IP Paging Service for Mobile Hosts

R. Ramjee
Bell Labs
Lucent Technologies
ramjee@bell-labs.com

L. Li
Dept. of Computer Science
Cornell University
lili@cs.cornell.edu

T. La Porta and S. Kasera
Bell Labs
Lucent Technologies
tlp,kasera@bell-labs.com

ABSTRACT

In wireless networks, mobile hosts must update the network with their current location in order to get packets delivered. Paging facilitates efficient power management at the mobile host by allowing the host to update the network less frequently at the cost of providing the network with only approximate location information. The network determines the exact location of a mobile host through paging before delivering packets destined to the mobile host. In this paper, we propose the concept of paging as an IP service. IP paging enables a common infrastructure and protocol to support the different wireless interfaces such as CDMA, GPRS, wireless LAN, avoiding the duplication of several application layer paging implementations and the inter-operability issues that exists today. We present the design, implementation, and detailed qualitative and quantitative evaluation, using measurements and simulation, of three IP-based paging protocols for mobile hosts.

1. INTRODUCTION

In networks that support mobility, the precise location of a mobile host must be known before it can be delivered data. There is a tradeoff between how closely the network tracks the current location of a mobile host, versus the time and complexity required to locate a mobile host whose position is not precisely known.

Tracking the location of a mobile host is performed through *update* procedures in which a mobile host informs the network of its location at times triggered by movement, timer expiration, etc. Locating a mobile host is performed through search procedures which include *paging* the mobile host. Paging typically includes transmitting a request for a mobile host to a set of locations, in one of which the mobile host is expected to be. This set of locations is called a paging area and consists of a set of neighboring base stations.

A network that supports paging allows the mobile hosts to operate in two distinct states - an *active state* in which the mobile host is tracked at the finest granularity possible such as its current base station (resulting in no need for paging), and a *standby state* in which the host is tracked at a much coarser granularity such as a paging area. The network uses paging to locate the mobile host in standby state. The mobile host updates the network less frequently in standby state (every paging area change) than in active state (every base station change). Since the power spent in updating the network is an order of magnitude greater than the power spent in standby mode [20], battery power consumption at the mobile host is reduced significantly [19].

Thus, the main benefit of providing a paging service is to facilitate efficient power management at the mobile host. The cost of paging is the complexity of the algorithms and protocols required to implement the procedures, and the delay incurred in locating a mobile host before data can be delivered.

Many research and standardization efforts are underway to integrate both indoor (LAN) and outdoor (WAN) wireless access technologies over a common IP-based access network [21] [22] [23]. This will allow more flexibility in deploying equipment which may greatly reduce network operations costs [24]. In addition, an IP-based access network will be able to support both voice and data services on a common infrastructure resulting in seamless support of services across wired and wireless networks. These IP-based networks are expected to be the basis for mature third and fourth generation wireless networks.

Paging service is available in wireless wide-area networks (WAN) such as General Packet Radio Service (GPRS) [3] and CDMA data [2]. Wireless local area network (LAN) protocols such as IEEE 802.11 [4], also have the notion of a power-save state [20]. Here the paging functionality is limited to waking the mobile host from power-save (standby) state to active state at a single base station. The paging architecture and protocols in each of these networks are defined independently and do not inter-operate. This precludes seamless movement between local-area and wide-area networks or between wide-area networks of different types. An IP-based paging architecture is a key element that would facilitate inter-operability between these different wireless networks.

Mobile IP [11], the mobility protocol for IP networks, currently does not support paging. The focus of this paper is on architectures and protocols for adding paging in IP networks that support mobility. Together, IP paging and IP mobility will support the full functionality present in current wide-area wireless networks, thereby serving as the basis for efficient and cost-effective all-IP third and fourth generation wireless networks.

An effective paging service should have the following characteristics. First, it should be simple to implement. Second, it should be flexible in terms of supporting various update and paging algorithms, and in terms of being compatible with existing IP mobility protocols. Third, it should be scalable to large networks with large number of hosts. Fourth, it should be efficient in terms of the amount of resources it requires to execute (memory, processing power) and overhead it incurs (number of messages, number of updates). Fifth, the paging service must be reliable, because its failure may render operational mobile hosts to become unreachable. Finally, ease of deployment of the paging service is also important.

In this paper, we make three important contributions. First, we propose paging as a novel IP layer service in order to facilitate the deployment of a common, IP-based infrastructure for different wireless networks such as GPRS, CDMA, wireless LAN etc.

Second, we present the design of three paging architectures and protocols. The first two, called Home Agent (HA) paging, Foreign Agent (FA) paging, are based on the architecture defined for Mobile IP [11] and its derivatives [12]. These protocols, like the systems on which they are based, manage paging in a centralized and quasi-centralized manner respectively. The third, called Domain paging, is a distributed approach that is still compatible with existing IP mobility protocols. While these paging protocols determine which network nodes initiate and process update and paging messages, algorithms are necessary to determine under what condition updates occur and in what locations a mobile host is paged. We implement three well-known paging algorithms that demonstrate the flexibility of our IP paging architectures.

Third, we present a thorough characterization of the performance of the different paging protocols and paging algorithms in terms of the six goals we outlined above. We use the measurements obtained from our implementation and real mobility traces as input to a simulator to study the behavior of IP paging under high load. We also use simple analysis to characterize reliability.

The remaining sections are organized as follows. In Section 2, we describe three IP paging architecture and protocols. In Section 3, we describe three well-known paging algorithms that we implemented using the Domain paging protocol. In Section 4, we present the software architecture of our implementation and highlight some issues in implementing the Domain paging protocol in the FreeBSD operating system. In Section 5, we characterize the performance of the different paging protocols and algorithms using real mobility traces and performance numbers from implementation. In Section 6, we discuss reliability aspects of the paging protocols. In Section 7, we discuss related work and other relevant issues. In Section 8, we present our conclusions.

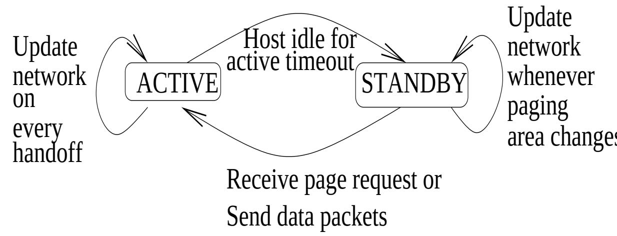


Figure 1: Mobile host state diagram

2. IP PAGING ARCHITECTURE AND PROTOCOLS

In systems that use paging, mobile hosts will be in one of two states as shown in Figure 1. When actively transmitting or receiving data, the mobile host is in active state. In this state the network knows the precise location of the mobile host and can therefore deliver data quickly. If the mobile host is inactive for a period of time, it will transition into an idle, or standby state, during which time its location is not precisely known[1]. During this period, if data arrives in the network for the mobile host, the host must first be located before data can be delivered. This procedure of locating the mobile is broadly referred to as paging.

Recall that, when in active state, the mobile host updates the network each time it changes its point of network attachment (we assume each point of attachment is a separate base station with an IP address). When in standby state, the mobile host updates the network less frequently, i.e. only when it changes its paging area. If a mobile host spends sufficient time in standby state, power saving is achieved through reduced transmissions of updates. Thus, power saving is a benefit in a network that supports paging service.

A network that supports paging service consists of several components such as architecture, protocols, and algorithms. A *paging architecture* determines where paging state is stored. A *paging protocol* determines which node initiates paging, defines the messages exchanged between various nodes in the architecture, and is responsible for updating and maintaining paging state. A *paging algorithm* determines how and where a mobile host is searched - for example, searching entire paging areas in parallel by broadcasting paging messages, or searching paging areas in smaller increments sequentially.

