An Adaptive Per-Host IP Paging Architecture

Claude Castelluccia
INRIA Rhône Alpes, PLANETE Team
655 avenue de l'Europe, Montbonnot
38334 Saint Ismier Cedex France
claude.castelluccia@inria.fr

Pars Mutaf
INRIA Rhône Alpes, PLANETE Team
655 avenue de l'Europe, Montbonnot
38334 Saint Ismier Cedex France
pars.mutaf@inria.fr

ABSTRACT
IP Paging has received considerable attention recently. The IETF has decided to develop an IP Paging protocol and some IP Paging protocols have been proposed by researchers [13, 12]. However all of these proposals use static and manually configured paging areas. We argue that the size and the shape of paging areas are very critical for the performance of a paging system. A system that allows each host to use a paging area that adapts to its mobility and communication patterns will perform better. This paper proposes to extend the IETF IP Paging functional architecture to support *adaptive and per-host* paging areas. We introduce a new functional module, the *Paging Area Configuration Agent* (PACA), that automatically configures the paging area relative to each cell of a cellular domain.

1. INTRODUCTION
In this paper we consider wireless access to the Internet via cellular IP networks [2, 11]. We envision the next generation of cellular wireless networks as pure IP-based networks where base stations are IP-routers. In these networks, mobility should be handled at the IP layer. Different mobility management protocols have been proposed for such networks (e.g. HAWAII [10], CellularIP [2]). Mobile IP [9, 5] is of course a good candidate but needs some extensions to be used efficiently in such environments. Mobile IP requires that a mobile host sends a *location registration* message to its Home Agent whenever it moves from one point of attachment to another one. These location registrations are required even if the mobile host is in *dormant mode* (i.e. idle) while moving. This signaling cost can become quite significant as the number of mobile hosts increases and the cell sizes get smaller. Furthermore with Mobile IP, a dormant host has to wake up each time it moves to send a registration. This is obviously not very power-efficient.

Several proposals have been made to extend Mobile IP with paging [13, 12]. Paging is based on the division of the network in several *paging areas*. A dormant mobile host only sends a location registration message to its Home Agent when it enters a new paging area. When the Home Agent needs to contact the mobile host for packet delivery, the host is paged in its current paging area. The host then reports its exact location to its Home Agent using standard Mobile IP. The IETF Seamoby Working Group [4] has been chartered to develop an IP Paging protocol. Recently, the working group has proposed a functional architecture and is currently working on the protocol.

Current IP Paging protocol proposals use static and manually configured paging areas. We argue that a system that allows each host to use a per-host paging area that adapts to its mobility and communication patterns will perform better than a scheme that uses static paging areas (in terms of signaling load and power consumption). This paper proposes to extend the IETF IP Paging functional architecture to support *adaptive per-host* paging areas. We do not propose a new IP Paging protocol but rather an extension to be used with existing protocol proposals. Section 2 reviews the IETF IP Paging current architecture. Section 3 presents our adaptive per-host IP Paging extension. Section 4 describes and analyses the new entity and algorithm that automatically configure paging areas. Section 5 presents some related work. Finally, Section 6 concludes the paper.

2. IETF IP PAGING
2.1 Overview
Current mobility management schemes for IP networks, such as Mobile IP [9, 5], track the mobile hosts' locations through registration procedures. As a result, a mobile host has to register with its Home Agent each time it changes its point of attachment even if it is in a dormant mode.

In cellular systems such as GSM [8], the network tracks the mobile hosts' locations through *paging/registration* procedures. In these systems, cells are grouped into *paging areas*. A host that is dormant only registers with the network when it moves out of its paging area. As a result:

- Signaling load is reduced because a dormant host does not have to register each time it changes its point of attachment.
- Power consumption is reduced because a dormant host wakes up to send a registration message only when it moves out of its current paging area.

There is therefore a need for IP-based cellular networks that support paging at the IP layer. One of the goals of the IETF Seamoby Working Group is actually to develop an *IP Paging* protocol. The working group has delivered two RFCs so far: one describing the problem and the motivations of IP Paging [6] and the other presenting the requirements and the IP Paging functional architecture displayed in Figure 1 [7].

The IETF IP Paging functional architecture is composed of the following elements:

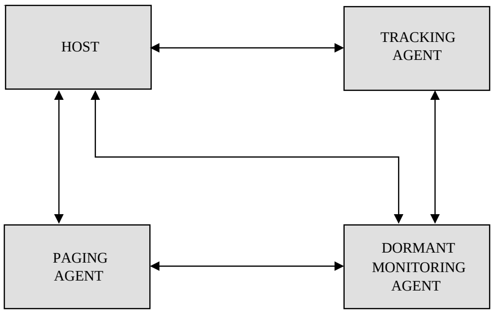


Figure 1: IETF paging functional architecture

- *Host* - A standard IP host.
- *Paging Agent* - The Paging Agent is responsible for alerting the host when a packet arrives and the host is in dormant mode.
- *Tracking Agent* - Tracking Agent is responsible for tracking a host's location while it is in dormant mode or active mode, and for determining when Host enters inactive mode.
- *Dormant Monitoring Agent* - The Dormant Monitoring Agent detects the delivery of packets to a host that is in dormant mode (and thus does not have an active L2 connection to the Internet).

In this architecture, a cellular network is divided into *domains*. Each domain contains a Tracking Agent and a Dormant Monitoring Agent. Domains are themselves subdivided into *paging areas* (defined as a set of cells), each being identified by a unique identifier and Paging Agent.

