A Multicast-based Protocol for IP Mobility Support

Ahmed Helmy
Assistant Professor of Electrical Engineering
University of Southern California
3740 McClintock Avenue, EEB 232
Los Angeles, CA 90089-2562
Tel (213) 821-1329
helmy@ceng.usc.edu

ABSTRACT

Several architectures have been recently proposed to support IP mobility. Most studies, however, show that current protocols, in general, fall short from satisfying the performance requirements for audio applications. In this study, we propose a multicast-based protocol to reduce latency and packet loss during handoff and provide the base for IP mobility support. We use extensive simulation to evaluate our protocol's performance over a variety of real and generated topologies, and we compare it to several other mobility protocols, especially the Mobile IP protocols. We take handoff delay estimates and routing efficiency as metrics for our comparisons. We take a route analysis (as opposed to packet analysis) approach in our study, and we apply it in the context of wide-area networks.

Our simulation results show significant improvement for our proposed protocol. On average, basic Mobile IP consumes almost twice as much network bandwidth, and experiences more than twice as much end-to-end and handoff delays, as does our proposed protocol. In addition, average handoff delay estimates for our protocol prove to be less than that for other protocols in this study, even with route optimization. We further propose an extension to Mobile IP to support our protocol with minimal modification.

Keywords

Mobility, Multicast, Efficient Handoff, Network Simulation.

1. INTRODUCTION

The recent advances in wireless communication technology and the unprecedented growth of the Internet have paved the way for wireless networking and IP mobility. Unlike conventional wired networks, wireless networks possess different channel characteristics and mobility dynamics that render network design and analysis more challenging. Performance during *handoff* - where the mobile moves from one cell, or coverage area, to another - is a significant factor in evaluating wireless networks. In addition, route efficiency is a measure to evaluate the impact of the mobility architecture on the network, in terms of resources consumed, or overhead incurred.

IP-multicast provides efficient algorithms for multiple packet delivery. It also provides location-independent group addressing. The receiver-initiated approach for IP-multicast enables new receivers to join to a nearby branch of an already established multicast tree. Hence, IP-multicast provides a scalable infrastructure for efficient, location-independent, packet delivery.

In our study, we merge advantages of using IP-multicast with IP mobility to design our protocol. Earlier studies [1,2] have suggested that using multicast principles may improve performance of IP mobility. However, these studies did not quantify this improvement in the wide-area network or Internet contexts. We design and present our multicast-based protocol for mobility support according to clear design requirements, and evaluate its performance with respect to these requirements using large-scale simulations. We focus on performance during handoff and route efficiency as the primary metrics for evaluation.

The intuition behind our proposal is simple. As the mobile node roams across the network, we want packets destined to it to *follow it* throughout its movement. Imagine a dynamic distribution tree with branches reaching all locations visited by the mobile during its journey. These branches constitute the shortest paths from the packet source to each of the visited locations. The tree is dynamic such that the branches grow and shrink to reach the mobile node as necessary, when necessary. This architecture is realized by having the correspondent node send packets to a multicast address, to which the mobile node joins from each location visited. In this paper, we describe the mechanisms involved in realizing such architecture. In addition, we analyze the performance of these mechanisms through simulation and compare it to Mobile IP [3,4].

Our simulation results show that, on average, basic Mobile IP consumes almost twice as much network bandwidth, and experiences more than twice as much end-to-end and handoff delays, as does our proposed protocols. In addition, on average, handoff delays experienced by our protocol are significantly less than those experienced by other (non-multicast-based) protocols considered in this study, including Mobile IP with route optimization. For systems already using Mobile IP with route optimization, we propose a minimal extension to support the architecture presented in this paper. The multicast address assigned to the mobile node can be sent to the correspondent node through a *binding update* in the start-up phase of communication.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 outlines the design requirements. Architectural overview is presented in Section 4, and protocol description is given in Section 5. In Section 6 we present our performance evaluation, while Section 7 discusses design issues and alternatives. We conclude in Section 8 and present directions for future work.

2. RELATED WORK

Several architectures have been proposed to provide IP mobility support. Work by the IETF on Mobile IP (MIP) is given in [3]. In MIP a mobile node (MN) is assigned a permanent *home* address and a *home agent* (HA) in its home subnet. When the MN moves to another *foreign* subnet, it discovers a foreign agent (FA) on that subnet and acquires a temporary care-of-address (COA) through a solicitation/advertisement message exchange with the FA. The MN informs the HA of its COA through a *registration* process. From that point on, packets destined to the MN's home address are sent first to the home network, are picked up by the HA and then are *tunneled* to the MN through the FA. This is known as the *triangle routing* problem, which is the major drawback of the basic MIP.

A proposed mechanism, known as *route optimization*, attempts to avoid triangle routing. In [5] route optimization is achieved by sending *binding updates*, containing the current COA of the MN, from the HA to the correspondent node (CN). In MIPv6 [4] binding updates are sent from the MN to the CN with every move. Although this alleviates the triangle routing problem in MIP, the communication overhead is still high during handoff rendering MIP unsuitable for *micro* mobility and causing it to be inadequate for audio applications.

Caching techniques are proposed in [4] to reduce packet loss during handoff. These techniques, in general, cause the old FA (before the move) to forward cached packets to the new FA (after the move) to recover packets that would otherwise be lost during the transition. The old FA needs to know the new COA of the MN before forwarding the cached packets; hence this technique still incurs handoff latency and results in out-of-order packets.

A hierarchical mobility management scheme was proposed in [6] that defines three hierarchical levels of mobility; local, administrative domain and global mobility. This scheme proposes to use MIP for the global mobility, while using *subnet foreign agents* and *domain foreign agents* for the other levels. It is not clear, however, how this hierarchy will be formed or how it adapts to network dynamics, partitions or router failures. The above study describes, implements and evaluates the local handoff protocol on the same subnet.

In [7] an end-to-end architecture is proposed for IP mobility, based on dynamic DNS updates. Whenever the MN moves, it obtains a new IP-address and updates the DNS mapping for its host name. A migration process is required to maintain the connection. The transport protocol is aware of the mobility mode during the migration process. Such architecture avoids triangle routing. However, we believe that such architecture incurs similar handoff delays to those experienced in MIPv6 (which we analyze in this study), or even worse due to DNS update delays and migration delays. The end-to-end approach is geared toward TCP-based applications, but we feel it is not suitable for real-time multimedia applications (such as audio) with stringent delay and jitter bounds.