The focus of this paper is on novel IP paging architectures, protocols, and algorithms that are necessary for providing the paging service. IP paging architecture and protocols determine the scalability, reliability, deployability, and efficiency of the paging service but do not affect the amount of power saving directly - power saving is a benefit in a network that supports paging and is directly impacted by factors such as the usage, mobility pattern, and update policies of the mobile host.

Note that networks that support paging also have well-defined wireless link-layer paging support. The link-layer support allows for a mobile host to listen to broadcast page messages

[1]Note that exact synchronization is not required.

periodically in a very energy efficient manner. Thus, we assume that the IP paging protocols defined in this paper leverage the efficient link-layer mechanism when the page messages are transmitted over the wireless medium and use IP paging protocol messages only over wired links.

In the following sections, we describe the three paging architecture and protocols in detail, and discuss how they meet our goals qualitatively. We quantify their characteristics in subsequent sections.

Since IP paging is intricately connected with IP mobility support, we present our paging protocols in the context of Mobile IP. Mobile IP defines two entities to provide mobility support: a *home agent* and a *foreign agent*. Foreign agents are identified by a care-of address. Packets sent to the mobile host are intercepted by the home agent which then tunnels the data packets to the mobile host through the foreign agent using the care-of address. The foreign agent may be present at the base station or run locally at the mobile host, in which case the care-of address is known as a co-located care-of address.

2.1 Home Agent Paging

Home Agent paging is performed in a centralized manner at the Mobile IP home agent. This is illustrated in Figure 2(a). When a mobile host registers with its home agent, it includes the identity of its current paging area. When a packet destined for a mobile host in standby state arrives at the home agent, the home agent buffers the packet and contacts all the base stations in the paging area (message 1). The base stations subsequently page over the air (message 2). The mobile host then registers its current location with the home agent (messages 3 and 4), which delivers the buffered as well as subsequent packets. It is possible to use a multicast address to identify a paging area. This requires the use of a globally visible multicast address which may not be scalable. Otherwise, the home agent must have access to a database containing paging area information. A discussion of the use of multicast for paging can be found in Section 7.

We characterize this protocol as centralized because the paging initiator (home agent) is a statically determined, centralized entity. This results in a *simple* protocol implementation that is required only at the home agent and mobile host. It is *flexible* in that different update and paging algorithms may be implemented in the home agent and mobile hosts.

However, this approach has some drawbacks. Since home agents may be located some distance from the mobile hosts, search costs may be high resulting in reduced *efficiency*. Also, since home agents in wide-area wireless networks are expected to serve thousands of mobile users, increasing the *scalability* of the centralized home agent could result in high costs. For the same reason, the failure of a home agent or paging process in a home agent would render all the mobile hosts it serves as unreachable, decreasing *reliability*. However, this is a characteristic of home agents for data delivery in basic mobile IP as well.

In terms of *deployment*, the home agent requires the addresses of the base stations in the paging area; since the base stations and the home agent can belong to different

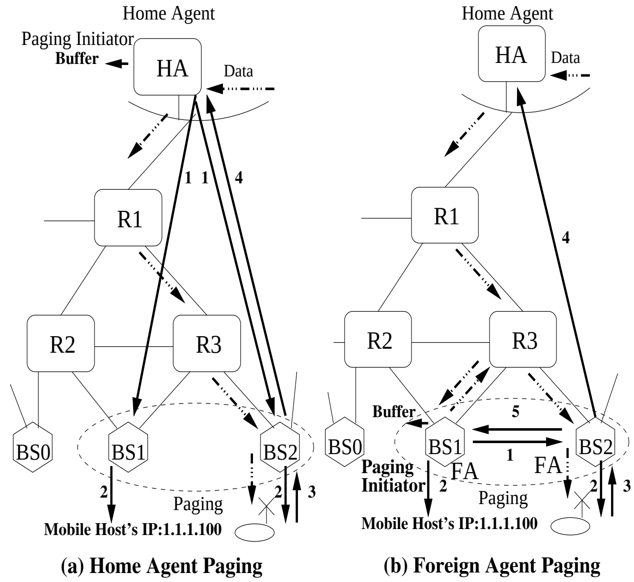


Figure 2: Home Agent and Foreign Agent paging

administrative domains, the paging information could be considered confidential and may not be available. Also, for a home agent to support global roaming of its mobile hosts, it would have to access databases with network configuration information for every access network, which is problematic.

The Home Agent paging approach is similar to the centralized paging implementations in current wide-area cellular networks; in GPRS, paging is performed at the Serving GPRS Service Node (SGSN) [3], while in CDMA, it is implemented at the Mobile Switching Center (MSC) [2]. The main difference is that cellular paging is always performed inside the visiting network rather than from the home network as in Home Agent paging, thus avoiding the confidentiality and scalability issues of Home Agent paging.

2.2 Foreign Agent Paging

Foreign Agent paging addresses many of the issues with Home Agent paging. In Foreign Agent paging, paging is initiated from the mobile host's last attached foreign agent (base station). As shown in Figure 2(b), when a packet destined for a mobile host in standby state arrives at the home agent, the home agent tunnels the packet to the foreign agent as in basic Mobile IP. Thus, the home agent is unaware of paging or standby states of the mobile host. The foreign agent buffers the packet and contacts the base stations (foreign agents) in the paging area (message 1); the base stations subsequently page over the air (message 2). The mobile host then registers its new location (foreign agent) with the home agent (messages 3 and 4), and also simultaneously informs the previous foreign agent (message 5) so that the buffered packet can be forwarded. If the mobile host happened to remain at its previous base station, messages 4 and 5 are avoided.

We characterize Foreign Agent paging as quasi-centralized because while paging for a given mobile host is distributed among the different foreign agents in the network, the paging initiator is statically determined and fixed (previous foreign agent) at any given time. This approach is fairly *simple* to implement, requiring changes to foreign agents and mobile hosts. It also provides *scalability* since the processing load of paging mobile hosts is distributed among the different foreign agents in the network. Furthermore, the confidentiality issues of Home Agent paging are avoided since paging is localized to one administrative domain, making the system *efficient* and *deployable*. The system is somewhat *flexible* in that different update and paging algorithms can be implemented in the mobile hosts and foreign agents, and it is compatible with a basic Mobile IP network. However, this approach will not work with the mobile host CCOA option of Mobile IP.

In Foreign Agent Paging, the *reliability* concerns are more serious than the Home Agent paging approach. In Home Agent paging, home agent failure would leave the mobile host disconnected. In Foreign Agent Paging, in addition, even in the presence of an end-to-end path to the mobile host, the failure of the previous foreign agent could result in the mobile host being unreachable indefinitely (since the previous foreign agent is the paging initiator).

Foreign agent paging is similar to paging in cellular data networks such as GPRS or CDMA in that paging is initiated in the current serving area of a mobile host. The main difference is that the cellular networks rely on link level protocols to disseminate paging information to the base stations, while in Foreign Agent paging, the foreign agents are on base stations, so IP is used to distribute the paging messages.

2.3 Domain Paging

In order to comprehensively address the shortcomings of Home Agent and Foreign Agent paging, we propose a router assisted paging scheme called Domain paging. In Domain paging, paging state is distributed among the routers and base stations in a domain rather than at one fixed node such as the foreign agent or the home agent. This is similar to the HAWAII micro-mobility protocol [13] where routing state is distributed among the routers and base stations in the domain. In this paper we define a domain to be an autonomous system in the Internet (like a stub domain in the transit stub domain model of the Internet [1][2]). The gateway into each domain is called the *domain root router*. The routers/base station in the path from the domain root router to a given mobile host maintain routing and paging information for that mobile host, while other routers in the domain have no specific knowledge of the same mobile host.