2.2 Protocol Description

In this architecture, a typical IP Paging protocol will operate as follows [1]:

2.2.1 Registration

When a host moves into dormant mode or when it moves out of its current paging area, it registers with the domain's Tracking Agent. This registration specifies the host's identity (home address for example) and the identifier of its current paging area. Upon reception of a paging registration, the Tracking Agent creates an entry that binds the host's identity with the Paging Agent that is in charge of the paging area specified in the registration. It also sends a report to the domain's Dormant Monitoring Agent indicating that the host has entered dormant mode.

2.2.2 Packet Delivery

When the Dormant Monitoring Agent receives packet(s) for the host, it buffers them (because the host is registered as dormant). The Dormant Monitoring Agent then asks the Tracking Agent the host's current Paging Agent. Finally, the Dormant Monitoring Agent requests the host's Paging Agent to page the host. The Paging Agent pages the host in its registered paging area. The host wakes up, moves into active state and registers its current point of attachment with the Dormant Monitoring Agent.The Dormant Monitoring Agent then forwards the packets to the host.

[1]Note that the logical architecture makes no commitment to any physical implementation of these functional entities. A physical implementation may merge particular functional entities.

3. ADAPTIVE PER-HOST IP PAGING

3.1 Motivations

Traditional cellular systems, such as GSM [8], use *aggregated* and *static* paging areas. Typically an operator configures the paging areas of a network (i.e. size and shape) according to the characteristics of the average host. All hosts use the same paging areas independently of their communication and mobility pattern. The *size* and *shape* of a given paging area are very critical for the performance of a paging system: A large paging area will generate a high paging cost and will consequently degrade the overall performance. A small paging area will not provide the complete benefit of the paging mechanism since it will generate unnecessary registrations. The optimal location area size of a host depends on its average speed, the average incoming call rate and its distance to its Tracking Agent [3]. On the other hand, a misconfigured paging area shape will generate a higher registration cost because a host will have to register more often than necessary. This problem is illustrated by Figure 2. This figure shows two paging areas that have the same size (18 cells) but different shapes. With the first configuration (Figure 2a) the mobile host moves out of the paging area after only 8 moves. With the second configuration (Figure 2b) the mobile host moves out of its paging area after 13 moves. The second configuration is obviously better since it reduces the number of registrations. The optimal location area shape of a host depends on its mobility pattern. Since mobile hosts have different and time-varying communication and mobility patterns, we argue that mobile hosts would benefit from systems that customize and adapt the paging area to each host's characteristics. We therefore propose to extend the IP Paging architecture defined in RFC3154 [7] to support *adaptive* and *per-host* paging areas[2].

3.2 Design Overview

3.2.1 Paging Area Model

In traditional paging systems, paging areas are static and non-overlapping. Each paging area is then defined by an identifier: the *paging area identifier*. This identifier is broadcast by each cell that belongs to this paging area. A mobile host just has to monitor this identifier to detect that it has moved out of its current paging area.

In adaptive per-host paging systems, each host has its *own customized* paging area that is defined as the *IP addresses' list* of the composing cells. A host detects that it moves out of its paging area when the address of the base station is attached to does not belong to its current paging area. Whenever a host moves out of its current paging area, it gets a new one that is built *around* its current location and that is *customized* to its mobility and communication patterns. Figure 3 illustrates an adaptive per-host paging system. In

[2]It is noteworthy that one of the requirements specified in RFC3154 is the support of customized paging areas.

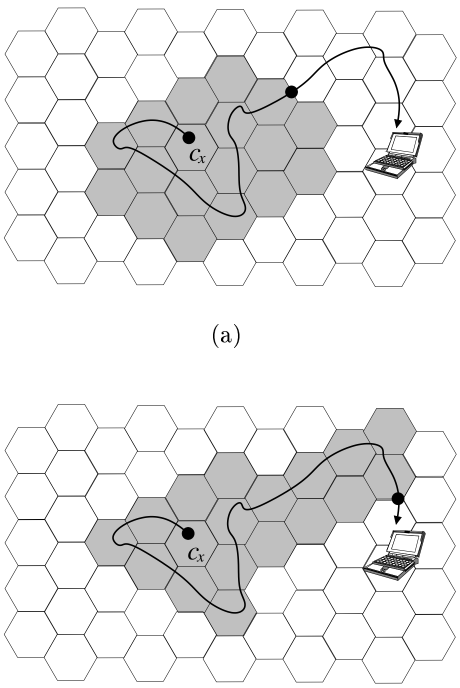


Figure 2: Two different paging area shapes (solid dots represent location registrations).

this figure a mobile host enters into dormant mode at cell c_1. It then gets a paging area PA_1, that shape and size are customized to its mobility and communication pattern. The host then moves out of PA_1 at cell c_2. At this point it gets a new customized paging area, PA_2. It then moves out of PA_2 at cell c_3 and gets the new paging area, PA_3 and so on.

3.2.2 Paging Area Configuration

In Adaptive Per-Host IP Paging, each host uses a paging area that size and shape are dynamically adjusted. As described in [3], the "optimal size" of a paging area essentially depends on the host's communication pattern (i.e. incoming data-session rate) and its mobility speed (i.e. how often it moves from one cell to another). A host's paging area size can actually be computed by the host itself or by the network. However it is obviously easier and more scalable for the host to monitor these parameters (i.e. mobility and communication characteristics) and computes its optimal size.

