
Department of Electrical Engineering, Technion - Israel Institute of Technology

Abstract We present MaGMA (Mobility and Group Management Architecture), an architecture for groupware support in mobile networks. MaGMA's main objective is enabling mobile users to use real-time group applications over the IP infrastructure. Our solutions address group management as well as support for QoS and seamless handoff. We illustrate the advantages of MaGMA using mathematical analysis and simulations.

1. Introduction

The widespread availability of the Internet has enabled the use of many groupware and collaborative computing applications (e.g., chat, ICQ, NetMeeting, Exchange, Lotus Notes, Webex, desktop video conferencing, etc.). With the advance of wireless personal communication, such groupware applications are becoming popular in cellular and mobile networks [16]. For example, major cellular providers (Verizon, Nextel, Orange) offer, or plan to offer soon, group services such as push-to-talk (PTT) [7, 18]. The PTT cellular revenue, which was $84 million in 2003, is expected to reach $10.1 billion by 2008; and the 2.3 million PTT cellular subscribers community of 2003 is expected to grow to 340 million by 2008 [19]. While traditional PTT is limited to voice, the emerging convergence is expected to merge real time and non-real time aspects of group communication.

The converged Internet infrastructure is starting to provide the required support for real-time applications, such as voice over-IP (VoIP) and video-conference over-IP (VCoIP), which require quality of service (QoS) among stationary endpoints. This has led to the emergence of many QoS standards and technologies, e.g., Diffserv, RSVP, and MPLS, as well as real-time protocols such as RTP, H.323, MGCP, and SIP.

At the same time, wireless access to the global Internet is becoming widely supported, and WLAN access points are ubiquitously available. Given the trends predicted in wireless standard forums such as the Open Mobile Alliance

(OMA) [9] and 4G, it is expected that the next phase in the evolution of converged group services will be their integration with wireless mobile devices. These networks are most likely to adopt the TCP/IP architecture including its related convergence standards.

In this paper, we propose MaGMA an architecture for group management in mobile networks interconnected via the global Internet. MaGMA provides an initial comprehensive solution for the mobile world, addressing aspects such as scalable group management, mobility, handoff, and QoS provision. MaGMA's architecture consists of a collection of mobile group managers (MGMs), which manage group membership and also implement a multicast overlay for data delivery. Each mobile node (MN) interacts with an MGM proximate to it. MaGMA supports a subscription model in which nodes can request to be notified of other nodes' mobility.

We propose a number of group management protocols. We have implemented MaGMA in the ns2 network simulator [8]. We present simulation results and validate them through mathematical analysis.

This paper proceeds as follows: Section 2 gives background on current mobility solutions. Section 3 describes the network model and our proposed architecture. Section 4 presents solutions for mobile group management. Section 5 evaluates the proposed solutions through simulations and analysis. Section 6 addresses transport issues, and Section 7 concludes.

2. Related Work

We are not aware of any previous comprehensive solution for mobility support in groupware applications. We now overview leading mobility solutions for IP in general, and group communication systems in particular.

Mobility Solutions Mobile IP [10] is the current standard for seamless mobility in the IPv4 based Internet. Mobile IP uses a simple method of encapsulation and tunneling. Every MN is associated with a home domain, in which resides a proprietary server named *home agent*. While at its home domain, the MN receives packets as a regular stationary node. When the MN moves to a foreign domain and changes its IP address, it notifies its home agent of its new IP address. Thus, the home agent can forward to the MN packets destined to the MN's home IP address through a tunnel it creates to the new location. This forwarding scheme, called *triangle routing*, generally leads to routes that are longer than the direct path, and therefore suffers from poor performance. Moreover, Mobile IP applies only to unicast sessions between the MN and a corresponding node and does not include QoS support.

In [11], a route optimization to Mobile IP that avoids triangle routing is proposed. In this approach, the MN sends *binding* information to the corresponding node, thus enabling direct communication between the two. It is unclear whether Mobile IP with route-optimizations can support simultaneous movements of both endpoints of a communication. Moreover, this solution requires a modification of the host IP stack, and a home agent in each MN's home domain, and is therefore difficult to deploy. In addition, creating new connections to a MN must always involve its home domain, even if the MN is distant from it for an extensive period. In contrast, MaGMA provides a flexible architecture in which the communication infrastructure is deployed in the network, and communication with an MN is independent of its home domain.