In Domain paging, when a mobile host is in active state, it sends an update message to the domain root router whenever it moves out of its current base station. When the mobile host is in standby state, it sends an update message to the domain root router only when it moves out of its paging area. These messages are propagated hop-by-hop along the path from the mobile host's current base station to the domain root router, thereby creating new routing and paging state on each router/base station in the path[3]. The

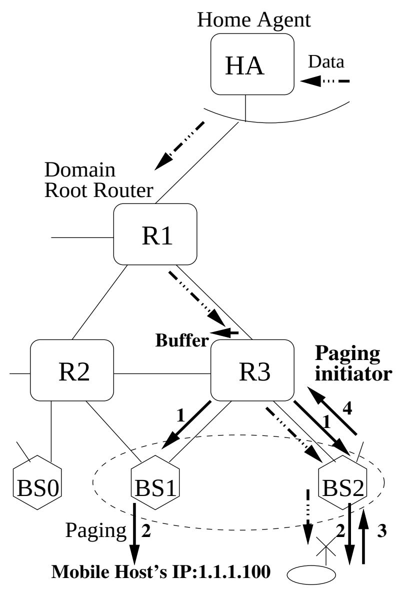


Figure 3: Paging from routers in Domain paging

identity of the domain root router is periodically advertised in all the paging areas of the domain[4].

Figure 3 illustrates Domain paging using a simple example. Here, the foreign agent is co-located with the mobile host (CCOA option of Mobile IP). The paging area is identified using an administratively-scoped IP multicast group address (239.0.0.1) [9]. When a packet is received for the mobile host that is in standby state, a router or base station along the path from the domain router to the last (recorded) base station of the mobile host dynamically selects itself to be the page initiator (router R3 in Figure 3). The page initiator then buffers the packet and sends out a page request for the mobile host in the paging area (message 1). On receiving a page request (message 2) the mobile host sends a page response to the initiating router (message 3). This page response also propagates hop-by-hop thereby updating the routing and paging information along the path (message 4) . On receiving the page response, the page initiator transmits the packet which is now routed to the mobile host with the help of the state created due to the page response. Any future packets meant for the mobile are also routed in the same way. We note that paging functionality can be implemented in a router without affecting its fast path forwarding prowess through the use of virtual interfaces (as described in the next section). The detailed description of Domain paging procedures is presented in Appendix A.

The decision as to whether to initiate paging from a router or base station is configurable; it is dependent on how much of the paging load needs to be shared by each router or base station. One possible measure of paging load, that we use in this paper, is the number of outstanding paging requests.

---

[2]In the transit stub domain model there might be a few stub-to-stub edges but these occurrences in the Internet is rare and will be ignored in this paper.

[3]In Home/Foreign Agent paging, such an update would result in a new tunnel establishment between the home agent and the foreign agent.

[4]If routers in the domain do not support Domain paging, this reduces to Foreign Agent paging.

The distribution of paging functionality among the routers and base stations in the domain serves the dual purpose of load balancing for performance and fail-over for reliability. When base stations or routers fail, paging can be initiated from another node, using a simple soft-state refresh mechanism.

One can view Domain paging as a generalization of Foreign Agent paging where a base station or a router is dynamically selected as the paging initiator, thus resulting in a completely distributed approach. This avoids the *reliability* issues with Foreign Agent paging (also examined quantitatively in Section 6) while retaining the *scalability* and *efficiency* benefits of a distributed approach. Domain paging is *flexible* and allows multiple paging algorithms to be implemented. Also, since Domain paging is localized to one administrative domain, confidentiality issues of Home Agent paging are avoided. Furthermore, unlike Foreign Agent paging, which could generate updates to home agent (message 4 in Figure 2(b)), *Domain paging is truly localized and results in no updates to the home agent due to paging.* Unlike Home Agent and Foreign Agent paging, this approach works both with the co-located care-of option of Mobile IP (shown in the example) as well as the network-based foreign agent option.

The benefits of Domain paging do not come for free. Domain paging requires additional functionality in the routers of the wireless network domains for buffering packets at page initiators and also for processing and maintaining paging state at routers. Therefore, Domain Paging is not as *simple* to implement nor as easy to *deploy* as the other protocols. One way to ease deployment is to allow for incremental updates of selective routers in the domain with paging functionality and augmenting the protocol with tunneling as proposed in [6] but this will result in reduced performance.

3. PAGING ALGORITHMS

Paging algorithms determine how and where (which base stations) a mobile host is searched. In this section, we describe three well-known paging algorithms [18]. In the descriptions of the three paging protocols so far, we assumed that all the base stations in the paging area are paged. We refer to this paging algorithm as the Fixed algorithm. More sophisticated paging algorithms, which search the paging area hierarchically or exploit locality of user movement, termed Hierarchical and Last-location paging, have also been proposed and studied in the literature [18]. Since paging algorithms are intricately connected with paging protocols, we would like to study the impact of different paging algorithms on the three paging protocols. We describe these well-known paging algorithms briefly.

- Fixed paging: In Fixed paging, the base stations that comprise a paging area are fixed by a network administrator. Thus, the network knows the current paging area of the mobile host and broadcasts a page message to all the base stations in the paging area.
- Hierarchical paging: This is a generalization of the Fixed paging. The paging area is divided into hierarchies by the network administrator. The network pages the base stations at the first level of hierarchy. If there is no response within a timeout interval, the network pages the next level of hierarchy and this process is repeated until the entire paging area is searched.
- Last-location paging: In Last-location paging, the network first pages only the mobile host's last known base station. If there is no response to the page within a timeout interval, the network then pages all other base stations in the paging area. In the case of networks with low mobility users, probing the user's last known location can reduce paging load significantly.

4. IMPLEMENTATION

In this section we describe our implementation of Domain paging. The two main goals for this implementation are to

- show how paging functionality can be implemented in a router (though PC–based) without affecting its fast path forwarding prowess, and,
- measure processing times associated with different paging operations in a real system.

We choose to implement Domain paging since it has the main functionality of both Home Agent paging (when a router initiates paging) and Foreign Agent paging (when a base station initiates paging), thus serving as a superset of all three paging protocols. We implement all three paging algorithms (described in Section 3) in conjunction with the Domain paging protocol.

Our implementation platform is a PC–based router running FreeBSD 3.1. Equivalent functionality, such as virtual ports, is also available on most routers and our implementation could be adapted for them. Figure 4 illustrates the different components of our protocol software and its interaction with the kernel. The paging protocol is implemented in user space and send/receives protocol messages on a well-known UDP port. This allows for ease of testing and deployment.

Recall that mobile hosts may be in one of two states, active and standby, and paging is initiated when the host is in standby state only. Since the paging daemon is implemented in user space, it is most efficient if only packets destined for mobile hosts in standby state are sent to the user level paging daemon while packets destined for mobile host in active state are forwarded in the kernel itself. We achieve this through the use of a tunnel interface in FreeBSD. The tunnel interface is a pseudo network interface that delivers packets to user level processes. The paging daemon simply maintains the kernel routing entries through the routing socket interface such that routing entries for active mobile hosts point to real outgoing interfaces; when the mobile hosts are idle and change to standby state, the paging daemon changes the respective routing entries to point to the tunnel interface. In this way, the paging protocol selectively receives only those packets for which paging is necessary while the kernel forwards all incoming packets as usual in the fast path.