The "optimal shape" of a paging area depends essentially on the host mobility pattern. In fact a host that is driving along a highway would benefit from a linear paging area. In contrast, a host moving in an urban area with a random pattern would benefit from a symmetrical, i.e. a circular, paging area. The mobility pattern of host depends on the geographic layout of the environment it is roaming in. There are two possible configuration alternatives: In *host configuration*, each host derives its paging area shape from its mobility pattern. A host then needs to keep in memory the

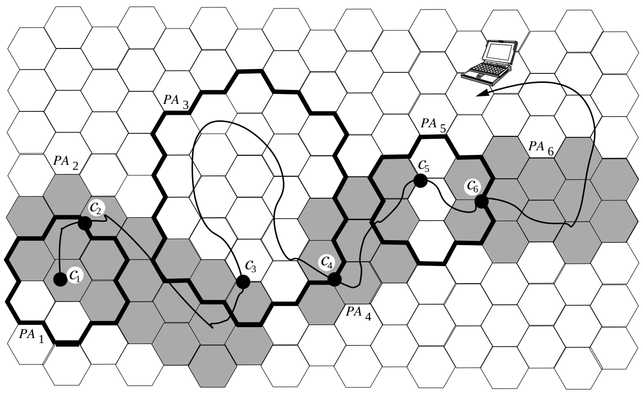


Figure 3: Adaptive paging area sizes and shapes (solid dots represent registrations).

history of its movements and derives from it the optimal paging area in each network it visits. This solution has two drawbacks: (1) the memory requirement might be very large and (2) the history might not always be available (i.e. when the host visits a network for the first time) or out-of date (for example when base stations are added or removed). Another possibility is *network configuration* where the network derives an *aggregated* paging area shape from the mobility pattern of the hosts in its coverage. This solution might not be as optimal as the previous one because it does not capture the particular mobility pattern of each host. However it reduces the memory requirement of mobile hosts. Additionally it does not require that each host builds its own mobility history. As a result a host can obtain an optimal paging area even in networks it did not visited previously. This solution is more practical. We have therefore decided to adopt it for our proposal.

To summarize, in our architecture, *the paging area size is computed by the mobile nodes whereas the paging area shape is computed by the network.* Given these design choices, we are faced with the two following problems:

1. *How does a host compute its optimal paging area size?* This paper does not address this problem. We make use of the algorithm that we developed in [3].

2. *How does the network compute the optimal paging area shape in a given geographical area?* A low-cost and scalable solution to this problem is proposed in Section 4.

3.2.3 IP Paging Functional Architecture Extension

This section describes an extension to the the functional architecture defined in [7] to support adaptive per-host paging (see Figure 4). We define a new entity, the *Paging Area Configuration Agent* (PACA), that is responsible for computing for each cell of its domain the "optimal" paging areas. The PACA receives inputs from the mobile hosts roaming in its domain and derives from them the optimal paging areas using the algorithm described in Section 4. This new entity has two interfaces: one with the Tracking Agent and one with the host module. The PACA is described in more details in Section 4

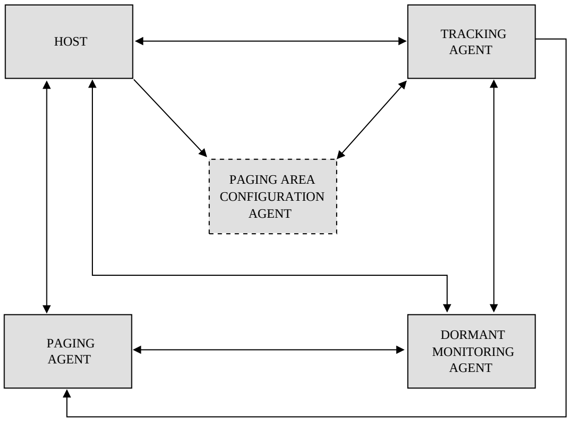


Figure 4: Extended paging functional architecture.

3.3 Protocol Description

A typical adaptive per-host IP Paging protocol will operate as follows:

3.3.1 Registration

When a host moves into dormant mode or when its moves out of its current paging area, it computes its optimal paging area size, S, using the algorithm defined in [3]. It then registers with the domain's Tracking Agent. This registration specifies the host's identity, paging area size (S) and current point of attachment. Upon reception of a paging registration, the Tracking Agent requests the domain's PACA the optimal Paging Area corresponding to the host's current point of attachment and size S. It then sends a report to the domain's Dormant Monitoring Agent indicating that the host has entered dormant mode. The PACA derives the paging area and sends it to the host and the corresponding Paging Agent, i.e. the Paging Agent in charge of the host's current point of attachment. As a result of this registration phase: (1) The host has obtained its paging area (i.e. the list of the cells composing its paging area). (2) The Tracking Agent has registered the binding between the host and its current Paging Agent. (3) The Paging Agent has registered the host's current paging area.

3.3.2 Packet Delivery

Packet delivery is carried out the same way as described in Section 2.

3.3.3 Paging Implementation

When packets arrive for a dormant host, the network (more specifically the Paging Agent) needs to page it. This is performed by broadcasting a paging request to the cells of the host's current paging area. Upon reception of this paging request, the host wakes up, moves into active state and registers with the network. There are different ways to implement this paging process: First, the Paging Agent can simultaneously *unicast* the paging request to each of the paging area's cells. While this solution is probably the simplest to implement and configure, it is certainly the less efficient in term of signaling load generated in the network. Second, the Paging Agent can *broadcast* the paging request to the paging area cells using a multicast routing protocol as in [10]. In this proposal, a multicast address is assigned to each Base Station of a paging area. A Paging Agent pages a mobile host by sending a paging request to the paging area multicast address. This solution generates a multicast signaling load that might be quite significant especially if adaptive paging areas are considered.