The Daedalus project [2] proposes to tunnel the packets from the HA using a pre-arranged multicast group address. The base station, to which the MN is currently connected, and its neighboring base stations (BSs), join that group and get the data packets over the multicast tree. Using beacons and signal strength measurements, the MN determines which BS should join the group and to which BSs it is likely to move in the near future. Advance buffering is used to achieve very low latency and reduced data loss. Experiments in a limited topology show that very low latency (< 15ms) and no data loss can be achieved up to 3-hop distance between BSs. It is not clear how the scheme performs in larger wide-area topologies. This approach suffers from the triangle routing problem; packets are sent to the HA first and then to the MN.

An approach for providing mobility support using multicast (MSM-IP) is presented in [1]. In this approach, each MN is assigned a unique multicast address. Packets sent to the MN are destined to that multicast address and flow down the multicast distribution tree to the MN. This is similar, in concept, to the Daedalus project approach. However, it is not the HA that tunnels the packets using the multicast address, rather, it is the CN that sends packets directly to the multicast address. This approach avoids triangle routing, in addition to reducing handoff latency and packet loss by potentially using advance buffering. Experiments for MSM-IP were performed in a limited testbed. More work is needed to measure the protocol performance in larger networks. There are architectural differences between our approach and MSM-IP. For example, we propose a start-up phase that is a minor modification to Mobile IP to implement our protocol. In MSM-IP, a hierarchy of servers is proposed for location management. Such hierarchy is complex, susceptible to failures, and imposes restrictions of placement of the Rendezvous Point in PIM-SM as an underlying multicast protocol. In addition, by using binding updates and using the destination option in IPv6, we avoid potential MSM-IP problems with TCP (and other protocols) due to the use of multicast addresses for the MN. The multicast address is used within the network for packet routing, but the applications are only aware of the permanent unicast home address of the MN. Hence, no change to the application or transport protocol is needed.

In this paper, we present our multicast-based protocol for supporting IP mobility. We avoid triangle routing and caching drawbacks suggested by the MIP architectures. Also, we avoid creating our own hierarchy of agents. Rather, we re-use the existing hierarchy and infrastructure of wide-area and inter-domain multicast routing [8,9]. We leverage off of some ideas offered by the Daedalus project and MSM-IP to build our architecture and evaluate it in a wide-area networking context.

3. DESIGN REQUIREMENTS

The main requirement of the proposed protocol is to provide an efficient mechanism for IP mobility. Mobility support should be transparent to applications, and should provide adequate mechanisms for security. In addition, the protocol should provide smooth handoff, efficient routing in terms of end-to-end delays and consumption of network resources, and should conserve wireless bandwidth. Deployment of the protocol should be applicable to the Internet under reasonable assumptions. Note that we strive to achieve these requirements with least protocol complexity and least changes to the network infrastructure.

3.1. Transparent Mobility Support

Protocols that provide support for IP mobility are intended to enable seamless mobility as the mobile nodes move continuously throughout heterogeneous networks, changing their point of attachment frequently. These protocols should allow a mobile node to communicate with other nods with tolerable (preferably minimum) service disruption. The mode of communication during movement should be transparent to higher-layer transport protocols and applications. In general, to avoid changing any of the applications, these applications should be aware of a single (non-changing) identity for the MN, such as, a permanent unicast home address.

3.2.Performance Requirements

Our protocol should meet the following requirements.

(a) *Smooth handoff*: A general requirement for IP mobility is to provide 'smooth' handoff. Handoff *smoothness* can be measured by several criteria, such as delay, jitter, data loss and communication overhead during handoff. The efficiency of handoff depends heavily on the specific application. For example, audio in general is tolerant to loss, but has stringent delay requirements. Handoff delay is a function of the number of hops added to the data path during handoff[1]. We strive reduce handoff latency for much of the operating conditions studied.

(b) *Efficient routing*: One major drawback of the Mobile IP protocol is triangle routing, where packets from the correspondent node travel to the home agent (in the mobile's home network) before being tunneled to the mobile node. This approach: (i) consumes a lot of network resources, (ii) is more susceptible to network partition and (iii) degrades the performance perceived by the end applications. It further complicates the handoff. In our mechanisms we attempt to avoid the above drawbacks. Routing efficiency may be measured as the number of hops traversed by data packets from the correspondent node (CN) to the mobile node (MN), as well as the overall network bandwidth consumed due to the mobility architecture.

(c) *Low waste of network and RF bandwidth*: Since the MN may often be connected to the Internet via a wireless connection, such as radio frequency (RF) links, where link bandwidth and device power are scarce, mobility support protocols should be designed to conserve bandwidth and MN energy. Some approaches attempt to minimize delay and loss during handoff by multiple packet forwarding. Hence, these approaches are likely to waste more bandwidth than others are. We design protocol mechanisms that conserve network bandwidth in general, and reduce wasted RF bandwidth during handoff. In addition, the communication overhead should be reduced by the MN to conserve power.

3.3.Security

Security is always an issue when designing protocols for the Internet. More so for mobility support, where the continuous movement and change of attachment point is part of the normal operation. Such setting is prone to *remote redirection* attacks, where a malicious node redirects to itself packets that were originally destined to the mobile node. Authentication should be used with any message revealing information about the mobile node. Another concern is hiding the location of the mobile node. A mobile node may desire to conceal information about its current location. In Mobile IP with route optimization, the new care-of-address is revealed to the home address and/or the correspondent node with every move. In our proposed protocol, however, the MN is assigned a location-independent multicast address that does not change with movement, and hence does not reveal information about the MN's current point of attachment.

[1] For our architecture, this will be the number of links added from the new location to establish a branch from the existing multicast delivery tree.

3.4.Applicability

Our architecture assumes multicast capability in the Internet. Our simulations use the Protocol Independent Multicast-Sparse Mode (PIM-SM) [8] as the multicast routing protocol. Our architecture also assumes that each mobile node is assigned a multicast identifier. This identifier is typically a multicast address. For IPv6, we do not expect this to be a problem[2].