One issue with initiating paging is the need for the daemon to know on which interface the packet arrived (Line 2 in Figure 10 in appendix A). Since the FreeBSD tunnel interface

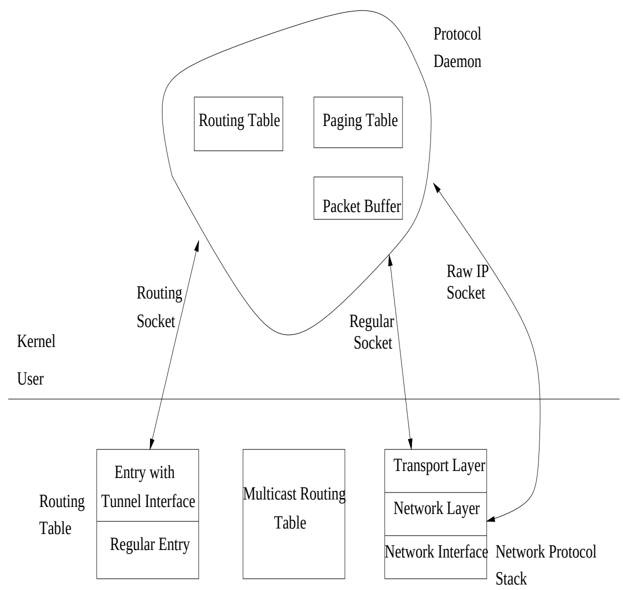


Figure 4: Protocol Software Architecture

does not provide the interface information with the data packet, we added a new ioctl option to pass the interface information from the kernel to the paging protocol daemon in the user address space with each data packet. This new ioctl option requires adding about 10 lines of code in the kernel.

Finally, when a page response is received, the paging initiator updates the kernel routing entry for the mobile host and uses the raw IP socket to deliver the buffered IP packet(s) to the mobile host. Subsequent IP packets for the mobile host are forwarded directly by the kernel.

We built a testbed consisting of one PC serving as a router, two PCs serving as base stations, and a mobile host. The router is connected to the base stations through a 100Mb/s Ethernet and the mobile host is connected to the base stations through a 2Mb/s WaveLAN. We present the CPU processing time for different aspects of the paging protocol processing. All these measurements were obtained on 333MHz PCs running the FreeBSD 3.1 operating system.

Each experiment involves sending a series of ICMP echo request (ping) packets, with a period of one second, to a given mobile host. When this packet arrives at the testbed domain, the ICMP packet is buffered and a page request is initiated by the Domain paging protocol. When the mobile host responds to the page request from one of the base stations, the buffered ICMP echo request packet is forwarded to the mobile host, which then replies with an ICMP echo reply packet. Thus, the latency for the first ping packet includes the cost of paging. Subsequent ping packets are routed along the fast path, resulting in no paging overhead. This experiment is repeated over 100 times to obtain various paging processing timings in the presence of paging.

Now, in Domain paging, paging can be initiated either at a router or at the base station. The processing needs of router initiated Domain paging and base station initiated Domain paging are comparable to the processing needs of Home Agent paging and Foreign Agent paging respectively. In Table 1, we present results for both possibilities of page initiation for all three paging algorithms (Fixed, Last-location, and Hierarchical). Table 1 also represents the three algorithms for the two cases of whether the user was located at the previous base station (found) or not (not).

Paging protocol processing can be classified into four main functions: *init_page_request* (receive the IP data packet from the tunnel interface, buffer the packet, and initiate a page request to appropriate base station(s)), *retry_page_request* (increment page request sequence number and send page request to different base stations) *recv_init_page_request* (receive the page request and initiate a page request over the air interface), *recv_page_response* (receive the page response, update the paging/routing entry in the kernel, and, if necessary, forward the response to the initiator). Note that there are minor differences to the functionality depending on whether the function is implemented at a base station or in a router.

We make a couple of interesting observations regarding the processing numbers. Consider the case when a router is initiating the page and the user is found at the previous location (highlighted numbers in columns 2, 4, and 6, in Table 1). The *init_page_request* processing at the router in the Fixed and Hierarchical algorithms is $0.173ms$ and $0.196ms$ respectively but the same item is $0.323ms$ for Last-location algorithm. The reason is that the Last-location scheme needs to perform a route lookup in the kernel in order to send a unicast paging request to the right base station while, in the Fixed and Hierarchical case, we just send a page message to the appropriate multicast group. Another observation is that, in general, *recv_page_response* is more expensive than other processing ($0.2 - 0.4ms$ vs less than $0.2ms$ for init processing) because only *recv_page_response* processing involves updating the kernel fast path routing entries.

In the next section we use the measured paging processing times from Table 1 in order to compare the paging latencies of different paging protocols in a simulator that simulates a highly loaded network with millions of users and up to hundred nodes. While the above measurements have assumed that processing is devoted solely to paging, paging processing will have to share CPU resources with other protocols and administrative tasks in a router. Thus, while the qualitative results of the next section will still hold true in a deployed system, processing paging protocol messages in real networks. could become a bottleneck at much lower loads than the loads we simulated.

5. PERFORMANCE RESULTS

We simulate the three paging protocols for different paging algorithms in order to characterize paging latency and updates. We use the paging processing measurements from Table 1 and mobility and call traces available from [16] to drive our simulator. These mobility traces have been shown to provide a realistic framework for modeling connection oriented traffic in Personal Communication Networks and have been corroborated using measurements and surveys of ac-

Router initiated (HA)	Fixed (found)	Fixed (not)	Last-loc. (found)	Last-loc. (not)	Hier. (found)	Hier. (not)
init_page_request (router)	**0.173**	0.173	**0.323**	0.316	**0.196**	0.203
retry_page_request(router)	-	-	-	0.157	-	0.155
recv_init_page_request(bs)	0.080	0.080	0.082	0.068	0.079	0.066
recv_page_response(bs)	0.378	0.378	0.331	0.317	0.334	0.316
recv_page_response(router)	0.279	0.279	0.190	0.193	0.204	0.215
Base station initiated (FA)						
init_page_request (bs)	**0.197**	0.199	**0.183**	0.189	**0.197**	0.213
retry_page_request(bs)	-	-	-	0.117	-	0.118
recv_init_page_request(bs)	0.106	0.113	-	0.114	0.106	0.114
recv_page_response(bs)	0.237	0.233	0.251	0.234	0.249	0.232
recv_page_response(router)	-	0.429	-	0.413	-	0.428

Table 1: Paging processing times in milliseconds

tual human activities [8]. The simulation results presented in this paper use two traces of one hour duration, one representing wide-area traffic and other representing local-area traffic, called Bay Area Location Information (BALI) and Stanford University Local Area (SULA) respectively. In addition to call traces, we add trace information representing paging due to *messaging traffic* in order to simulate higher paging load representative of future applications. The inter-arrival time between two messages to a given user is assumed to be exponentially distributed as in [18].

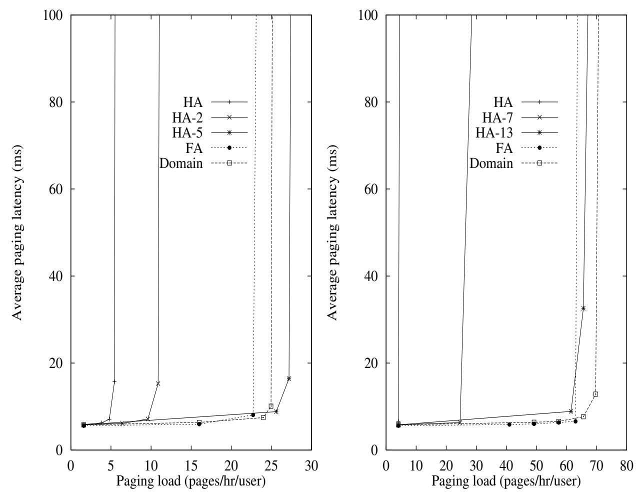
The simulation network topology of BALI is the San Francisco Bay area with 90 zones. These zones can be mapped logically to the base stations in our earlier discussion since IP-level handoffs in currently deployed networks will occur initially only across these zones rather than base stations. The simulation network topology of SULA is 36 zones in a six-by-six wrapped mesh. The total number of users is about 1.8 to 2 million in each of the traces, thus making SULA a more dense network than BALI. The trace file contains call and mobility details. We map each call trace to a paging event for the callee. The paging load due to such calls is up to 2 pages/hr/user. In addition, the paging load due to simulated messaging traffic is up to 60 pages/hr/user.