We propose that paging messages sent by the Paging Agent to the different cells of the host's paging area be carried out in *Small Group Multicast* (SGM) style [1]. In SGM schemes, the source encodes the list of destinations in the IP header, and then sends the packet. Each router along the way parses the header, partitions the destinations based on each destination's next hop, and forwards a packet with an appropriate header to each of the next hops. When there is only one destination left, the packet could turn into a normal unicast packet, which can be unicast along the remainder of the route. This multicast delivery method is well suited to adaptive per-host IP Paging schemes since the paging area information (i.e. addresses of composing cells) is hold by the Paging Agent. SGM paging is more efficient when paging area sizes are small. When a destination paging area contains a large number of cells, the initial SGM paging packet header might be quite large. However, in adaptive per-host IP Paging, a paging area is large when the mobile host rarely receives incoming packets i.e. when the probability of paging is small. In situations where it is important to have very large paging area sizes, it is possible to divide the paging process in two or more SGM processes in which a mobile host is searched in one subset of the paging area by each process.

3.3.4 Paging Area Coding

In adaptive IP Paging, a paging area is defined as the IP addresses's list of its composing cells (see Section 3.2.1). When a dormant host registers with the network, it receives an acknowledgement that contains its new paging area as a list of IP addresses. When a paging area size is large, the transmission of this information over wireless links may consume a significant portion of the bandwidth, and may increase the packet error probability. Furthermore, the reception of large packets may have an impact on battery consumption on hosts. As a result, it is important to have a scheme that compresses the paging area identifiers (i.e the lists of addresses).

We propose a bitmapping procedure based on the cell addressing scheme illustrated in Figure 5. A domain is divided into clusters. A cell is coded with 128 bits where the 128-N least significant bits represent the cluster ID it belongs to. The remaining N bits identify this cell within the cluster. Each cluster contains N cells therefore only one bit is needed to identify a cell in a given cluster. Each base station broadcasts this information in its Router Advertisement messages. This is manually configured by the operator (alternatively, an autoconfiguration method similar to IPv6 router renumbering can be used for flexibility of administration). Then, a number of cells (S in Figure 5) which belong to the same cluster can be coded as a single 128 bit identifier: a cluster ID and the bitwise OR of the bits corresponding to the cells included in the paging area. Upon movement, a mobile host can check if it has crossed the boundaries of a given paging area, as follows:

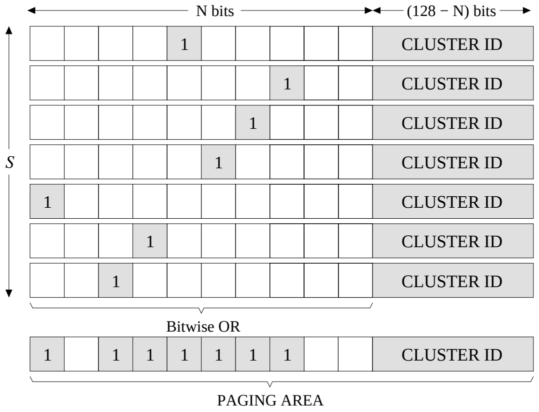


Figure 5: Paging area coding scheme.

if $\neg(c \wedge PA)$
then send registration

where c and PA are the current cell and paging area of the mobile host. Different parts of a paging area may fall in different clusters. Clearly, in this case the coding performance will be reduced since more than one 128 bit identifiers will be needed to represent the paging area. However, N can be chosen large in order to reduce the probability of such a situation. Note that since this addressing scheme is not used for routing, the cluster IDs are reusable. However, any two clusters having the same ID, should be enough distant from each other in order to avoid any confusing paging area information.

4. PAGING AREA CONFIGURATION AGENT

The Paging Area Configuration Agent (PACA) is responsible for configuring the "optimal" paging area shape relative to each cell of a domain. It receives feedback messages from the mobile hosts roaming in its domain and uses them to configure the domain's paging areas. If the mobility patterns of the hosts change, because a new road has been built or a new base station has been installed, the PACA adapts the paging areas automatically.

This section describes and analyses the algorithm that the PACA uses to automatically configure the paging areas.

4.1 Problem Statement

The PACA configures for each cell, c_x, of its domain the "optimal" paging area shape. By "optimal" shape, we mean the shape that minimizes the average number of registration messages sent by the hosts that request a new paging area at cell c_x.

This optimization problem can be stated as follows: *Given the set of hosts that request a new paging area of size S at cell c_x, what is the paging area (i.e. set of S cells) that maximizes the average time spent by these hosts in this paging area?*

Intuitively, this paging area is composed of the S cells that a host that is currently in cell c_x will most likely visit without leaving the paging area. In order words, if $P_x(c_i)$ is the probability that a host which is currently in cell c_x will visit cell c_i without *leaving the paging area*, the optimal paging area is composed of the S cells with the largest $P_x(c_i)$.

The rest of this section presents a paging area auto configuration algorithm that provides a low-cost solution to this problem. This algorithm is composed of two major parts: *sampling* and *paging area composition*. Sampling is the process of extracting the transition probabilities in each cell (i.e., the probabilities of moving from one cell to each of its neighboring cells). Composition is the process of configuring a paging area relative to a given cell.

4.2 Sampling

Sampling is the most challenging part of paging area auto-configuration. A sampling algorithm should be low-cost and scalable. By low-cost, we mean that the algorithm should not call for excessive CPU operations which will degrade the performance of hosts and/or the network. We define the following sampling procedure:

We define a sample as an ordered pair of adjacent cells. Samples are randomly generated by mobile hosts. For example, if a host has visited the following 5 successive cells $(c_1, c_2, c_3, c_4, c_5)$, then $[c_1, c_2]$, $[c_2, c_3]$, $[c_3, c_4]$ and $[c_4, c_5]$ are 4 different samples. A given sample $[c_i, c_{i+1}]$ gives information about the transition probabilities in cell c_i.