4. ARCHITECTURAL OVERVIEW

In order to provide mobility, the packets sent to the mobile node (MN) need to be forwarded to every location visited by the MN. Forwarding takes place according to the temporal and spatial pattern of movement. One may view this problem as follows. The set of locations that the mobile will visit may be viewed as a *group* of receivers, to which the packets should be delivered. The temporal *pattern* by which packets are delivered to these receivers represents the temporal component of the movement.

Instead of sending their packets to a unicast address, nodes wishing to send to the MN send their packets to a *multicast group* address. The MN, throughout its movement, would join this multicast group through the locations it visits. Because the movement will be to a geographical vicinity, it is highly likely that the join from the new location (to which the mobile has recently moved) will traverse a small number of hops to reach the multicast distribution tree (already established to the previous location of the mobile node). Hence, performance during handoff (in terms of latency and packet loss) will be improved drastically. In this section, we discuss how the main components of our architecture interact to realize the desired behavior.

4.1.Dynamics of packet distribution

Optimally, the packets destined to MN should be delivered over the shortest path, traversing the minimum number of links and experiencing minimum delay. The packets should traverse only those links that lead to the current location of MN, but also should be delivered to the new location with minimum handoff latency. This may be achieved by redirecting the packet delivery tree (of the old location) to grow in the direction of minimum distance to the new location. That is, a branch should be established from the new location of the MN to the nearest point of the delivery tree. This is illustrated in Figure 1. As MN continues to move, branches are established to deliver packets to the new location, while other branches, those that no longer lead to the MN, are torn down and pruned.

4.2.Establishing the delivery tree

Initially, the MN is assigned a multicast address *'G'*. The CN sends its packets to *G.* To establish the delivery tree from the CN to the MN, the MN sends a source-specific *(CN,G)* join message towards the CN[3], as in Figure 2 (a). As the MN moves and connects to another location[4], it joins *(CN,G)* through the new location, as in

[2] For IPv4, however, this may prove to be problematic due to the limited multicast address space (class D IP-addresses). This is still an open issue under investigation.

[3] We assume that the multicast protocol allows explicit joining and pruning and supports source-specific trees, similar to PIM-SM[8] or BGMP[9] capabilities, and that the host interface allows for source-group specific joins, as in IGMPv3 [10].

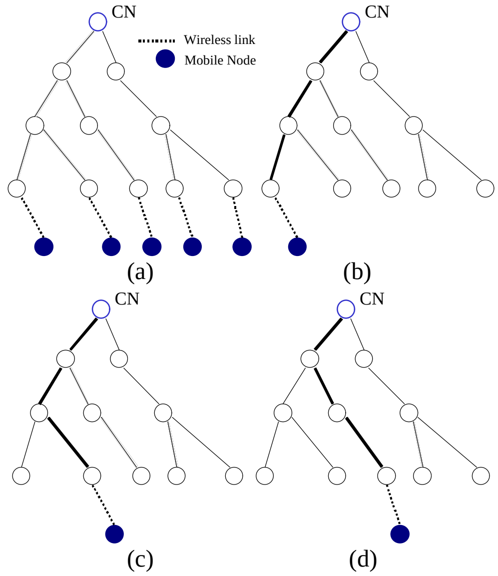


Figure 1. (a) All locations visited by MN are considered part of the distribution tree, (b) when a mobile moves to a certain location, only that location becomes part of the tree (shown by bold lines). When the mobile moves to a new location, as in (c) and (d) the tree changes to deliver packets to the new location.

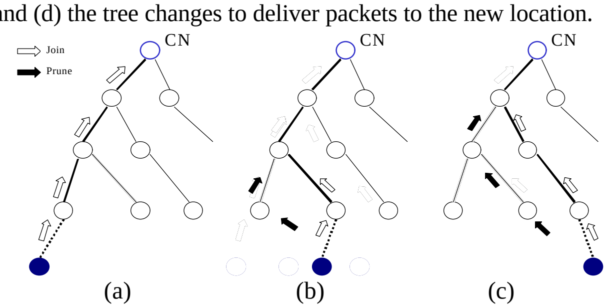


Figure 2. The distribution tree adapts to movement of the MN: (a) MN joins towards CN. As the MN moves, as in (b) and (c), the MN joins the distribution tree through the new location and prunes through the old location.

Figure 2 (b). The join is forwarded upstream (according to the multicast routing protocol) towards CN. When the join reaches the nearest point of the multicast tree, the join process is complete and the packets start flowing down the newly established branch towards the MN. Upon receiving data packets from the new location, the MN issues a prune message to the old location. The prune message tears down the branches that no longer lead to the MN[5].

4.3.Smooth Handoff

In general, at any point in time, the MN should accept packets from only one location. However, during transient movement, the MN may be joined to *G* through multiple locations. The dynamics of joining and leaving/pruning *G* during handoff directly affect handoff latency and smoothness. To allow a smooth handoff, the MN should not prune the old location until/unless it starts getting packets from the new location. This is illustrated in Figure 3. To further ensure smoothness and to conserve RF bandwidth, mechanisms should be designed to reduce join and leave latencies. We discuss this issue in Section 7.

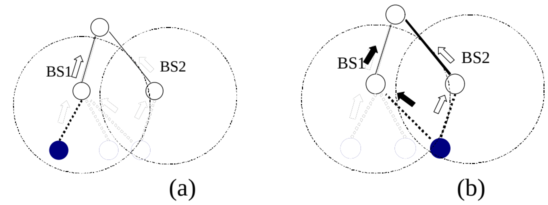


Figure 3. Smooth handoff between neighboring cells: (a) MN joins through BS1, (b) as the MN moves into BS2's covering region it joins through BS2, when it starts getting packets from BS2 it sends a leave/prune through BS1.

4.4.Start-up Phase

In order to send packets to the multicast address of the mobile node, the correspondent nodes need to obtain this multicast address. One may argue that such an address may be obtained in a way similar to obtaining the home address of a mobile node in the conventional Mobile IP (e.g., through DNS lookup, or otherwise)[6]. However, we do not assume such a service to be available, but provide an alternative design, one that requires minor modification to MIPv6 [4]. We propose that the multicast address be conveyed to the CN by the MN using *binding update*. However, unlike MIPv6, the binding update occurs only once during the initial establishment of communication, not with every move[7].

