An Intelligent Paging Strategy using Rule-Based AI Technique for Locating Mobile Terminals in Cellular Wireless Networks

Madhubanti Maitra¹, Debashis Saha², Partha S. Bhattacharjee³, and Amitava Mukherjee⁴

¹Department of Electrical Engineering, Jadavpur University, Calcutta 700032, India
m_madhubanti@hotmail.com

²MIS and Computer Science Group
Indian Institute of Management (IIM)-Calcutta, Calcutta 700 104, India
ds@iimcal.ac.in

³BSNL, Calcutta Telephones, Calcutta 700001, India
vigcaltel@rediffmail.com

⁴IBM Global Services India, Salt Lake, Calcutta 700 091, India
Presently at Royal Institute of Technology, Stockholm, Sweden
amitava.mukherjee@in.ibm.com
(Author for Correspondence)

Abstract

This work proposes an efficient rule-based paging strategy (RBPS) using a well-known concept of artificial intelligence, namely rule base. The novelty of the scheme lies in devising “rules” that offer a potential mapping from seemingly disparate input data items (yet having some statistical relations) to an almost exact position of mobile terminals (MTs). Considering the conventional models of call arrival, cell residence and mobility, we have developed a stochastic model to analyze the performance of the scheme. Interestingly, RBPS requires no additional processing at MTs, and involves a nominal overhead at mobile switching centers. Simulation results reveal that RBPS significantly outperforms the GSM-adopted blanket paging scheme. Also, results are very much encouraging, when compared with the popular shortest-distance-first [3] scheme. Finally, RBPS is generic enough to be potentially used in the next generation wireless networks too, irrespective of any standards, only with minor adaptations to conform to the respective standards.

Keywords: Paging, Artificial Intelligence (AI), Rule-based Systems, Paging Area (PA), Rule-Based Paging Strategy (RBPS), Shortest-Distance-First (SDF) Scheme.
List of Notations Used