Each PACA is responsible for collecting samples in its domain. A host in a given domain send samples to its current PACA at a very low rate i.e. once every 200 registrations As a result, the overall signaling overhead due to sampling is in the order of 0.5% of the location registration traffic (which is negligible) [3]. The impact of sampling on battery consumption is negligible as well. Since every mobile host in a domain participate in this sampling process, the PACA will eventually collect enough samples to compute the transition probabilities of each cell. The identity of the host which sends a given sample is not needed by the PACA. In fact the PACA captures the aggregated movement characteristics which, in a given geographical area, are more or less common to each host.

4.3 Paging Area Composition

The collected samples are processed by the PACA in order to extract the transition probabilities in each cell in its domain. The PACA then uses the procedure **compose**(x) shown in Figure 7 for composing a paging area relative to a given cell x denoted PA_x. The composition procedure begins with an empty PA. This is necessary for adapting to the most up-to-date transition probabilities. The first cell which is added to the PA_x is x itself (lines 0-1). $P_x(i)$ is the probability of visiting the cell i without leaving PA_x (we always have $P_x(x) = 1$). Next, the algorithm finds the neighboring cells of the PA using the transition probabilities and adds the most probable one to the PA (lines 4-7). The

[3]If the Tracking Agent and the PACA functional modules are implemented on a same physical entity, samples can be sent along with registration messages.

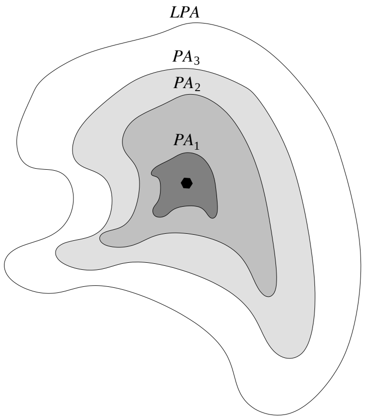


Figure 6: Paging areas of different sizes and relative to a same cell.

Procedure: **compose**(x)
00 $PA \leftarrow x$
01 $P_x(x) \leftarrow 1$
02 $N \leftarrow \{1\}$
03 **while** $|PA| < S_{MAX}$ *and* $N \neq \emptyset$
04 $N \leftarrow \emptyset$
05 **for each** $i, j \mid i \in PA, j \notin PA, P(i \rightarrow j) > 0$
06 $P_x(j) \leftarrow P_x(j) + P_x(i) \times P(i \rightarrow j)$
07 $N \leftarrow$ *all* $k \mid k \notin PA$ *and* $P_x(k) > 0$
08 $PA \leftarrow PA \cup k \mid k \in N$ *and has the largest* P_x
09 return(PA)

Figure 7: A procedure for composing a paging area relative to the cell x.

set N holds most recently discovered neighboring cells of the PA. An important point to note is that the algorithm adds one neighboring cell at a time (line 8). This is necessary for discovering more probable cells among new neighbors (if any). The procedure continues until an upper limit on the number of cells S_{MAX} is achieved (line 3). The obtained paging area is a list of cells arranged in the decreasing order of $P_x(i)$. Note that if S_{MAX} is large, and if the same cell order is preserved, it is possible to obtain a paging area of an arbitrary size S relative to a same cell (by selecting the S first cells of the list). We define the largest paging area relative to a given cell as LPA. Then, given that the PACA has configured the LPA relative to a each cell, mobile hosts can obtain paging areas of individually customized sizes. Note that, whatever the chosen size S, the obtained paging area will be the subset of its LPA and will also have an optimum shape. Figure 6 illustrates three such paging areas PA_1, PA_2 and PA_3 relative to a same cell and of different sizes, where $PA_1 \subset PA_2 \subset PA_3 \subset LPA$.

The proposed paging area configuration scheme is scalable and low-cost. In fact, each PACA is responsible for the paging area configuration of the cells of its own domain. As a result, the paging area configuration effort is efficiently distributed. Furthermore the configuration algorithm is low-cost and the sampling process minimizes the mobile hosts' involvement.

4.4 Memory Optimization

In order avoid excessive memory consumption in PACAs, we make use of the paging area coding scheme described in Section 3.3.4 by defining layered paging areas as previously illustrated in Figure 6. Each layer comprises a number of cells encoded with the paging area coding scheme. For example, given a LPA of size $S_{MAX} = 50$; we define five paging areas PA_1, PA_2, ..., PA_5 ($PA_5 = LPA$) relative to a same cell and having the sizes 10, 20, 30, 40 and 50 (respectively). Then, each layer can be represented by 128 bits instead of 128×10 bits. As a result, a memory gain up to 10 is possible (if the LPA falls in a single cluster). This scheme however, reduces the the granularity of paging area sizes.

4.5 Performance Analysis

This section presents a performance analysis of the auto-configuration algorithm presented in Section 4 and evaluates the convergence properties of this algorithm (i.e. how fast does the algorithm converge to the optimal paging areas) and the registration gain compared to Mobile IP (i.e. the number of registrations saved by using adaptive per-host paging areas). We mesure the efficiency of different paging areas by comparing their *utilization rates*. This new measure is defined in the following section.