In order to simulate IP paging in this network, we divide the zones into paging areas. In our simulated network topology, each set of six neighboring zones form a paging area. The domain has a router hierarchy of two with one mid-level router for each paging area and one domain root router.

The two metrics that we are interested in are *average paging latency* for delivering a packet to the paged host and the *number of updates* to the home agent due to paging and movement of mobile hosts. It is essential to keep the average paging latency low since this directly contributes to the delay in contacting the user. It is also important to the keep the number of updates to the home agent low so that cross-network traffic and home agent processing overhead are reduced.

5.1 Paging Latency

In Figure 5, we plot the average paging latencies for different paging loads in SULA and BALI traces for Home Agent, Foreign Agent, and Domain paging protocols. Paging loads throughout the rest of this section is in number of



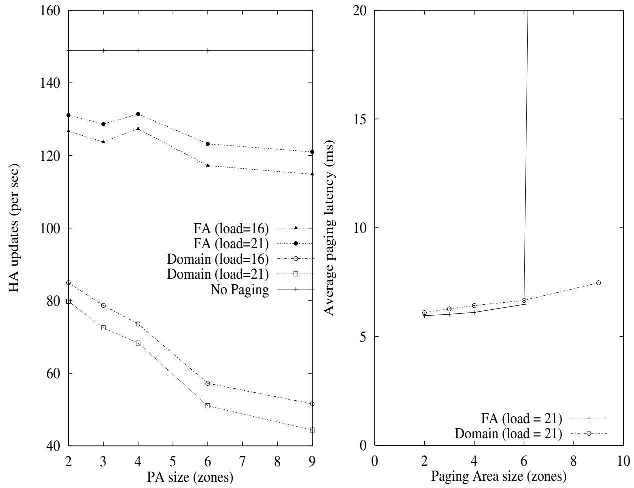
(a) SULA (average) (b) BALI (average)

Figure 5: Paging latency vs paging load

pages/hr/user. The Fixed paging algorithm is used with a paging area size of six zones. We also make the conservative assumption that the delay to the Home Agent is negligible; otherwise, a factor of 1.5 round trip delay between the Home Agent and the base station must be added to the Home Agent paging latencies.

First consider Home Agent paging. In order for Home Agent paging to scale to reasonable loads, multiple processors are used for a single home agent. In this case, mobile hosts are statically mapped to the different home agent processors uniformly. It is clear from Figure 5(a) that Home Agent paging with just one processor does not scale well with load since the centralized home agent becomes a bottleneck. In order to achieve comparable performance with Domain and Foreign Agent paging, the Home Agent paging needs 5 and 13 processors for the SULA and BALI traces respectively

Now consider Foreign Agent and Domain paging. In this topology with six zones in the paging area, Foreign Agent paging uses six base station processors for processing while Domain paging uses one additional mid-level router as well. By using the router, which gives 16% gain in processing power, Domain paging is able to support about 11% in



(a) SULA (update) (b) BALI (latency)

Figure 6: Impact of paging area size

higher paging load over Foreign Agent paging in this particular configuration at $100ms$ latency. Our experiments (not shown) also indicated that the additional processing power reduces the 99 percentile paging latency more significantly than the average latencies. Thus, *Domain paging is able to support the highest call load among the three protocols by utilizing available processing resources (routers) in the domain efficiently.*

5.2 Impact of varying paging area size

We now consider the impact of paging area size. Having a larger paging area reduces the number of updates to the home agent since the user sends an update in standby state only when the user crosses a paging area. However, a larger paging area implies that the user has to be paged in more zones, resulting in higher processing load and higher paging latency. Thus, there is a trade-off between latency and number of updates that impacts the desired paging area size.

Figure 6(a) shows the number of updates to the home agent versus different paging area sizes for Foreign Agent and Domain paging at different loads (Home Agent paging, not shown, results in a very high number of updates because each page request results in an update). The number of updates when there is no paging is 149/s. When the paging area size is 9, the number of updates in Foreign Agent and Domain paging is reduced by 19% and 72% respectively. In the case of Foreign Agent paging, recall that, apart from movement related updates, updates to the home agent occur in cases when the user is paged and is found at a new foreign agent (message 4 in Figure 2(b)). This results in higher number of updates than Domain paging in which there are only movement related updates and no updates when the user is paged (Figure 3). Thus, *Domain paging has the least number of updates to the home agent.*

Now consider figure 6(b) where paging latency is plotted against different paging area sizes for Foreign Agent and Domain paging. As expected, the paging latency increases with increase in paging area size due to the increased paging processing load. However, since Domain paging uses the routers in the network in addition to the base stations, Domain paging is able to support higher paging area sizes than Foreign Agent paging. In the figures shown, Domain paging is able to support paging area sizes from two to nine while Foreign Agent paging is unable to support a paging area size of nine (the paging processors get overloaded). Finally, a paging area size of six seems optimal for this trace resulting in low latency and lower updates to the home agent.

5.3 Impact of varying paging algorithms

Finally we consider the difference in the three paging algorithms: Fixed, Last-location, and Hierarchical. In Figure 7(a),(b), and (c) we plot the impact of the three paging algorithms on the three paging protocols. The results are for the SULA trace where user mobility is low (high locality).

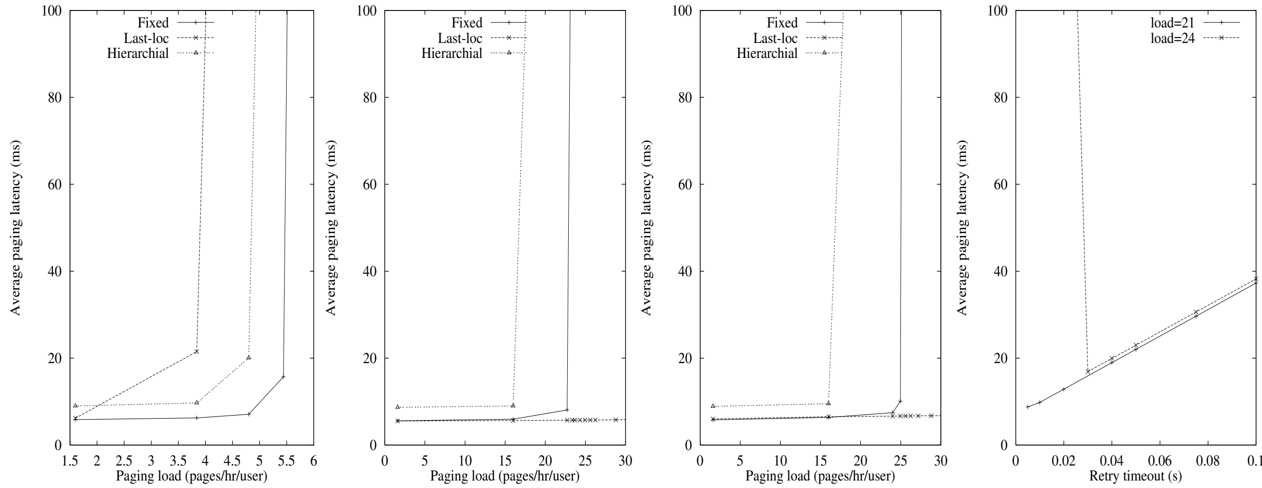
In the case of Home Agent paging in Figure 7(a), we find that the Last-location algorithm performs the worst while the Fixed algorithm performs the best. This seemingly unexpected result even in the presence of high user locality has a reasonable explanation: the Last-location algorithm increases the load at the initiator (see Table 1, row 1) while reducing the load in the base stations. In the case of Home Agent paging, since the paging initiator (home agent) is the bottleneck, Last-location algorithm has the undesirable effect of reducing the paging performance of Home Agent paging. On the other hand, in the case of Foreign Agent and Domain paging, where base station processing is the bottleneck, Last-location performs the best.