Balakrishnan and Snoeren [14] propose a DNS-based solution to IP mobility. Similar to Mobile IP, every node has a home domain. When an MN moves and changes its IP address, it registers a secure DNS update at its home domain DNS server. In order to avoid the use of stale binding information, DNS caching is minimized (by setting TTL=0) and direct binding is used. The major drawbacks of this approach are that both endpoints cannot move simultaneously, that DNS standards do not support user self-configuration, and that operating systems and DNS servers often do not comply with DNS TTL directions. Finally, DNS caching elimination will overload the DNS system.

Mysore and Bhaghavan [6] propose to use the IP multicast infrastructure for mobility support. While this solution can potentially provide good performance, unfortunately, IP multicast is not widely deployed. Therefore, this scheme cannot provide seamless mobility in today's Internet.

Group Communication A closely-related group-management protocol is CONGRESS [1], which was designed for ATM environments. Like MaGMA, CONGRESS uses an overlay among servers. However, in contrast to MaGMA, the overlay is hierarchical and restricted to membership management, and does not support QoS multicast. Moreover, CONGRESS was not designed with mobility in mind and does not incorporate a handoff solution.

Prakash and Baldoni [12] propose protocols for group communication support in virtual cellular networks where base-stations can move, and for ad-hoc networks. In contrast, we consider a network with stationary base-stations. Bartoli [2] proposes a totally-ordered multicast protocol for a dynamic membership in wireless networks. In contrast to MaGMA, it assumes a failure-free environment and focuses on reliability and ordering rather than QoS support.

3. Model and Architecture

3.1 Network model

Similarly to mobile IP, we model the network as a collection of autonomous domains. Every MN has a unique ID (UID), which identifies the MN in all of its locations. Upon moving to a new domain, the MN obtains a new local IP address, e.g., using DHCP. We do not address intra-domain handoff, i.e., micro mobility. Rather, we assume that a micro-mobility mechanism is in place (e.g., [5, 13, 17]), and that an adequate routing protocol exists in each domain. We assume that hosts can crash, and that such crashes are detectable by other hosts.

3.2 Design goals

Our main goal is to provide support for managing and keeping a coherent up-to-date view of each group in a highly dynamic mobile environment. The solution we seek should address the following issues:

- Mapping group names to their current subscribers.
- Mobility support with seamless handoff.
- QoS support for real time applications.
- Transport efficiency, including the avoidance of triangle routing, and minimizing the number of duplicates of multicast messages sent.
- Low control overhead and a scalable control plane.
- Support for incremental deployment.

3.3 Architecture

Our architecture consists of a collection of MGMs positioned in different domains. For simplicity's sake, through most of this paper we assume that MGMs are static and well known. In Section 4.5, we discuss possible extensions of the basic architecture in which MGMs can be added on-the-fly. The role of MGMs is twofold: managing group membership and forwarding packets in order to facilitate QoS multicast. Our architecture calls for the use of multiple servers for the following reasons:

- to offer scalability in the number of groups and the number of group members;
- to efficiently support groups with geographically dispersed members, as well as localized ones;
- to facilitate QoS reservation among domains;
- to reduce traffic overhead; and
- to provide fault-tolerance in the presence of network partitions (where a node may not be able to communicate with a remote server) as well as server failures.

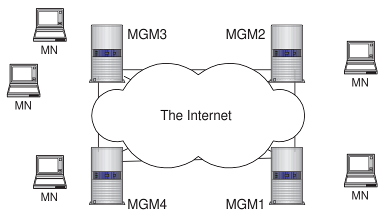

Figure 1. The MaGMA network architecture.

Our architecture is depicted in Figure 1. Ideally, one MGM is located in each domain, although this is not mandatory. Each MN is served by the closest MGM to its domain. The MGMs provide the following services to MNs:

- *Joining* or *leaving* a group.
- *Moving* to a new location - the moving node sends a *move* message to its new MGM.
- *Retrieving* the current membership view (list of current subscribers) of a given group and *multicasting* to a given group.

The MGMs form an overlay network among them. The overlay is used both for membership management and QoS multicast. The overlay construction can be employ known techniques for building efficient QoS-aware overlays, e.g., [15, 4], and its discussion is beyond the scope of this paper. We assume that MNs are likely to remain in the groups they join for extensive periods. Therefore, move messages dominate the control traffic.

4. MGM Protocols

We now present protocols for managing group membership of mobile users. The first two protocols are based on existing mobility solutions. We then propose two additional protocols, which handle mobility explicitly.