4.5.1 Utilization Rate of a Paging Area

In this section, we develop a measure for paging area efficiency. We consider the following example illustrated in Figure 2: A particular mobile host arrives at cell c_x and requests its current Tracking Agent a paging area relative to that cell denoted PA_x. PA_x has a particular shape. For the moment, we assume that the paging area size S is constant ($S = 18$ in Figure 2). We denote the set of different cells that the mobile host visits before leaving PA_x as $\leadsto_x$, and call it a *track*. We denote the i^{th} track of M in PA_x as $\leadsto_x^i$. Then, for a given track $\leadsto_x^i$ of the mobile host we define

$$u_x^i = \frac{|\leadsto_x^i|}{S} \tag{1}$$

and say that the shape of the paging area PA_x is not efficient when u_x is small. Figures 2a and 2b illustrate two different paging area shapes which give $u_x^1 = 9/18$ and $u_x^2 = 13/18$, respectively. Clearly the paging area shape shown in Figure 2b, is more efficient since upon an incoming call, the mobile host will be paged in a larger number of cells that it has visited. Furthermore, in this figure more registrations are avoided by adapting the paging area shape to the mobile host's track. If we generalize the above formulation, we can define the overall shape efficiency of PA_x as

$$\rho_x = \lim_{L \to \infty} \frac{1}{L} \sum_{i=1}^{L} u_x^i \tag{2}$$

where ρ_x is what we call the *utilization rate* of the paging area PA_x. L is the number of times that the mobile host visits and leaves PA_x. The utilization rate is bounded by

$$\frac{1}{S} \leq \rho_x \leq 1 \tag{3}$$

The lower bound is not zero because every track $\leadsto_x^i$ of the mobile host contains at least one cell which is c_x. On the other hand the upper bound can be achieved if and only if we have the probability $P\ (|\ \leadsto_x\ | = S) = 1$. Then, given that cells are numbered as $c_0, c_1, c_2, ...$, and if we define $P_x(c_i) = P(c_i \in \leadsto_x)$ it is easily shown that the utilization rate of PA_x is

$$\rho_x = \frac{\sum_{c_i \in PA_x} P_x(c_i)}{S} \tag{4}$$

where $P_x(c_i)$ is the probability that the mobile host will visit the cell c_i before leaving PA_x.

Note that the numerator of the utilization rate is the average number of different cells visited before leaving a paging area. Therefore we can reasonably define

$$\rho = \frac{S_{eff}}{S} \tag{5}$$

where S_{eff} is the effective paging area size. S_{eff} can be used for comparing the efficiency of two paging area shapes proposed in a same set of conditions: same paging area size, same cell structures and same mobility pattern.

If we define RG_x as the average number of movements of the mobile host in PA_x i.e., the average registration gain of M in PA_x compared to Mobile IP, then we have

$$RG \geq S_{eff} - 1 \tag{6}$$

In other words, S_{eff} is the lower bound on registration gain. As a result, the utilization rate is a reasonable measure of location area shape efficiency because if the utilization rate of a paging area is large, then the average rate of registrations that the mobile host sends upon leaving that paging area will be small (RG will be high). We will use the utilization rate concept for the convergence analysis of paging areas. By "convergence" we mean the evolution of the utilization rate in time.

4.5.2 Methodology

We define two concentric zones as illustrated in Figure 8. The sampling zone (SZ) is a small portion of the cellular network where we can assume a fixed mobility pattern. We evaluate the performance of the $LPAs$ relative to the cells which are in the test zone (TZ). Note that a LPA relative to a cell in the TZ is not necessarily a subset of the TZ (hence the need for two concentric zones). The size of the SZ is chosen large enough so that it can cover all possible $LPAs$. We set $S_{MAX} = 50$ for LPA sizes. We compute the utilization rates of the $LPAs$ relative to each cell in the TZ (initially 1/50) using the Formula 2 with $L = 1000$ and $S = S_{MAX} = 50$. The overall performance is the average of the utilization rates. For simulation simplicity we use square shaped and equal sized cells. We analyze four different mobility patterns:

-Random Walk Areas (RW): In each cell of the mobility zone, there are 4 possible directions (transitions) that a mobile host may take. These are North(N), South(S), West(W) and East(E). The transition probabilities are equal and uniformly distributed over the sampling zone.

-Crossroads (4/2D): Cells are layed out according to the Manhattan grid model. We are interested by the LPA convergence at crossroads where a mobile host may take one of 4 directions with the probabilities described above. Mobile hosts change their directions only at crossroads. Therefore there are only 2 possible transitions (N-S or W-E) in the cells connecting them, (hence the name 4/2D). Each crossroads is 10 cells away from the next one.

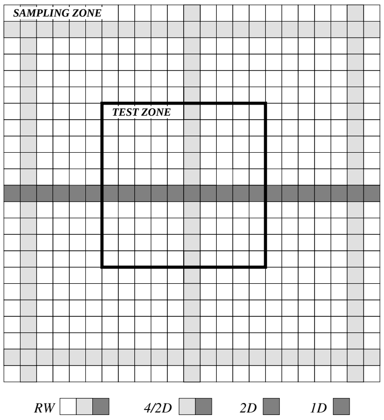


Figure 8: Mobility patterns and corresponding cells.

-Two-way Highways (2D): Cells are layed out linearly. In each cell there are 2 and equally probable directions (W-E). A mobile host does not change its direction in the sampling zone.

-One-way Highways (1D): The cell structure is the same as 2D. However, there are only one possible direction (W) in each cell.

Figure 8 illustrates the cells that may be visited with each pattern. Both the SZ and TZ are square shaped. The size of the TZ is 10×10 cells. The size of the SZ is large enough to cover all $LPAs$. Each pattern is analyzed as a separate case study. In practice, host mobility is the combination of these patterns (and many others that we are not able to model). However, our goal is to analyze the convergence of $LPAs$ in separate portions of the mobility area where we can reasonably assume a single, fixed pattern. We generate a random pair of adjacent cells $[c_i, c_{i+1}]$ in the SZ (in accordance with the mobility pattern) and count the frequency of the corresponding direction in cell c_i in order to compute the probability of moving to cell the c_{i+1}. This procedure simulates a single sample sent along with a registration message of a mobile host. We continue until the $LPAs$ converge.