Finally, while the Hierarchical algorithm is a generalization of the Fixed algorithm, it does not perform as well as the Fixed algorithm in terms of paging latency for all three protocols because 1) the Hierarchical algorithm could result in multiple paging initiations, the cost of which is much higher than paging processing at the base station and 2) the impact of unnecessary retries, discussed next. The main motivation to using the hierarchical algorithm is to potentially reduce the number of page messages over the wireless link.

In the Last-location and Hierarchical algorithms, a retry timeout interval was used to retry paging at different locations. An unnecessary retry occurs when the paging initiator's retry timeout expires before the host's page response to the initial page request arrives (page response is delayed due to queueing and processing at different nodes). Higher number of unnecessary retries due to low retry timeout value results in increasing the paging load and latency. Hence, in Figure 7(d), as the retry timeout value is increased, the average paging latency decreases up to a certain value (because of reduced number of unnecessary retries). Beyond this value, the average paging latency starts increasing. This is because the cost of locating the host after the retry timeout is now higher due to the larger timeout value. Thus, the Hierarchical algorithm needs a load dependent retry timeout value. The adaptive tuning of this parameter is the subject of future study.

5.4 Performance results summary

We find that, among the three paging protocols, Domain paging supported the highest paging load at a given paging latency. Foreign Agent paging is also able to support a fairly high load in comparison to Home Agent paging. However,



(a)Home Agent (b) Foreign Agent (c) Domain (d) Hier. Algo.

Figure 7: Impact of paging algorithms (SULA, paging area=6)

Home Agent paging can still be useful in small networks with low paging load due to its simplicity.

With respect to the number of updates to the home agent, the Domain paging protocol results in the least number of updates with about 70% reduction compared to the case when paging is not used; Foreign Agent paging results in a reduction of about 20%. These results highlight the efficiency of a truly localized paging architecture. We also find that optimal sizing of paging area is impacted by a trade-off between update rates and paging processing latencies.

Finally, among the three paging algorithms, we find that while the Last-location algorithm performs the best in Foreign Agent and Domain paging, it is the worst performer in Home Agent paging. This unexpected result for the Home Agent paging architecture is explained by the fact that Last-location exacerbates the source of bottleneck in Home Agent paging, viz., paging initiation processing.

6. RELIABILITY

In this section, we perform a preliminary unavailability analysis to illustrate the qualitative differences in the reliability of the three paging protocols. We calculate the unavailability based on the time the network is unavailable to a mobile user. This results in a conservative estimate since this assumes all users are actively using the network during failures, a likely scenario in peak-hour usage.

In our availability analysis, we are primarily interested in the failure of three components: home agents, Foreign Agents, and Domain Routers. In our failure model, we assume that failures of these components are independent, and multiple simultaneous failures do not occur. For simplicity, we also assume that the mean recovery time for each of these components are the same in the three paging protocols and be denoted t_{HA}, t_{FA} and t_R respectively. Let the mean time between failure for each of the components be denoted by t_M.

Let N denote the average number of routers in a domain in the path from the home agent to any Foreign Agent. Let α be the percentage of the users in a domain that move out of the domain coverage while still in session and δ be the percentage of users under the coverage of a foreign agent that move out of the foreign agent coverage in standby mode.

Network unavailability probability[5], U, in our failure model can be defined as

$$U = \Sigma_{i \in component} \frac{\text{Mean recovery time(i)}}{\text{Mean time between failure(i)}}$$

In the case of Home Agent paging,

$$U_{HA} = (t_{HA} + t_{FA} + Nt_R)/t_M,$$

since users are impacted equally when an home agent, foreign agent or a router in the path between home agent or foreign agent fails.

In the case of Foreign Agent paging,

$$U_{FA} = (t_{HA} + t_{FA} + \delta t_{FA} + Nt_R)/t_M.$$

As compared to Home Agent paging, the extra term, δt_{FA}, models the impact of failure of foreign agents that serve as paging initiators for users who move away from the foreign agent and thus are unreachable.

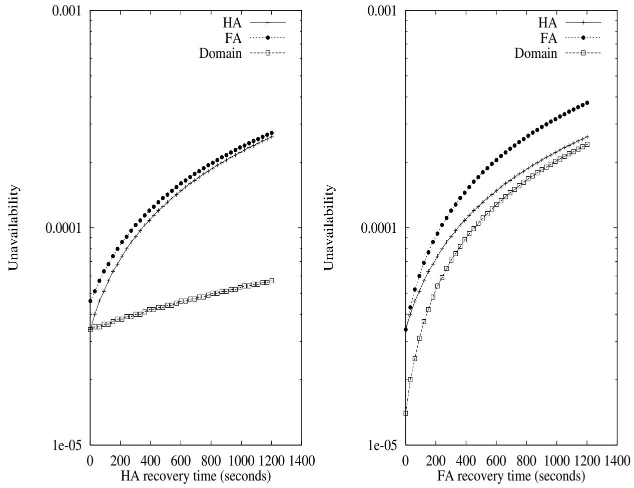
In the case of Domain paging,

$$U_{Domain} = (\alpha t_{HA} + t_{FA} + Nt_R)/t_M,$$

since users are affected by their home agent failure only when they move out of the domain while still in session (otherwise, they have no home agents).

For the performance examples in this section, we use $N = 2$, $\alpha = 0.1$, $t_M = 1$ month, and $\delta = 0.5$. We plot the sensitivity of U for each of the paging protocols to home agent and foreign agent recovery times (all protocols are equally impacted by the failure of routers in our model). Figure 8(a)

[5]Availability would simply be 1 - unavailability



(a) HA recovery time (b) FA recovery time

Figure 8: Unavailability probability

plots the unavailability probability of the three paging protocols in log scale versus home agent recovery time with $t_R = 30$ and $t_{FA} = 120$ seconds. Typical unavailability for systems such as the mobile switching center ranges between 0.0001 and 0.00001 [10]. It is clear that the Domain paging protocol's unavailability is lower than the other two (thus, availability is higher) and this gain can be primarily attributed to the fact that the home agent is only needed for a small subset (α) of the users. One way to reduce the unavailability of Foreign Agent and Home Agent paging is to improve the reliability of home agents through replication or other means.

Figure 8(b) plots the unavailability probability of the three paging protocols in log scale versus foreign agent recovery time with $t_R = 30$ and $t_{HA} = 120$ seconds. In this case, the difference in unavailability between the protocols is not significant with the unavailability probability of Home Agent and Domain paging approaching the same value asymptotically while Foreign Agent paging performs the worst due to the additional impact of the foreign agent failure as a paging initiator failure.

While the above simple analysis illustrates the important differences in reliability in the three paging protocols, there is scope for future work both in terms of generalizing the analysis as well as designing protocols enhancements to further improve the reliability of all three protocols.

7. RELATED WORK AND OTHER ISSUES

There has been a flurry of activity recently on IP paging support. The P-MIP [25] approach is very similar to the Foreign Agent paging approach discussed in this paper where Foreign Agents serve as paging initiators. The Domain paging architecture presented in this paper is a generalization of the HAWAII [14, 15] architecture. The Cellular IP [17] approach also uses a fully distributed paging architecture that is similar to Domain paging; however, Cellular IP controls who initiates paging through static placement of paging caches rather than dynamically determining the paging initiator as is done in Domain paging. The Regionalized paging [6] protocol can also be viewed as Domain paging, albeit with tunnels to skip over routers that do not understand paging. This paper is the first to comprehensively present, analyze, and discuss the various paging protocols in an integrated framework.