4.1 Exploiting Mobile IP

Since IP Mobility is the de facto standard for mobility on the Internet, we first consider a naïve solution based on this approach. One can delegate the responsibility for mobility management to Mobile IP, and have the MGMs only map group names to nodes' home addresses. This eliminates the need to handle *move* requests. The main drawback of this solution is the uncontrolled delay and QoS degradation resulting from Mobile IP's triangle routing.

4.2 DNS based solutions

As an alternative to Mobile IP, DNS-based mobility [14] can be used when DNS support is available and user mobility is limited. We now examine possibilities for extending this solution to support group management.

A simple approach, similar to the Mobile IP based solution suggested above, can delegate mobility handling to the nodes' home domain DNS servers, and have the MGMs map each group to its subscribed MNs in a domain name format. With this approach, a retrieve operation first gets from the MGM a list of MN names and then employs DNS queries to translate the MN names to actual IP addresses. This solution is simple and can be gradually implemented in today's Internet. However, the retrieve-translate procedure can take a substantial amount of time. Moreover, this solution suffers from the previously described problems associated with dynamic DNS resolution.

An alternative DNS-based solution replaces the MGM with a proprietary group DNS server used only for group translations. This DNS server maps group names directly to their subscribed MNs' IP addresses, thus reducing the translation delay. An additional drawback of this suggestion is that current standards do not support DNS server updates by hosts from foreign domains.

4.3 MGMFlood

We now turn to solutions in which mobility is handled by the MGMs and not delegated to other services. In our first such scheme, MGMFlood, each MGM forwards (floods) to all other MGMs all control messages (*join/leave/move*) received from MNs in its domain. When an MN crashes, its local MGM detects the crash and sends an appropriate *leave* message to all other MGMs.

Note that the protocol must guarantee view consistency in the presence of rapid mobility. When an MN frequently changes its location, it updates a different MGM each time it moves. Thus, different MGMs can receive the flooded *move* messages in different orders. In order to ensure consistency, each MN keeps an increasing *Domain Hop* (DH) counter, counting the number of times it moves between domains. This counter is sent to the MGM in every *join/leave/move* message, and is stored along with the MN's UID at the MGMs. An MGM that receives a move message with a lower DH than the one associated with the UID in its current view ignores this message.

The MN stores the DH as long as it is up. In case the MN re-joins a group after a crash, it registers with DH=0, which indicates to the MGM to send back, when available, the highest DH value associated with this MN.

MGMFlood is simple and allows for seamless handoff due to its prompt reaction to mobility updates. However, it entails high control message overhead, as all MGMs keep views of all groups, including groups not residing in their domains. This solution may not scale well, especially if there are many small groups and localized memberships.

4.4 MGMLeader

Our second solution reduces the overhead by propagating updates only to those MGMs that have group members in their domains. When an MGM receives an MN's message regarding a group that is represented in its domain, it extracts the MGMs that have members in the group from its local view, and forwards the message only to those MGMs.

If an MGM receives a control message (*join* or *move*) for a group that does not yet exist in its domain, then it needs to discover the group's up-to-date view, and to forward the event to the appropriate MGMs. The challenge is preserving a coherent view at all MGMs in the presence of concurrent operations without inducing excessive overhead.

In order to minimize the control overhead and ensure view consistency, only one of the participating MGMs sends the view to the new MGM. To this end, one MGM is designated as the *coordinator* of the group. Every active group has a *coordinator*, and a single MGM can be the coordinator of multiple groups. If the *coordinator* fails or leaves the group (because all the MNs in its domain leave) then a new *coordinator* is elected, as explained below.

When a new MGM joins a group due to a *move* event, it extracts the moving MN's former MGM from the *move* message, and sends the event message to that MGM. The former MGM, in turn, forwards the message to the *coordinator*. When the *coordinator* receives a *move* message originating from an MGM that is not already in the group, it sends the group's view to the new MGM and forwards the message to all the group's MGMs. This message flow is illustrated in Figure 2.

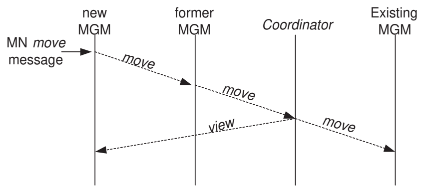

Figure 2. *Move* message flow.