5. PROTOCOL DESCRIPTION

The main entities of our protocol are the mobile node (MN), the correspondent node (CN), the base station (BS), and (if needed) the home agent (HA). Foreign agents (FAs) are not required in our protocol. Most of the protocol mechanisms are implemented in the MN and the BS. Following is a description of the main mechanisms needed in each of these entities.

5.1.Mobile Node (MN)

The mobile node implements mechanisms for join and leave, movement detection, binding updates, and obtaining care-of-address.

(a) Join/Leave: The MN must run a membership protocol that enables it to inform the first hop router (or BS) to join and leave the multicast group at various points in time. This mechanism should support Join/Leave messages with explicit source list, and should also support the option to specify joining to the source-specific tree (SPT) directly (without going through a shared tree).

---

[4] By location, we mean a new point of attachment that leads to change in the first hop router. For simplicity, without loss of generality, we assume that each base station is a router.

[5] If upon receiving the first few packets from the new location the MN cannot communicate with the old location, then the prune should be triggered through another implicit mechanism, such as a time-out mechanism.

[6] Note that if such service exists, our protocol may be simplified drastically, where the home agent (HA) is no longer needed, nor is binding updates, encapsulations or decapsulations.

[7] Our initial design does not require the functionality of the home agent (HA) or foreign agent (FA). However, we feel it is sometimes desirable to have the HA for accounting and security.

The MN should always join to the home agent (HA) to get packets from new sources. To reduce join latency, we recommend that Join messages be acknowledged. Hence, the MN needs to set 'ack' timer for each Join sent. If the timer expires before the reception of a Join-ack, the Join message is re-sent. A Join message may contain a list of sources, to which the MN wishes to join.

(b) Movement Detection: The base stations (BSs) send beacons periodically within their coverage area. The MN receives these beacons from nearby BSs and measures their strengths. Based on these measurements, the MN determines to which BS it is likely to handoff in the future. In general, we assume that the MN will not change its primary BS more than a few times a minute. However, hysteresis may still be used to dampen the handoff frequency. The MN sends a message for the future BS to join the group and start caching the data packets intelligently. When the MN finally moves into the cell covered by the new BS, it issues a *cache-forward* message to the new BS with the ID of the last packet the MN has received. Upon reception of the data packets from the new BS, the MN issues a Leave message to the previous BS. The MN also gets a temporary unicast care-of-address (COA) address for the new network. The unicast COA is used by the MN to send unicast packets.

(c) Binding updates: During the startup phase, where the CN presumably has only knowledge of the MN's unicast home address, the first few packets are sent to the home network. The home agent (HA) uses proxy ARP to get the packets and encapsulate them (in multicast packets) to the MN. When the MN receives the packets encapsulated by the HA it triggers a *Binding update* to the CN containing the MN's multicast address. (The binding updates may be acknowledged or just triggered by packets encapsulated by the HA. If triggered then a rate-limit mechanism should be used to reduce the number of *Binding update* messages sent by the MN.) Normally, a binding update is sent only once per CN during the start-up phase of communication (unlike MIPv6, in which binding updates are sent with every move).

(d) Care-of-address: When the MN moves into the cell coverage of a new BS, it requests a care-of-address (COA), using either auto-configuration, DHCP, or others. The MN uses COA to send packets. This is necessary for two reasons. First, for unicast packets, *ingress filters* may reject packets with the MN's home address in the source field, when the MN is not in the home network. Ingress filters are used to combat denial-of-service attacks. Second, for multicast packets, the reverse path forwarding (RPF) check may fail at a router if the interface upon which the packet arrived does not lead to the home network. Hence, the home address cannot be used to source data packets from a MN at a foreign network.

5.2. Base Station (BS)

The base station implements mechanisms for join and leave, caching and forwarding, and sending beacons. In addition, we propose an election mechanism to increase robustness to BS crashes.

(a) *Join/Leave*: The base station connecting to the MN must implement a membership protocol to be able to accept and process Join and Leave messages sent by the MN. The BS should be able to send Join-ack messages in response to Join messages. It should also implement a multicast routing protocol, to be able to construct the distribution tree to the desired senders. If the BS loses link-layer connectivity with the MN, it should not forward packets to the MN over the wireless channel. Rather, it should go into caching mode for a specified period of time, after which the cache and the entry are discarded (as if a Leave message was received from the MN), if no further messages were received from the MN.

(b) *Caching and forwarding*: When the BS gets a Join message from the MN, the join may also contain a request to *start caching*. This usually occurs in the advance-join scenario, where the BS establishes a branch to the multicast distribution tree, and caches the data packets in anticipation of MN movement to its cell. Upon receiving a *forward* request from the MN, the BS forwards the cached packets that have IDs higher than that included in the forward request. From that point on, data packets are forwarded normally from the BS to the MN, until the MN issues a Leave message or is disconnected from the BS.

(c) *Sending beacons*: Beacons are messages sent by the BS to its coverage area, to notify MNs in the coverage area of its signal strength. Beacons facilitate the movement detection and smooth handoff of the MN.

(d) *Election*: A primary BS is elected dynamically to be responsible for MNs in the coverage area. If the BS fails, another BS is elected to carry on its duties. The state (previously saved in the failed BS) should either be re-created (e.g., through new Join messages), or be replicated periodically and consistently.

5.3. Correspondent Node (CN)

The correspondent node implements a very simple mechanism (if start-up phase is required) for receiving and updating its binding cache.

Binding update reception: Upon receiving a binding update message from the MN, the CN updates its *binding cache* and sends a binding-ack (if required). A binding cache entry is created per MN, and contains the unicast home address and the corresponding multicast address of the MN. When a new data packet is sent to the MN, the CN checks if a binding cache exists for this MN. If so, then the packets are destined to the corresponding multicast address directly[8].

5.4. Home Agent (HA)

The home agent (HA) is required only if the start-up phase is required. HA performs encapsulation of packets to the mobile node. We also propose an election mechanism to increase robustness to HA crashes.

(a) *Encapsulation*: The home agent (HA) maintains a mapping entry for each of the MN's for which it is responsible. The mapping entry contains the home address and the corresponding multicast address of the MN. When the HA receives a data packet destined to the home address of the MN, it encapsulates the data packet, with its own address in the source field and the MN's multicast address in the destination field, and sends it to the MN.