One of the important issues to consider in paging protocol design is the use of IP multicast. Recall that multicast could be used to distribute paging request messages to the base stations in the paging area. The multicast group address serves as a succinct representation of all the IP base stations in the paging area. Furthermore, administratively scoped addresses in Foreign Agent and Domain paging avoids unnecessary conflicts with other multicast sessions - a disadvantage for use of multicast in Home Agent paging. The use of IP multicast allows for tremendous flexibility. For example, if a new base station is installed due to a cell split, the base station just creates/joins the appropriate multicast group. Base stations can also regroup into different paging areas, if necessary, by joining the corresponding multicast groups. These changes are transparent to other routers in the domain; the multicast routing protocol will automatically compute the new multicast tree for each of the paging areas. Furthermore, implementation of sophisticated paging algorithms such as Hierarchical paging is simplified and only requires the configuration of appropriate multicast addresses for the different levels.

The price for this flexibility in network management is the complexity of IP multicast. In our testbed, we used the DVMRP multicast routing protocol. This protocol works by a flood and prune mechanism to maintain the multicast tree; the multicast route cache values at each node timed out fairly quickly and resulted in unnecessary flooding within the domain. The use of multicast protocols such as PIM sparse-mode [5] or the more recent EXPRESS approach [7] should help alleviate this problem.

Another issue is the amount of buffering needed for incoming packets while paging is being performed. We believe that for most applications a buffer size of one packet for each mobile host that is being paged is sufficient. For example, typical voice-over-IP applications using the Session Initiation Protocol would send a SIP Invite packet and wait for a response before sending more packets; an invoke of a TCP session to the mobile host would result in a TCP SYN packet being sent first. Thus, for these common applications, one packet per paged-host buffer would be sufficient, making adequate buffer sizing fairly inexpensive.

The IP paging service proposed in this paper is similar to the Address Resolution Protocol (ARP) mechanism present in today's IP routers and hosts. The key difference is that while ARP is used to locate a host's link layer address on a broadcast LAN, paging can be thought of as a service to locate a host's local area network (in other words, current base station) from a router that is one or more hops away. Also, there are some similarities between message delivery in ad-hoc wireless networks and paging. However, the mobile nature of routers themselves in ad-hoc networks results in the need for some form of global message broadcast, while in the case of paging, locating the mobile host attached to a fixed infrastructure results in a more localized and directed

Goals	HA	FA	Domain
Simplicity	yes	yes	somewhat
Flexibility	somewhat	somewhat	yes
Scalability	somewhat	yes	yes
Efficiency	no	somewhat	yes
Reliability	somewhat	somewhat	yes
Deployability	somewhat	yes	somewhat

Table 2: Paging protocols

search.

Finally, note that paging functionality is not necessary for reducing battery power consumption in mobile hosts that only initiate sessions, for example, when using applications such as web browsing. This is because the host can implicitly identify its location every time it initiates a session and can be powered off when not in use. In fact, for these applications, Mobile IP is also not necessary since sessions last only a short duration; every time a mobile host initiates a session from a new location, it could obtain a new address using, for example, DHCP, and contact the appropriate server directly. However, Mobile IP and paging become critical in future all-IP networks with voice-over-IP and messaging applications[6]. In this case, sessions could be server initiated and mobile hosts might stay powered on for long durations, possibly awaiting incoming sessions.

8. CONCLUSION

In this paper, we first introduced the concept of paging as an IP service. IP paging enables a common infrastructure and protocol to support the different wireless interfaces such as CDMA, GPRS, wireless LAN, avoiding the duplication of several application layer paging implementations and the inter-operability issues that exist today. We then proposed three paging protocols, Home Agent paging, Foreign Agent paging, and Domain paging. A comparison of the three protocols is presented in Table 2. Domain and Foreign Agent paging have broad applicability with trade-offs going either way, while Home Agent paging is applicable only to limited scenarios - for example, small networks with complete administrative control.

9. REFERENCES

[1] K. Calvert, M. Doar and E. Zegura, “Modeling Internet Topology,” in *IEEE Communications Magazine*, June 1997.

[2] IS-657, “Packet Data Service Option Standard for Wideband Spread Spectrum Systems,” 1996.

[3] Digital cellular telecommunication system, “General Packet Radio Service, Service description - Stage 2,” GSM 03.60, Version 6.0, ETSI, 1998.

[4] IEEE 802.11 “Wireless LAN Medium Access Control and Physical Layer specifications,” 1999.

[5] D. Estrin et al., “Protocol independent multicast-sparse mode,” in RFC 2117, 1997.

[6] While email uses client-based polling, email server “push” avoids unnecessary polling, reducing battery consumption.

[6] H. Haverinen, ”Mobile IP Regional Paging,” Internet Draft, Work in Progress, June 2000.

[7] M. Holbrook and D.R. Cheriton, “IP Multicast Channels: EXPRESS Support for Large-Scale Single Source Applications,” in SIGCOMM’99.

[8] D. Lam, D.C. Cox, and J. Widom, ” Teletraffic Modeling for Personal Communications Services,” IEEE Communications, 35(2):79-87.

[9] D. Meyer, “Administratively Scoped IP Multicast,” Request for Comment 2365, July 1998.

[10] K. Murakami, R. Buskens, R. Ramjee, Y-J Lin, and T. La Porta, “Design, Implementation, and Evaluation of Highly Available Distributed Call Processing Systems,” in proceedings of FTCS’98.

[11] C.E. Perkins, “IP Mobility Support,” RFC 2002.

[12] C Perkins, G Montenegro, Pat Calhoun, ”Mobile IP Regionalized Tunnel Management”, Internet Draft, Work in Progress, August 1999.

[13] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and S-Y Wang, “HAWAII: a domain based approach for supporting mobility in wide-area wireless networks,” in proceedings of ICNP’99.

[14] R. Ramjee, T. La Porta, L. Salgarelli, S. Thuel, K. Varadhan, and L. Li, “IP-based access network infrastructure for next generation wireless data networks,” in IEEE Personal communications magazine, Aug 2000.

[15] R. Ramjee, L. Li, and T. La Porta, ‘Paging support for IP mobility,” Internet Draft, Work in Progress, Jul 2000.

[16] “Stanford University Mobile Activity TRAces” available at http://www-db.stanford.edu/pleiades/SUMATRA.html

[17] A. Valko, A. Campbell, and J. Gomez, “Cellular IP,” Internet Draft, Work in Progress, Nov 1998.

[18] T. Woo, T. La Porta, J. Golestani, and N. Agarwal, “Update and Search Algorithms for Wireless messaging: Design and Performance”, in INFOCOM’98.

[19] S. Sarkar, E. Tiedemann, “cdma2000: Battery Life Improvement Technqiues,” Proc. IEEE INFOCOM’00.

[20] T. Simunic, L. Benini, P. Glynn, G. De Micheli, “Dynamic Power Management for Portable Systems,” Proc. ACM MOBICOM’00.

[21] D. Wisely, W. Mohr, J. Urban, “Broadband Radio Access for IP-Based Networks (BRAIN) - A key enabler for mobile internet access,” Proc. IEEE PIMRC’00.

[22] K. Svanbro, J. Wiorek, B. Olin, “Voice-over-IP-over-Wireless,” Proc. IEEE PIMRC’00.

Routing entry	Paging entry	Host state	Router action
Y	Y/N	Active	Regular IP forwarding
N	Y	Standby	Paging processing
N	N	Null	Drop if no default route

Table 3: Router processing for a given mobile host

[23] G. Patel, S. Dennett "The 3GPP and 3GPP2 Movements Toward an All-IP Mobile Network," IEEE Personal Communications Magazine, Vol. 7, No. 4, Aug. 2000.