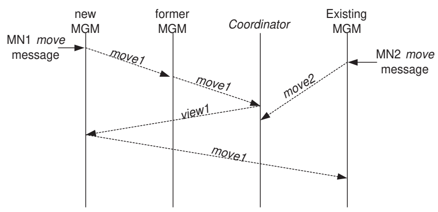

Figure 3. Potential view inconsistency in over-simplified leader-based solution.

This communication between the two MGMs also facilitates establishing a tunnel from the former MGM to the new one, so that the former MGM can forward data packets destined to the moving MN via its new MGM, to guarantee smooth handoff; such tunneling is suggested in [11].

When a new MGM joins a group due to a *join* message, it broadcasts the message to all the MGMs. As before, when the *coordinator* receives this message from the new MGM, it sends the group's view to the MGM.

As with the previous solution, MGMs may receive certain event messages out of order. The ordering of events related to the same MN is solved, as in the MGMFlood protocol, using the DH counter. However, this does not suffice to ensure view consistency when MGMs dynamically join and leave the group. Figure 3 illustrates a problematic scenario that can occur if concurrent joins are handled carelessly. In this example, while a new MGM retrieves the group's view from the *coordinator*, an existing MGM sends another event to the group's MGMs. The existing MGM is unaware of the new MGM and thus does not forward the message to it. This causes the new MGM to an inconsistent view of the group.

In order to address this difficulty, each MGM maintains an increasing *Local Event Counter* (LEC) for every group. Whenever an MGM receives a *join*, *leave*, or *move* message from a local MN, it increments the appropriate LEC. The group's LEC is included in every message pertaining to this group sent by the MGM. When an MGM joins a group, it initiates the group's LEC to 1. In addition, the MGM keeps, for every group, a *LECvector*, holding the highest known LEC for each MGM in this group.

In every message sent from one MGM to another, both the sender's LEC and the receiver's latest known LEC (from the *LECvector*) are included. When an MGM receives a packet, it checks the LECs. If its local LEC is higher than the one known to the sender it sends back its local view and LEC. If it discovers that the sending MGM's LEC is higher than the one it knows, it retrieves the local view of the sending MGM. When the *coordinator* forwards *move* messages of new MGMs, it includes the LECs corresponding to the view it is sending to the new MGM. In case some events are not reflected in this view, the receiving MGMs forward their local views to the new MGM. In addition, we ensure consistency using periodic updates, whereby the group's *coordinator* sends the current view all to the group's MGMs.

We now discuss coordinator election. When the last MN in the *coordinator*'s domain leaves the group or moves to another domain, the *coordinator* appoints a new MGM as the new *coordinator* of the group and informs the group's MGMs of the new *coordinator* in the forwarded *move* or *leave* message. Subsequently, the leaving *coordinator* forwards control messages that it still receives to the new *coordinator*. In order to avoid appointing an MGM that has already left, an MGM can not leave the group until it receives the *coordinator*'s permission. If the *coordinator* notices, after receiving a *leave* or *move* message, that an MGM has no members in the group, it sends a permission-to-leave message to that MGM. The only scenario where the *coordinator* does not permit the MGM to leave is if the *coordinator* appoints the MGM to be the new *coordinator*. In this case the MGM needs to find a new MGM to replace it as the group's *coordinator*.

In case the *coordinator* crashes, the surviving MGMs start an election procedure by flooding their local views to all MGMs in the system. The MGM with the lowest ID is elected as the new *coordinator*, and it sends to the group's MGMs an up-to-date view computed using the local views sent during the election procedure. Throughout the election procedure, all new events are buffered by the MGMs, and are disseminated only after the new view is received.

4.5 Dynamic MGMs

Thus far, we have assumed that MGMs are static and well-known. However, our solution can be extended to support a dynamic architecture, where MGMs can join and leave. The MGMLeader is a natural choice for such an architecture, since it accommodates for a dynamic set of MGMs maintaining the membership of a single view.

We plan to extend the architecture to support a delegation mechanism, where MGMs can grant MNs permission to operate as membership servers. This will allow the use of our group services in wireless hybrid networks, i.e. islands of ad-hoc network interconnected via access-points [3].

5. MGM Protocol Evaluation

5.1 Packet delay evaluation

Mobile IP may exhibit poor performance due to its use of triangle routing. With the MGM architecture, on the other hand, packets are sent directly to the MN's current location. To illustrate the advantage of this approach, we simulate a single constant bit rate (CBR) UDP session, and measure the end to end delay with both approaches (MGM and Mobile IP).