(b) *Election*: For robustness, there should be more than one HA to avoid single point of failure scenarios. HAs on a LAN conduct an election to choose one primary HA for each MN. The mapping entries may be replicated for all HAs on a home network, however, such that each HA has full mapping table for all MNs that belong to this home network.

[8] IPv6 Routing Header may be used for this purpose.

name	nodes	links	avg. deg	name	nodes	links	avg. deg	name	nodes	links	avg. deg
r50	50	217	8.68	ts150	150	276	3.71	ts1008_3	1008	3787	7.51
r100	100	950	19	ts200	200	372	3.72	ti1000	1000	1405	2.81
r150	150	2186	29.15	ts250	250	463	3.72	ti5000	5000	7084	2.83
r200	200	3993	39.93	ts300	300	559	3.73	Mbone_1	3927	7555	3.85
r250	250	6210	49.68	ts1000	1000	1819	3.64	Mbone_2	4179	8549	4.09
ts50	50	89	3.63	ts1008_1	1008	1399	2.78	AS	4830	9077	3.76
ts100	100	185	3.7	ts1008_2	1008	2581	5.12	ARPA	47	68	2.89

Table 1. Topologies used in the simulation (r: flat random, ts: transit-stub, ti: Tiers)

6. PERFORMANCE EVALUATION

In this study, we examine the performance of our proposed protocol in the context of wide area networks. We also compare our protocol to various versions and variants of the Mobile IP protocol. We use simulation to perform the comparisons. The simulation scenarios consist mainly of the topology and movement models.

I. ***Topology:*** The topology model mainly defines the number of nodes and their link connectivity. Several methods may be used to generate simulation topologies. We have chosen three methods that we believe result in a good variety of topology samples. These three methods include two topology generators (GT-ITM[11], Tiers[12]) and a set of real measured topologies. Table 1 lists the topologies used in our simulations[9]. The topologies include 47 to 5000 nodes with different degrees of connectivity.

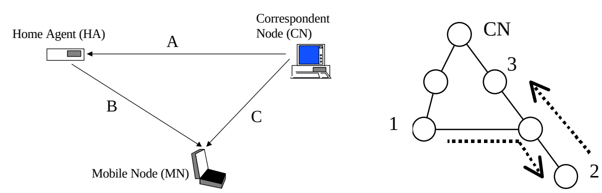
II. ***Movement:*** The movement pattern defines the sequence of nodes visited by the mobile node (MN) throughout the simulation. We have used three movement patterns as follows. The first allows the MN to visit any node in the topology randomly, we call this movement pattern *random*. The second allows the MN to visit only directly-connected nodes in the next movement step. The next neighbor is chosen randomly from the set of directly connected nodes. We call the second movement pattern n*eighbor*. The third movement pattern allows the MN to connect randomly to only one of 6 nodes that are likely to fall within the same cluster as the MN[10]. This movement pattern is called *cluster*.

6.1.Simulation

We use the VINT tool kit for simulation, including the network simulator (ns) [15]. The unicast routing we use is Dijkstra's algorithm, and the multicast routing protocol is the centralized PIM-SM [8] with the SPT switch to enable source-based trees[11].

For each simulation run, a correspondent node (CN) and a home agent (HA) are chosen randomly, and 100 nodes are chosen as movement steps. The 100 nodes are chosen according to one of three movement patterns defined above. These are the nodes visited by the mobile node (MN) during the movement. For each movement pattern, simulation is repeated 10 times with different random number generator seeds.

We simulate our proposed protocol and various versions of MIP; the basic MIP[3] and MIPv6 with route optimization[4]. We even propose an enhancement to MIPv6 in an attempt to reduce handoff latency. We call such approach the *previous location* approach, where packets are forwarded from the previous location to the new location with every move[12]. We compare these protocols in terms of route efficiency (bandwidth consumed and end-to-end delay) and estimated handoff delay. In this paper, we present and discuss our route analysis results.



(a) (b)

Figure 4. (a) Ratio ***'r'****= (A+B)/C.* **(b)** As the MN moves from node 1 to 2, added links *'L'* is 3 and links to previous location *'P'* (dashed lines) is 2. As it moves from 2 to 3, ***L****=0*, *P*=2.

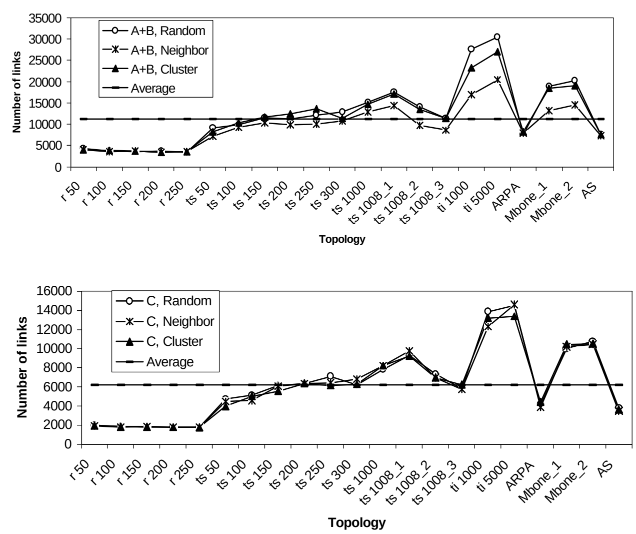


Figure 5. Links traversed in MIP 'A+B', and in our protocol 'C'.

---

[9] We included topologies used in previous studies on the Internet[13,14]. ARPA is the original ARPANET topology, the Mbone topologies are based on measurements by the SCAN project at USC/ISI, and the AS topology is provided by NLANR.

[10] This is especially meaningful in clustered topologies, where nodes are numbered sequentially in a cluster, so node numbering has geographical significance. We generally assume that this is a cellular system with 7 cell reuse architecture, where each node in the topology is a base station for a cell, hence the 6 node figure.

[11] In effect, the RPF source-based tree built by PIM-SM is the same as the unicast shortest path under assumption of path symmetry.