[24] R. Stanley, "The Influence of MTSO Placement on Wireless Network Cost," Proc. IEEE PIMRC'00.

[25] X. Zhang, J. Castellanos, A. Campbell, K. Sawada, and M. Barry, "P-MIP: Minimal paging extensions for Mobile IP," Internet Draft, Work in Progress, July 2000.

Appendix A: Detailed Operation of Domain Paging

We present the detailed operation of Domain paging in this section. We distinguish two types of entries, routing and paging, in the base stations/routers for forwarding packets to the mobile host. The routing entry allows for regular forwarding of IP packets while the paging entry enables paging processing. The operation of the router or base station with respect to these entries is summarized in Table 3.

Before we discuss paging processing, we first present the processing required to establish paging entries in the base station and routers. Recall that a mobile host in standby state updates the network with its paging area whenever it crosses into a new paging area. The pseudo-code in Figure 9 illustrates the processing of such a paging update message, containing the mobile host address and the multicast group address (MGA) of the paging area. These protocols messages are sent on a well-known UDP port. The code shown in Figure 9 and others presented in this section are executed in each base station/router independently. Observe that the execution of this code results in the processing of the update message from the base station, in a hop-by-hop manner, up to the domain root router establishing the latest paging area information in these nodes. Thus, this code maintains the following *Update* property:

- *Update:* Up-to-date paging entries are maintained for each mobile host in all the nodes in the path from its last attached base station in the paging area to the domain root router.

The pseudo-code for the main paging processing including the determination of which node initiates the paging request is shown in Figure 10. This maintains the following *Upstream* property:

- *Upstream:* Paging is initiated from routers with paging entry for the mobile host only if packets arrive from the Domain Root Router along the upstream interface.

```
1. Receive protocol message from neighbor with
     (MH IP ADDRESS, MGA) on Port A
2. If I am the Domain Root Router
3.   Set entry to (MH IP ADDRESS → MGA, Port A)
4. else
5.   Set entry to (MH IP ADDRESS → MGA, Port A)
6.   Forward to upstream neighbor along default route
7. endif
```

Figure 9: Paging update processing in base station/router

Maintaining the *Upstream* property is essential in order to be able to page the mobile host using the up-to-date paging area information. This is because stale paging entries created by old paging updates may exist in internal routers for several reasons including topology or routing changes. In order to avoid paging using stale paging entries for packets originating inside the domain and destined for a mobile host in standby state, these packets will first be forwarded along the default route to the domain root router. Paging is then initiated only when these packets arrive from the correct upstream node (line 2 in Figure 10).

```
1. IP packet for MH arrives at node with
     entry (MH IP address → MGA, Port A)
2. if (packet arrives from default route port
     or I am Domain root Router)
3.   if ((no refresh on Port A) /* Failure */
4.      or (page queue < β)/*lightly loaded?*/
5.      or (I am a base station)) /* Initiate Paging */
6.        buffer packet and send page to MGA
7.        increase retry counter and set retry timer
8.   else    /* Push paging initiation downstream */
9.        route the packet through Port A
10.   endif
11. else
12.   forward packet along default route to DRR
13. endif
```

Figure 10: Paging initiation in base station/router

Note that, paging is initiated at a router when there is a potential failure due to lack of soft-state refresh messages from downstream node (line 3) or when the queue size of outstanding page requests is less than an administratively configured value, β (line 4). The latter check is necessary for load balancing when the router is lightly loaded[7]. Otherwise, paging is initiated from the user's previous base station by default (line 6). In Appendix B, we discuss how β can be chosen; for now, observe that β can have any value between zero and infinity (representing always paging from base station and always paging from the router). A fractional value for β can also be implemented by using probabilistic techniques.

When the base stations in the paging area receive the paging request packet through multicast, they send out a link-layer page message on the air interface. The mobile host sends back a paging response message which is then processed in

[7] We also considered a simpler alternative where the router determined whether it was the paging initiator probabilistically and found it to have inferior performance.

the base station and all the routers between the base station and the paging initiator. This processing is shown in Figure 11. Observe that the processing of this code at each node from the base station hop-by-hop to the paging initiator helps maintain the following *downstream* property:

- *Downstream:* Processing of paging response establishes up to date routing entries along the path from the mobile host to the paging initiator.

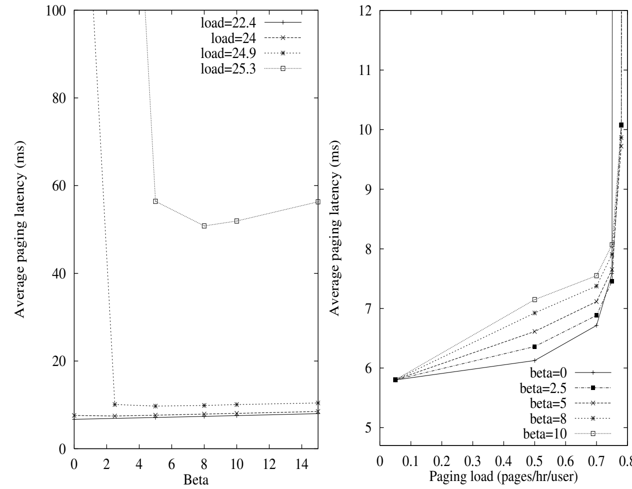
```
1. Receive protocol message from neighbor with
      (MH IP ADDRESS, MGA) on Port A
2. If I am the paging initiator
3.   Set entry to (MH IP ADDRESS → Port A)
4.   Forward buffered packets
5. else
6.   Set entry to (MH IP ADDRESS → Port A)
7.   Forward response hop-by-hop towards initiator
8. endif
```

Figure 11: Paging response processing in each base station/router

The correctness of the Domain paging protocol can be easily established from the *Update, Upstream,* and *Downstream* properties. The *Update* and *Upstream* properties imply that the router/ base station initiating the paging has the latest (up to date) paging entry for a given mobile host. The *Downstream* property guarantees that the routing path from that mobile host to the paging initiator is up to date. Combining the previous two statements, the routing path from the domain root router through the paging initiator to the mobile host is up to date after paging processing is complete, resulting in correct delivery of data packets.

Appendix B: Impact of varying β

We consider the impact of β on Domain paging performance. Recall that β (see Figure 10) represents the threshold for the number of outstanding paging requests at a router below which the router initiates paging. In Figure 12(a), we plot the average paging latency versus β. We make two important observations. First, for a given load, as β increases the average paging latency reduces upto a certain value of β after which the average paging latency starts increasing again. The reason for this behavior is that at low values of β, the base stations becomes the processing bottleneck while for high values of β, the router becomes the processing bottleneck. Thus, *an optimal value of β exists that minimizes average paging latency by balancing the processing load between the base stations and the router.* Second, as the load increases, the optimal value of β also increases (note the way the curves shift to the right in Figure 12(b)). This is because higher values of β imply that the router is taking a higher processing burden. At high paging load, one would expect the router to take a higher processing burden in order to alleviate processing load at the base stations. Thus, at high paging load, higher β values lead to better load balancing among the base station and routers. Note that this behavior has an unfortunate side-effect: at low load, higher β values result in higher latencies. Thus, Domain paging would benefit from an adaptive algorithm for tuning β depending on



(a) SULA (β),Domain (b) SULA (load),Domain

Figure 12: Impact of β

the paging load. This is the subject of future study. For now, we note that the penalty for choosing a higher β value is not significantly costly in terms of latency at low load while the higher β value supports higher loads that lower values of β cannot. In the traces we examined, performance was good for a wide range of β values from 2 to 10.