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_m )</td>
<td>Cell residence time of a mobile terminal (MT)</td>
</tr>
<tr>
<td>( f_m(t) )</td>
<td>Probability density function of the cell residence time</td>
</tr>
<tr>
<td>( F_m(s) )</td>
<td>Laplace-Stieltjes transform of ( f_m(t) )</td>
</tr>
<tr>
<td>( \hat{\lambda}_m )</td>
<td>Reciprocal of the mean of the probability density function of the cell residence time</td>
</tr>
<tr>
<td>( t_c )</td>
<td>Inter-call arrival time</td>
</tr>
<tr>
<td>( \zeta_{Ev} )</td>
<td>the probability that ( \chi_{tm} \leq t_c &lt; \psi_{tm} )</td>
</tr>
<tr>
<td>( \lambda_m )</td>
<td>Rate of the Poisson call arrival process for an MT</td>
</tr>
<tr>
<td>( L )</td>
<td>Index set of location areas (LAs) that comprises the total coverage area</td>
</tr>
<tr>
<td>( nr_{LA} )</td>
<td>Number of rings of cells that comprise an LA</td>
</tr>
<tr>
<td>( nr_{LA}, nr_{PA} )</td>
<td>Number of rings of cells that comprise the ( i )th LA, ( l \leq i \leq L )</td>
</tr>
<tr>
<td>( Nc_{r_i} )</td>
<td>Number of cells presents in ring ( i ) in a hexagonal cell topology, ( i=1,2,3... )</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Shaping factor of the Gamma Distribution</td>
</tr>
<tr>
<td>( \ell )</td>
<td>Number of paging areas (PAs) in an LA is partitioned into</td>
</tr>
<tr>
<td>( \varphi )</td>
<td>Maximum allowable delay for searching an MT</td>
</tr>
<tr>
<td>( g )</td>
<td>Minimum number of rings of cells per PA</td>
</tr>
<tr>
<td>( \alpha(K) )</td>
<td>Probability of ( K ) cell boundary crossings between two call arrivals by an MT</td>
</tr>
<tr>
<td>( pl_{ij}[K] )</td>
<td>Probability that an MT moves from LA(_i) to LA(_j), given that ( K ) cell boundary crossings are performed in the process, ( l \leq i \leq L, l \leq j \leq L )</td>
</tr>
<tr>
<td>( pa_{ij}[K] )</td>
<td>Probability that an MT moves from PA(_i) to PA(_j), given that ( K ) cell boundary crossings are performed in the process, ( 0 \leq i, j \leq \ell )</td>
</tr>
<tr>
<td>( \theta )</td>
<td>Call-to-mobility ratio (CMR)</td>
</tr>
<tr>
<td>( \text{Var} )</td>
<td>Variance of the Gamma distribution</td>
</tr>
<tr>
<td>( s )</td>
<td>Complex Frequency</td>
</tr>
<tr>
<td>( P_{LA} )</td>
<td>LA-transition probability matrix; dimension ( L \times L )</td>
</tr>
<tr>
<td>( PA_{tran} )</td>
<td>PA transition matrix; dimension ( \ell \times \ell )</td>
</tr>
<tr>
<td>( pr_{PAC}, pr_{PA} )</td>
<td>Probability that the ( i )th PA is the current PA, where the last call was delivered successfully to an MT, ( 0 \leq i \leq \ell - 1 )</td>
</tr>
<tr>
<td>( PA_c )</td>
<td>Current PA, where the last call was delivered successfully to an MT</td>
</tr>
<tr>
<td>( NC_{LA} )</td>
<td>Total number of cells present in an LA; the subscript ( T ) indicates total</td>
</tr>
<tr>
<td>( NC_{LA_1} )</td>
<td>Total number of cells present in the ( i )th LA or LA(_i), ( 1 \leq i \leq L )</td>
</tr>
<tr>
<td>( C )</td>
<td>Three-tuple structure of the rule base</td>
</tr>
<tr>
<td>( S(\delta) )</td>
<td>Set of system states with associated probability ( \delta ); the set of all possible PA(_c)’s</td>
</tr>
<tr>
<td>( T(\zeta) )</td>
<td>A set of transition rules with associated probability ( \zeta )</td>
</tr>
<tr>
<td>( A )</td>
<td>A set of actions to be taken when a transition rule ( T ) fires; Actions are the set of all possible combinations of PAs that is to be polled to search the MT</td>
</tr>
<tr>
<td>( pr_{TC}, pr_{TC_j} )</td>
<td>Probability of any transition rule ( T_j ) being fired, ( l \leq j \leq \ell )</td>
</tr>
<tr>
<td>( \rho_{ij} )</td>
<td>Probability for an MT to move from ( i )th PA to ( x )th PA when rule ( j ) fires, ( 0 \leq i, x \leq \ell - 1, l \leq j \leq \ell )</td>
</tr>
<tr>
<td>( nc_{PA_x} )</td>
<td>Number of cells in the ( x )th PA, ( 0 \leq x \leq \ell - 1 )</td>
</tr>
<tr>
<td>( nr_{PA_l} )</td>
<td>Number of rings in the ( l )th PA, ( 0 \leq l \leq \ell - 1 )</td>
</tr>
<tr>
<td>( nc_{PA_{x,j}} )</td>
<td>Number of cells to be polled in the ( x )th PA when rule ( j ) fires, ( 0 \leq x \leq \ell - 1, l \leq j \leq \ell )</td>
</tr>
<tr>
<td>( C_v )</td>
<td>Expected paging cost per call arrival</td>
</tr>
<tr>
<td>( V )</td>
<td>Unit cost for paging per call</td>
</tr>
<tr>
<td>( C_u )</td>
<td>Expected location update (LU) cost</td>
</tr>
<tr>
<td>( U )</td>
<td>Unit cost for an LU</td>
</tr>
<tr>
<td>( C_T )</td>
<td>Expected total cost for LU and terminal paging per call arrival</td>
</tr>
</tbody>
</table>
### I. Introduction