6.2. Simulation Results

We measure the total number of links traversed by data packets during the simulation, and the ratio between the paths[13]

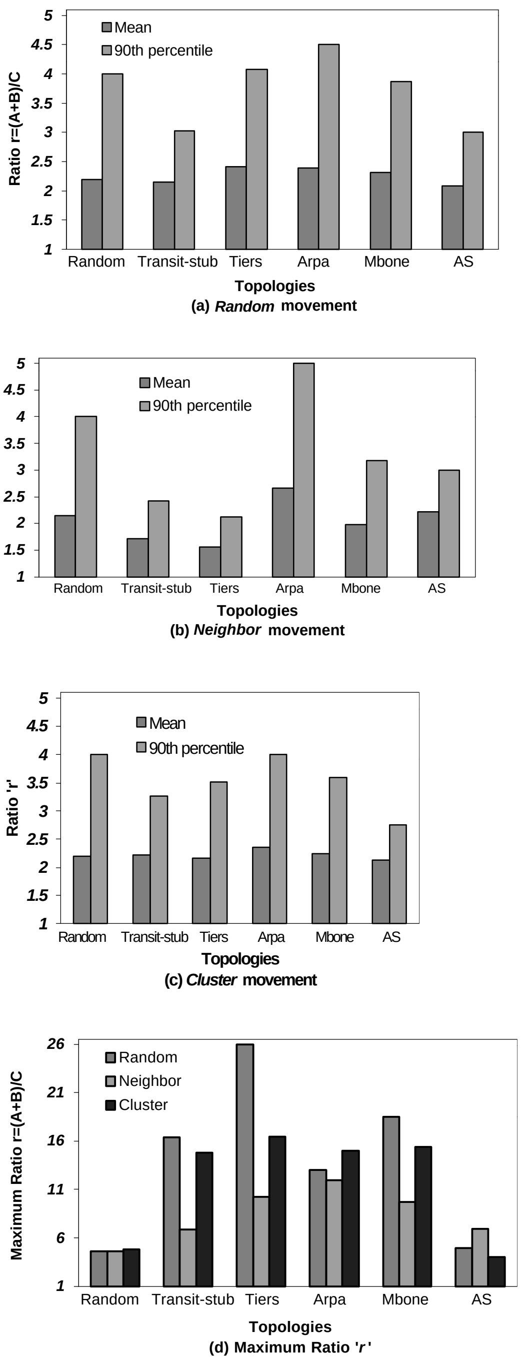


Figure 6. Ratio ***'r'*** for the different topologies and movement patterns.

taken by data packets in MIP triangle routing and those taken in our proposed protocol[14]. We call this ratio '***r***'. Figure 4 (a) shows ***r*** = *(A+B)/C*, where *A* is the unicast path (i.e., number of hops) from CN to HA, *B* is the unicast path from HA to MN, and *C* is the multicast path from CN to MN. We also measure, for our protocol, the number of hops added to the multicast distribution tree with every move. This number is denoted as '***L***', and is shown in Figure 4 (b). Also shown in Figure 4 (b) is the number of links to the previous location '***P***'. We measure the ratios '***B/L***', '***C/L***', and '***P/L***' as measures of the handoff latency between MIPv4, MIPv6, previous location and our protocol, respectively. Statistics are computed for 1000 samples (100 movements and 10 runs) for each topology, then averaged across same type topologies (e.g., random, transit-stub). We discuss the simulation results in detail in the rest of this section.

6.3. Network links traversed

The total number of links traversed by the data packets is one metric to measure the consumed network resources. Here, we present the total number of links traversed throughout our simulations. For our proposed protocol the total links are denoted by 'C', while the total links traversed in case of Mobile IP's triangle routing are denoted by 'A+B', both shown in Figure 5. In general, the total links traversed is quite similar for all types of movement models[15]. The average number of links traversed across all topologies in our architecture is 6,208 links (for 1,000 samples), while the average for triangle routing is 11,157. That is, on average, triangle routing for MIP consumes almost twice as much network resources as does a shortest path approach (such as MIP with route optimization or our proposed protocol) for the same scenarios.

6.4. Ratio '*r*'

For different topologies and movement patterns the ratio ***'r'*** is measured as the metric for routing efficiency. This ratio mainly compares the path lengths from the CN to the MN. This relates directly to the end-to-end delays experienced by the data packets, as opposed to the network bandwidth consumed represented by the total links traversed. The measurement results of the ratio '***r***' are given in Figure 6. The average ratio across topology types ranged from 1.56 to 2.42, while the 90th percentile point ranged from 2.13 to 5. The maximum ratio ranges from 4 to 26. The overall average of ***r*** is '2.11'. Hence, on average, end-to-end delay experienced by Mobile IP triangle routing is more than twice as much the delay experienced by our protocol (and similar shortest path approaches).

6.5. Added links '*L*'

In our proposed protocol, handoff delay is directly related to ***L***, since ***L*** represents the number of links traversed by the Join message to pull the data packets down to the new location. Simulation estimates for ***L*** are shown in Figure 7. The movement experiencing the least ***L*** is *neighbor* (with overall average of

---

[12] This is similar to always operating in *router-assisted smooth handoff* mode in MIPv6.

[13] The path length is measured in terms of number of hops.

[14] For network overhead and end-to-end delay, our protocol performs similar to MIP with route optimization. In general, this is a comparison between triangle routing and shortest path routing.

[15] Although it is a bit lower for the *neighbor* movement in case of Mobile IP, because it is the most constrained among the three movement models used.

1.18), then *cluster* (with average of 2.38), and then *random* (with average of 3.97). It is clear that the more restrictive the movement model, the lower the value of ***L***. The overall average number of added links ***L*** is 2.51 links. For most practical purposes, we believe that *cluster* and *neighbor* models are more suitable -than *random*- for representing mobile movement. For *cluster* and *neighbor* movements, the average is 1.78 links. Also, of practical relevance are the Mbone topologies simulated with *cluster* and *neighbor* movements. These scenarios have an average of 2.5 links.