We simulate a network of four domains. The transmission source is in Domain 0 throughout the simulation. The receiver is initially in its home domain (Domain 1) and then moves towards Domain3 through Domain2, as depicted in Figure 4. The domains are connected via 5Mb links with 20ms delay. The simulated wireless interface is 914MHz Lucent WaveLAN DSSS.

Figure 5 shows the average measured packet delays for three architectures: Mobile IP, a centralized architecture with a single MGM (in Domain 0) servicing all MNs, and a distributed architecture, with an MGM in each domain. The MGM solutions transmit messages via the optimal route, whereas Mobile IP uses the triangle route, degrading performance by a factor of 3.

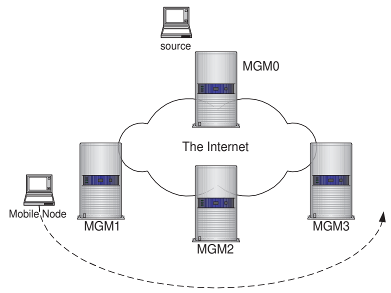

Figure 4. Scenario simulated in Figure 5.

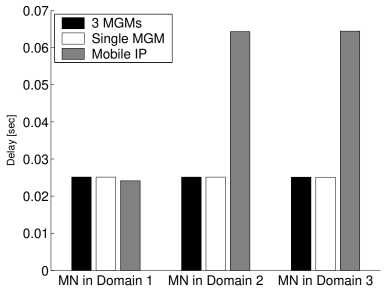

Figure 5. Average end-to-end packet delay for system depicted in Figure 4: simulation.

5.2 Control plane evaluation

We now evaluate the overhead associated with control messages. We measure the average control overhead associated with a single *move* event. We simulate the following uniform network model:

- 11 domains (Domains 0-11), 1 MGM in each domain;
- 10-100 receiving MNs, initially uniformly distributed in Domains 1-10, then moving among these domains;
- 8 groups, where every receiving MN participates in a single group chosen uniformly at random;
- a fixed number of sources in Domain 0.

We simulate the MGMFlood and MGMLeader protocols in this setting, and measure the average control overhead associated with a single *move* message. The average is calculated over 300 events for each number of MNs. In each event, a random MN moves to a new random domain. The results are depicted in Figure 6, with 95% confidence intervals for MGMLeader. We also mathematically analyze the expected control overhead. For MGMFlood, this is straightforward. Since each control message is sent to all MGMs, and there are ten MGMs, the overhead is exactly ten messages per *move* event. Not surprisingly, the analysis and simulation results for this protocol accurately match each other (see Figure 6).

For MGMLeader, our analysis provides an upper bound and a lower bound (both depicted in Figure 6). Recall that a new MGM joining a group communicates with the MN's former MGM, which forwards the message to the *coordinator*. The lower bound (*coordinator* case) occurs if the former MGM is the *coordinator* of the retrieved groups. The upper bound (non-*coordinator* case) occurs when the former MGM is not the group's *coordinator*. In this

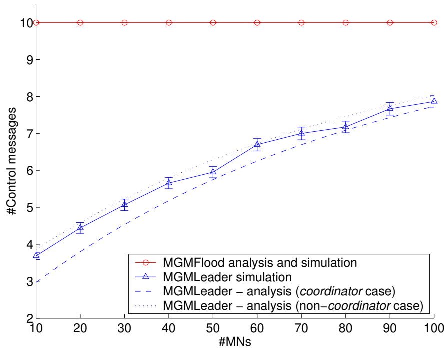

Figure 6. Average number of control messages per movement, uniform system, varying number of MNs, fixed number of domains: analysis vs. simulations.

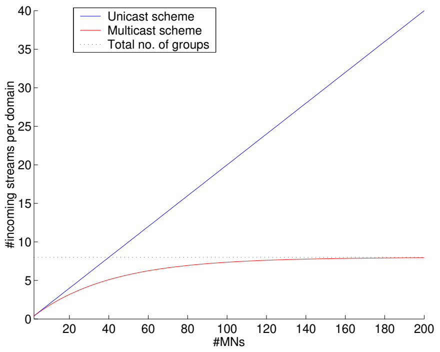

Figure 7. Average number of incoming streams per domain in uniform system with 8 groups, varying number of MNs, fixed number of domains: analysis.

case, one more message is sent compared to the *coordinator* case: from the former MGM to the *coordinator*. Due to space consideration, the detailed analysis derivation is given in the full paper.