4.5.3 Results and Discussion

Figure 9 shows the convergence of $LPAs$ as a function of number of samples. Each sample is processed in order to update the transition probabilities in the indicated cell and the $LPAs$ in the TZ. As the transition probabilities become more accurate, the utilization rates of the $LPAs$ increase. After the reception of a certain number of samples (which depends on the mobility pattern), the $LPAs$ shapes remain constant. At that point further sampling is not necessary. All $LPAs$ converge together since the same transition probabilities are used for the convergence of more than $LPAs$. However, in practice host flow rates may not be uniform and some of the $LPAs$ may converge faster than others.

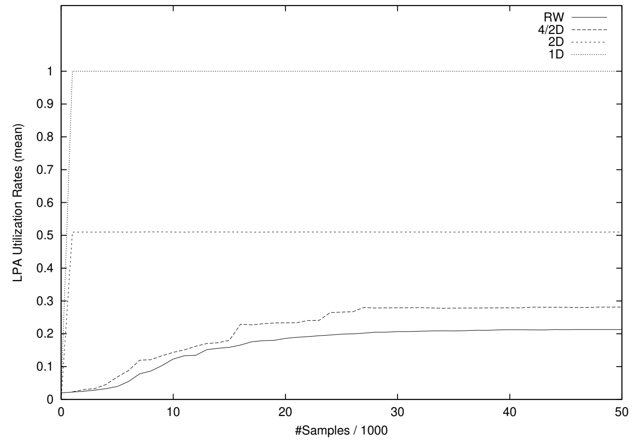


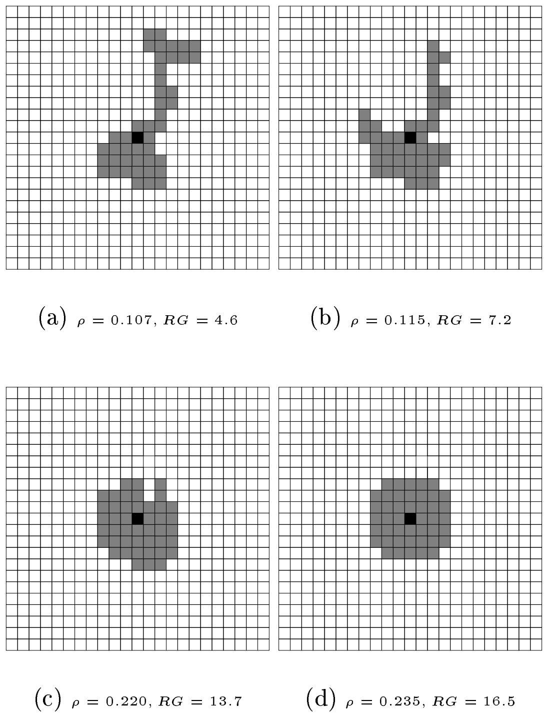
Figure 9: Mean of the LPA utilization rates as a function of number of samples.

The maximum utilization rate that can be obtained (upon convergence) does not depend on algorithmic performance but on the mobility pattern predictivity. A mobility pattern that is very predictible will tend to generate higher utilization rates. Indeed, full utilization rate can be obtained only with the 1D pattern because there only one direction is possible. In this case, $P\ (\mid \rightsquigarrow_x \mid = S) = 1$. The RW pattern is interesting because it is the most difficult to predict (in each cell of the SZ, there are four equally probable directions). This pattern obviously needs more samples than the others in order to converge. The SZ size for this pattern is $50 \times 50 = 2500$ (large enough to cover all $LPAs$in the TZ). Simulation results show that in this "worst" case , a PACA will need 40,000/2,500=16 samples per cell in order to have all $LPAs$ in its domain converged.

Finally, we analyze the effect of the paging area shape on RG (registration gain compared to Mobile IP). Figures 10 and 11 show different LPA shapes and their resulting utilization rates and registration gains compared to Mobile IP for two different mobility patterns. Figure 10 uses a RW pattern. Figure 11 uses a modified RW pattern (the probabilities of the directions N, S, E, W are 0.1, 0.1, 0.4 and 0.4 respectively). These figures show that the paging area shape has a direct impact on the generated signaling load. In fact, as the paging areas converge to their optimal shape, RG, the registration gain compared to Mobile IP, gets larger. Once the paging areas converge, the achieved gains are respectively 16.5 and 15.6. This means that a dormant mode will generate 16.5 and 15.6 time less signaling than a host that uses Mobile IP (and that therefore sends a registration each time it moves from one cell to an other). The achieved gains are smaller when the paging area shape are not choosen properly.

5. RELATED WORK

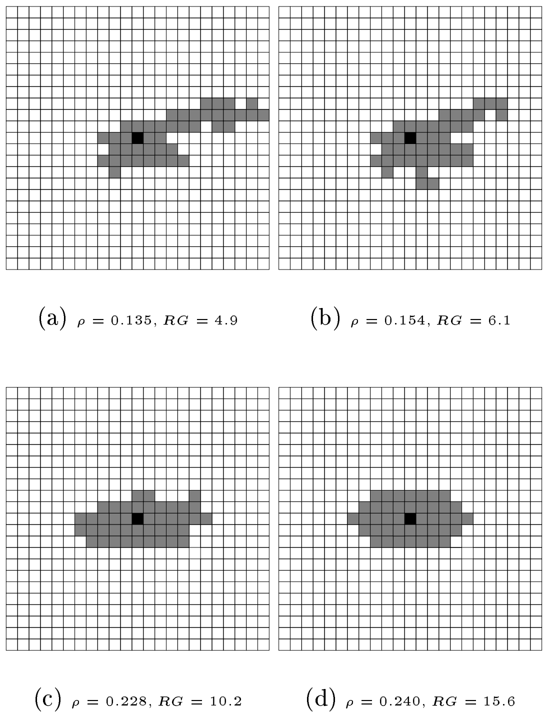
HAWAII [11] and Cellular IP [2] are two micro-mobility protocols that use a paging scheme. [10] and [2] present mechanisms to implement paging but do not describe how to configure location areas optimally. In contrast, we define an algorithm that finds the optimal location area of each mobile host. We believe these schemes can benefit from our



(a) $\rho = 0.107$, $RG = 4.6$ (b) $\rho = 0.115$, $RG = 7.2$

(c) $\rho = 0.220$, $RG = 13.7$ (d) $\rho = 0.235$, $RG = 16.5$

Figure 10: RW pattern.