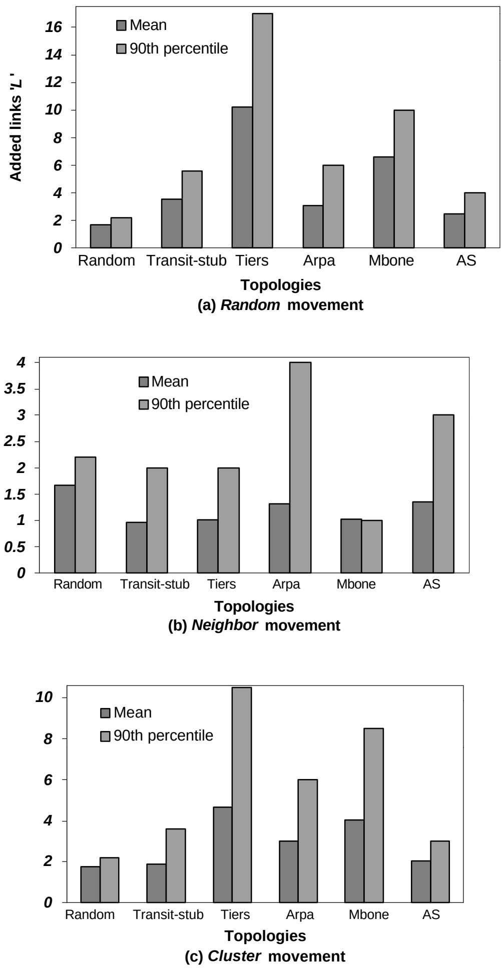


Figure 7. Added links '***L***' (average and 90th percentile) for different topologies and movement patterns.

6.6. *B/L* Ratio

As noted above, in our proposed protocol, handoff delay is a function of round trip time (***rtt***) to establish the new branch of the tree, which is, in turn, a function of ***L***[16]. For basic MIP, handoff delay is a function of the delay to register the new care-of-address by the MN at the HA, and have the packets flow to the new location from the HA (i.e., handoff delay is function of ***rtt*** from the MN to the HA). In turn, this is a function of ***B***, as shown in Figure 4. We define the ratio '***B/L***' as one measure to compare handoff latency between our protocol and MIPv4. ***B/L*** ratio measures for our simulations are given in Figure 8. The ratio ranges from 1.04 (for random topologies with cluster movement) to 4.32 (for tiers topologies with neighbor movement), with an average of '2.31' for all topologies and movements[17]. Hence, on average, the handoff latency in basic MIP is more than twice as much that for our protocol.

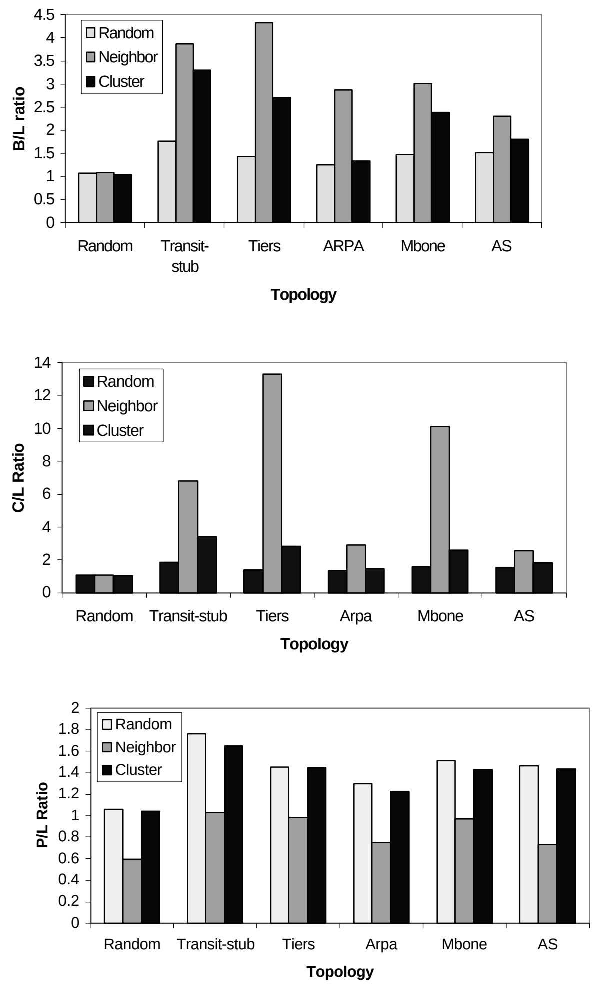


Figure 8. Average *B/L, C/L and P/L* ratios for the different simulated topologies and movement models

6.7. *C/L* Ratio

In MIPv6, the MN sends binding update to the CN during handoff. Hence, handoff latency is a function of the ***rtt*** from MN

[16] Handoff delay is also a function of the delay experienced over the wireless link and the detailed protocol mechanisms that are used, among other things. The detailed simulation is subject to future work. In this study, however, we focus on delays due to the general architecture (not mechanistic details), of which we believe ***L, B, C and P*** to be major components.

[17] This ratio '***B/L***' goes up to 2.7 if we exclude the flat random topologies, which are the least practical of the topologies used in our study.

to CN, which is, in turn, a function of ***C*** as shown in Figure 4. We use the ratio '***C/L***' to compare handoff performance between MIPv6 and our protocol. Results are shown in Figure 8. ***C/L*** ratio ranges from 1.05 (for random topologies with cluster movement) to 13.3 (for tiers topologies with neighbor movement), with an average of '3.38' for all topologies and movements. Hence, on average, the handoff latency in MIPv6 is more than three times that for our multicast-based protocol.

6.8. *P/L* Ratio

In the 'previous location' approach, binding updates are sent to the previous location to reduce handoff latency in MIPv6 (until a binding update is eventually sent to the CN, such that packets follow the shortest path, say). Handoff latency is a function of ***P*** as described above. Note that ***P*** is always '1' in the neighbor movement. Similar to the above ratios, we define the ratio '***P/L***'. Results are given in Figure 8. ***P/L*** ranges from 0.6 (for random topologies with neighbor movement) to 1.77 (for ts topologies with random movement), with overall average of '1.28'. It is interesting to note that only during neighbor movement does this ratio go below '1' (i.e., previous location approach gets lower handoff delay than our protocol). It is also interesting to investigate how ***P*** and ***L*** behave with individual topologies and movement models. This is shown in Figure 9, where ***P*** and ***L*** seem to have similar sensitivity to topology type and movement model, with ***L*** being less than ***P*** in most of the cases studied.