#### A. Background

In mobile cellular networks, total coverage area is usually partitioned into several location areas (LAs), which again are often partitioned into paging areas (PAs) to expedite the paging process [1]-[3] in general. The location information of a mobile terminal (MT), since its last location update (LU), is somewhat uncertain as its exact point of attachment with the network at any point of time is completely dictated by its mobility [2]-[11]. When a call arrives for an MT, naturally the network performs several search iterations (i.e., paging cycles) to locate the MT in order to deliver the call, where each iteration involves a PA (or a set of PAs, or even an LA sometimes) to be polled. However, there is a temporal constraint in terms of allowable paging latency for the complete process. This timeout period, therefore, determines the maximum number of polling cycles allowed for the search. Consequently, paging cost depends upon the number of polling cycles as well as the number of cells to be polled in each cycle [12].

#### B. Previous Works

An interesting research problem in location management for MTs is to resolve the trade-off between the LU frequency and paging latency [1]-[3],[12],[17]. Several paging schemes have been proposed in the literature for reducing the overall signaling load generated in the system [5]-[10],[13]-[16],[18]-[20], while maintaining the latency within a given bound. Most of the current systems (typically GSM), however, employs the blanket paging [1]-[3], in which all the cells under the coverage of a mobile switch-
ing center (MSC) (i.e., all the cells within an LA in some cases) are polled simultaneously in one cycle. Though the latency is minimal, a major drawback of this scheme is the significant amount of signaling (and hence, cost) involved for large LAs. Alternatively, a two-step paging process has been suggested in [5], where a selective portion of the LA (i.e., a set of PAs) is polled in the first cycle, and, iff the MT is not found in that portion, blanket paging is applied in the second (and last) cycle. In [14], for Universal Mobile Telecommunication Systems (UMTS), a similar scheme has been proposed, in which, depending upon the latest interaction information of the MT with the network, only a portion (for instance one PA) of the LA is polled first. In case, the MT is not found in that PA, the complementary portion of the LA is polled next. In Shortest-Distance-First (SDF) schemes, a network starts polling an MT, from the cell where the MT performed its last LU. The network subsequently polls the cells in the shortest-distance-first order, assuming that the MT is moving outward from the last known location, till it is found. The distance is measured in terms of the number of cells crossed by the MT since the last LU, which is governed by the LU scheme employed (for e.g., Distance based, Time based or Movement based [3]). Thus, in a sense, the distance identifies the cells to be polled at one go, which collectively constitute a PA. The delay constraint can be further imposed, by grouping the cells with increasing distances, for each polling cycle. For instance, we know that, in movement based LU schemes [3], an update is performed when the number of movements since the last LU equals a pre-defined value $d$. When SDF scheme is applied, all the cells within the distance $d$ from the last registered location (i.e., center cell) are polled in either a single step or multiple steps (in a sequential manner), depending upon the maximum allowable delay. In each step, the network selects a PA (beginning with the center cell) for polling. The number of PAs equals the number of maximum polling cycles allowed. Among other notable schemes, an optimal sequential paging scheme for highly mobile users has been proposed in [8], while a load adaptive threshold strategy has been proposed in [9]. In [24], a location management strategy with selective paging has been proposed. Tung et al. [25] have proposed a new sectional paging strategy, which reduces the paging cost, satisfying the delay constraint for the MTs with traceable roaming patterns. In [18], another sequential
paging scheme had been proposed as an extension of [11]. Obviously, the time-complexity of most of the above schemes is proportional to the square of the number of cells in an LA.

In summary, we can infer that, in order to improve the bandwidth utilization and to reduce the cost subsequently, several multi-step paging strategies under delay bounds have been proposed in the literature, where PAs are polled sequentially starting with the PA where the probability of locating the MT is higher. A major drawback of these schemes is that they result in an extra processing overhead at the MSC end (for storing and retrieving the recent interaction information of MTs), and may also amount to extra paging delays if MTs are found in the last step. This is evident from the findings that a uniform location distribution results in highest paging cost and delay [13]. To get around the problem, some researchers have proposed timer-based [6] and state-based [7] paging/registration (i.e., combined) strategies, which go best only with their specific LU techniques (but not with the general ones).