(a) $\rho = 0.135$, $RG = 4.9$ (b) $\rho = 0.154$, $RG = 6.1$

(c) $\rho = 0.228$, $RG = 10.2$ (d) $\rho = 0.240$, $RG = 15.6$

Figure 11: Modified RW pattern.

adaptive per-host paging area extension.

In [12], the authors analyze three different paging architectures where Paging Agent function is implemented on Home Agent, domain root router and last contacted Foreign Agent. The major contribution of this work is a performance comparison of these schemes in terms of paging load that can be supported and reliability. The authors also analyze the impact of varying paging area size, however the same paging area sizes are assumed for all hosts. We believe that one cannot reap the complete benefit of paging unless paging area sizes are individually customized by each host. Secondly, the paging area configuration and the paging area shape performance problems are not addressed.

In [13], the authors propose paging extensions to Mobile IP called P-MIP. The authors define the *data-session* concept. Data-sessions are analogous to connections in connection oriented telephone networks. A dormant mode host is paged only on the first packet of a data-session. Secondly, the dormant mode concept is redefined for a datagram network: a mobile host that has not transmitted data for a system-specific *active-state-timer*. A host that is in active state, makes a dormant mode registration when its active-state-timer expires. The choice of the timer value is implementation dependent. One has take into account several features of TCP/IP protocols such as packet retransmissions triggered by the transport layer, short-lived web connections, etc. [13] also presents a performance evaluation of paging. However, important parameters such mobile node speed and data-session rate are assumed to be the same for all hosts. As a result, the paging area sizes are aggregated. We argue that, these are fully personal parameters and the performance of paging cannot be improved unless they are individually defined. Secondly, paging areas are manually configured. This is difficult (especially with overlapping paging areas) and not necessarily well adapted to the host mobility patterns. We believe that that P-MIP can benefit from the adaptive per-host IP Paging architecture proposed in this paper.

6. CONCLUSIONS

The contribution of this paper is twofold. First, we propose an extension to the IETF IP Paging architecture to support *adaptive per-host* paging areas. Current IP Paging proposal use static and manually configured paging areas. We, instead, propose an architecture that assigns to each host an adaptive per-host paging area. In our proposal, a host computes its optimal paging area size and the optimal shape is derived by the network. Second, we present a paging area auto-configuration algorithm. This algorithm derives optimal paging areas using the transition probabilities of the cells in its domain. Simulations show that as the algorithm converges (i.e. as the paging area shape becomes optimal), the number of registrations gets smaller and therefore the registration gain increases.

Our final objective is to develop an adaptive per-host IP Paging protocol. The work presented in the paper contributes to this goal by proposing a new IP Paging architecture and a paging area auto-configuration algorithm. The implementation issues still need to be considered. This will be part of our future work.

7. REFERENCES

[1] R. Boivie and N. Feldman. Small Group Multicast. Internet draft, draft-boivie-sgm-02.txt, work in progress, February 2001.

[2] A. Campbell, J. Gomez, S. Kim, Z. Turanyi, C.-Y. Wan, and A. Valko. Design, Implementation, and Evaluation of Cellular IP. *IEEE Personal Communications*, August 2000.

[3] C. Castelluccia. Extending Mobile IP with Adaptive Individual Paging: A Performance Analysis. *ACM Mobile Computing and Communication Review (MC2R)*, April 2001. Available at http://www.inrialpes.fr/planete/people/ccastel/.

[4] Internet Engineering Task Force (IETF). The Seamless Mobility Charter. Available at http://www.ietf.org/html.charters/seamoby-charter.html.

[5] D. Johnson and C. Perkins. Mobility Support in IPv6. Internet draft, draft-ietf-mobileip-ipv6-14.txt, work in progress, July 2001.

[6] J. Kempf. Dormant Mode Host Alerting (IP Paging) Problem Statement. RFC 3132, June 2001.

[7] J. Kempf, C. Castelluccia, P. Mutaf, N. Nakajima, Y. Ohba, R. Ramjee, X. Saifullah, B. Sarikaya, and X. Xu. Requirements and Functional Architecture for an IP Host Alerting Protocol. RFC 3154, August 2001.

[8] M. Mouly and M. Pautet. *The GSM System for Mobile Communication.* ISBN: 2-9507190-0-7, 1992.

[9] C. Perkins. IP Mobility Support. RFC 2002, October 1996.

[10] R. Ramjee, T. La Porta, and L. Li. Paging Support for IP Micro-mobility Using HAWAII. Internet draft, draft-ietf-mobileip-paging-hawaii-00.txt, work in progress, 1999. Available at http://www.bell-labs.com/user/ramjee/.

[11] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan, and L. Salgarelli. IP Micro-mobility Support Using HAWAII. Internet draft, draft-ietf-mobileip-hawaii-00.txt, work in progress, 1999. Available at http://www.bell-labs.com/user/ramjee/.

[12] R. Ramjee, L. Li, and T. La Porta. IP Paging Service for Mobile Hosts. In *Proceedings of MOBICOM'2001*, Rome, Italy, July 2001.

[13] X. Zhang, J. Gomez, and A. Campbel. P-MIP: Paging Extensions for Mobile IP. *ACM Journal on Mobile Networks and Applications (MONET)*, 2001. to be published.