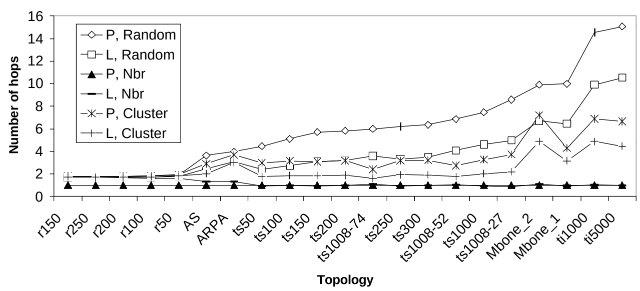


Figure 9. *P* has same sensitivity to topology as *L* (they follow each other closely with $P > L$ in most cases)

Handoff delay results are summarized in Table 2 ('w/o r' is the average if we exclude random topologies).

B/L ratio			
min	max	**avg**	**w/o r**
1.04	4.32	**2.31**	**2.7**
C/L ratio			
min	max	**avg**	**w/o r**
1.05	13.3	**3.38**	**4.11**
P/L ratio			
min	max	**avg**	**w/o r**
0.6	1.77	**1.28**	**1.4**

Table 2 Summary of handoff latency results

We should note here that we expect similar multicast-based approaches (such as Daedalus project and MSM-IP) to behave similarly with respect to handoff delay. For the Daedalus project, however, routing efficiency is similar to that of triangle routing of MIP shown earlier. As for MSM-IP, there are some architectural differences, such as the start-up phase, and mechanistic details.

7. SCALABILITY AND ROBUSTNESS

In this study, we have analyzed several performance aspects of our proposed architecture in a wide-area network context. There are several other essential aspects to be considered. Scalability is definitely a major concern. In this paper, we have shown that our architecture significantly reduces consumed network bandwidth resources, but we have not discussed multicast state overhead. In general, if there are '*x*' MNs, each communicating with '*y*' CNs on average, then routers in the network should create '*x.y*' *(S,G)* states. A state, however, is created only en-route from the corresponding CN to MN.

In terms of other scalability and robustness aspects, we retain properties of the underlying multicast routing infrastructure [8,9].

8. CONCLUSION AND FUTURE WORK

In this paper, we present a protocol for supporting IP mobility. Our architecture is multicast-based, in which a mobile node is assigned a multicast address, and the correspondent nodes send packets to that multicast group. As the mobile node moves to a new location, it joins the multicast group through the new location and prunes through the old location. Dynamics of the multicast tree provide for smooth handoff, efficient routing, and conservation of network bandwidth. We present extensive simulations of our protocol in the context of wide-area networks. We compare our protocol to basic Mobile IP, MIPv6, and an enhanced version thereof. Our results show that Mobile IP consumes, on average, almost twice as much network bandwidth, and experiences more than double the end-to-end delays and handoff latency, than does our protocol. Our protocol incurred the least average handoff delay. Our simulations used three movement models; *random, neighbor* and *cluster* movements, and 21 topologies of various sizes and degrees.

Our study here was based on route analysis. We plan to conduct packet-based simulations to measure packet loss, jitter, delay and throughput during handoff. We also plan to investigate interaction with higher layer TCP and multicast-based applications.

ACKNOWLEDGEMENT

This work is part of a collaboration project with Nortel Networks.

REFERENCES

[1] J. Mysore, V. Bharghavan, "A New Multicasting-based Architecture for Internet Host Mobility", *Proceedings of ACM MobiCom*, September 1997.

[2] S. Seshan, H. Balakrishnan, R. Katz, "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience", *Kluwer Journal on Wireless Networks*, 1995.

[3] C. Perkins, "IP Mobility Support", *RFC 2002*, Oct 96.

[4] C. Perkins and D. Johnson, "Mobility Support in IPv6", *Proceedings of MobiCom'96*, November 1996.

[5] C. Perkins, D. Johnson, "Route Optimization in Mobile IP", *Internet Draft, IETF,* February 2000.

[6] R. Caceres, V. Padmanabhan, "Fast and Scalable Handoffs for Wireless Internetworks" *MobiCom* '96.

[7] A. Snoeren, H. Balakrishnan, "An End-to-End Approach to Host Mobility", *ACM MobiCom*, Aug 00.

[8] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, V. Jacobson, M. Handley, C. Liu, P. Sharma, "Protocol

Independent Multicast – Sparse Mode: Protocol Specification", *IETF RFC 2362* March '98.
[9] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, M. Handley, "The MASC/BGMP Architecture for Inter-domain Multicast Routing", *Proceedings of ACM SIGCOMM*, August 1998.
[10] B. Cain, S. Deering, I. Kouvelas, A. Thyagarajan, "Internet Group Management Protocol, version3", *Internet draft of the IETF*, March 2000.
[11] K. Calvert, M. Doar, E. Zegura, "Modeling Internet Topology", *IEEE Comm*, p. 160-163, June 1997.
[12] G. Phillips, S. Shenker, H. Tangmunarunkit, "Scaling of Multicast Trees: Comments on the Chuang-Sirbu scaling law", *ACM SIGCOMM*, August 1999.
[13] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, D. Estrin, "On characterizing network topologies and analyzing their impact on protocol design", *USC-CS-TR-00-731*, March 2000.
[14] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, "Advances in Network Simulation", *IEEE Computer*, vol. 33, No. 5, p. 59-67, May 2000.

BIOGRAPHY

Dr. Ahmed Helmy is an assistant professor of computer networks at the Electrical Engineering department at the University of Southern California. Dr. Helmy received his Ph.D. (1999) in Computer Science (Computer Networks), his M.S. in Electrical Engineering (Computer Networks) (1995), both from the University of Southern California, his M.S. in Mathematical Engineering (1994) and his B.S. in Electronics and Communications Engineering (1992) both from Cairo University faculty of Engineering. His current research interests are in the areas of systematic design and testing of multicast protocols, multicast routing protocols, network mobile architecture, and power-aware protocol design.

Dr. Helmy is currently leading the STRESS project to develop scenario generation algorithms for multicast protocols. He has worked on the VINT and PIM projects at USC and ISI. He is in charge of the 'Multicast-based Architecture for IP Mobility' project to develop efficient mobility support protocols for the next generation Internet, in collaboration with Nortel. He is also the director of the computer networks design and testing laboratory at USC. email: helmy@ceng.usc.edu, webpage: http://ceng.usc.edu/~helmy.