We observe that with MGMLeader, the overhead increases with the number of MGMs that have members in the group. In sparse groups, few MGMs are involved, and hence few control messages are sent. MGMLeader outperforms MGMFlood in all the simulated situations, but its advantage is less significant in dense groups. We conclude that MGMLeader is preferable for sparse groups, whereas the much simpler MGMFlood may be adequate for dense groups in which all or most MGMs participate.

6. Transport Issues

We suggest two solutions for *multicast*. The first solution uses unicast: the transmitting MN *retrieves* the group's view and sends data directly to all the group members using multiple unicast streams. This solution incurs minimal load on MGMs, but suffers from poor scalability as domains with many users will have many incoming streams.

The second solution uses multicast over the MGMs' overlay. The transmitting MN sends the data to its MGM, which forwards the data to all the group's MGMs, which in turn forward the data to their locally subscribed MNs. This solution can exploit IP Multicast where available. It is more scalable, and enables each MN to use a single stream, thus extending its battery life.

Figure 7 depicts the number of incoming streams per-domain in both solutions, analyzed for the uniform network model of Section 5. We assume that all groups are active. In the unicast scheme, the average number of incoming data streams is the average number of MNs in the domain, whereas in the multicast scheme, it is the average number of groups in the domain. The detailed analysis derivation is given in the full paper.

7. Conclusions

We have presented MaGMA, an architecture for supporting group services in the emerging wireless networks. We presented and evaluated group management protocols for this architecture. MaGMA provides a comprehensive solution for seamless mobility with QoS support, important requirements that are not addressed in current solutions. MaGMA can be incrementally deployed since an MN may communicate with any MGM located in its vicinity.

References

[1] T. Anker, D. Breitgand, D. Dolev, and Z. Levy. CONGRESS: connection-oriented group address resolution services. *SPIE on Broadband Networking Technologies*, 1997.

[2] A. Bartoli. Group-based multicast and dynamic membership in wireless networks with incomplete spatial coverage. *Mobile Networks and Applications*, 3(2):175–188, 1998.

[3] E. M. Belding-Royer, Y. Sun, and C. E. Perkins. Global Connectivity for IPv4 Mobile Ad hoc Networks. Internet draft, IETF, draft-royer-manet-globalv4-00.txt, Nov. 2001.

[4] Y. Chawathe. Scattercast: an adaptable broadcast distribution framework. *Multimedia Syst.*, 9(1):104–118, 2003.

[5] E. Gustafsson, A. Jonsson, and C. E. Perkins. Mobile IPv4 Regional Registration. Internet draft, IETF, draft-ietf-mobileip-reg-tunnel-06.txt, Mar. 2002.

[6] J. Mysore and V. Bharghavan. A new multicasting-based architecture for internet host mobility. In *ACM/IEEE Conf. on Mobile Computing and Networking*, pp. 161–172, 1997.

[7] Nextel. http://www.nextel.com/services/directconnect.shtml.

[8] ns2 homepage. http://www.isi.edu/nsnam/ns.

[9] Open Mobile Alliance. http://www.openmobilealliance.org.

[10] C. E. Perkins. IP Mobility Support for IPv4. RFC 3344, IETF, Jan. 2002.

[11] C. E. Perkins and D. B. Johnson. Route Optimization in Mobile IP. Internet draft, IETF, draft-ietf-mobileip-optim-11.txt, Sep. 2001.

[12] R. Prakash and R. Baldoni. Architecture for Group Communication in Mobile Systems. In *17th IEEE Symposium on Reliable Distributed Systems*, pp. 235–242, 1998.

[13] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli. IP micro-mobility support using HAWAII. Internet draft, IETF, draft-ietf-mobileip-hawaii-01.txt, Jan. 2000.

[14] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *6th Int'l Conf. on Mobile computing and networking*, pp. 155–166, 2000.

[15] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. OverQoS: offering internet QoS using overlays. *SIGCOMM Comput. Commun. Rev.*, 33(1):11–16, 2003.

[16] The Yankee Group. NG Push-to-Connect Provides Simple UI and Enriches User Experience, Sep. 2003.

[17] A. G. Valkó. Cellular IP: a new approach to internet host mobility. *SIGCOMM Comput. Commun. Rev.*, 29(1):50–65, 1999.

[18] Verizon Wireless. http://news.vzw.com/news/2003/08/pr2003-08-14.htm.

[19] WinterGreen. Push To Talk (PTT) Market Opportunities, Market Forecasts, and Market Strategies 2003-2008, Dec. 2003.