To alleviate the afore-mentioned difficulties with the existing paging techniques, our proposed rule-based paging strategy (RBPS), first presented in [19], aims to predict the location of an MT within a focused region (thereby reducing the frequency of redundant paging). We have exploited the concept of recent interactions of the MT with the network towards developing a rule-based approach for heuristic search of MTs in the network. The idea of using rules for translating disparate and stochastic data items, which certainly have a definite relation to MT’s location, is a novel contribution of this work. In its core, RBPS uses a search engine, which predicts the location of an MT almost precisely, when the next call arrives for it (an event that triggers the corresponding rule).

C. Our Contribution

To the best of our knowledge, no previous work on paging employed a “rule-base” approach while the chain of processes involved in paging, according to our opinion, presents a solid case for decision-making based on rules, representing system knowledge up to the micro-level [4],[17], [21]-[23]. In rule-base, a rational application of event-driven artificial (AI) technique can almost certainly find an MT without a miss within the delay bound. Therefore, contrary to many other paging schemes, RBPS needs to
poll less cells because it is capable enough to decide on a probable PA, thereby reducing both cost and latency. In addition to user profiles available in location register (LR) databases, only one more look-up table (termed as *Decision Table* in this work) has to be referred in this case. An RBPS engine does the rest of the searching process. This entails in very little processing overhead at the MSC end. Since RBPS requires that MTs need not perform LU within an LA, any extra processing overhead may be avoided altogether at the MT end. So, implementation of RBPS will require a minor alteration in existing systems. Moreover, RBPS does not depend on the mobility pattern of MTs, whereas any variant of sequential paging requires information about the MT’s movement history, the formalism of which must be consistent with the LU scheme chosen along with. Consequently, in sequential paging, movement profile of an MT has to be generated in a format as appropriate to the associated LU scheme. Further, most other paging schemes partition one LA into several PAs and then only perform terminal paging (known as Intra-LA paging). RBPS considers both Intra- and Inter-LA paging, and, accordingly, computes LA transition probabilities too (it will be clear in future discussions). Since the construction of rules does not depend on LU information, RBPS is ideally independent of the LU method used in conjunction. However, in this work, we will show it to work with movement based LU scheme as a case study. Depicting the microscopic mobility pattern of MTs, movement-based LU is popular and well-cited, and is sometimes considered as a de-facto benchmark in the literature. We plan to combine RBPS with other LU schemes in our future works.

To assess the performance of RBPS, we have compared its performance with two well-known schemes, namely SDF paging [3] and blanket paging [1]. While calculating the total cost, we have considered the LU cost also so that we can assess the implication of RBPS on the total signaling cost. Though we have assumed hexagonal cells with random-walk mobility model [12]-[15] for the sake of simplicity in analysis, RBPS is not tied to these assumptions. In fact, RBPS can be used with any cell structure and any mobility model. However, since the premise of the work done in [11] and [18] is completely different from our premise, we are not in a position to compare our work with either [11] or [18].
D. Organization of the paper

The rest of the paper is organized as follows. In Section II, we describe the network topology along with the assumptions that we have considered. The concept of rule based system and the proposed RBPS have been discussed in Section III. Analytical model and the cost formulations are presented in section IV. Performance analysis of the model and its comparative study are presented in section V. Section VI concludes the work.

II. System Description

We assume that cells have unique cell identification numbers (CINs). Each LA is represented by the CINs of its cells. To indicate the boundary of an LA, we use a boundary cell identification number (BCIN) vector as defined next. The BCIN vector of LA\(_i\) contains CINs of the boundary cells of LA\(_i\), and the CINs of the boundary cells of the neighboring LAs of LA\(_i\) along with the indices of the respective adjacent LAs. Since the LA topology is assumed to be completely known, this information can be derived from the LR databases in each LA. So we assume that an LR database stores the CINs of boundary cells of the LA along with the information that which boundary cell falls in which neighboring LA. Specifically the BCIN vector helps determine the LA boundary crossing by MTs. For example, let us refer to Figure 1. The coverage area of the toy network has been partitioned into two LAs, namely LA\(_i\) and LA\(_j\), where i, j \(\in [1, L]\). Hence, a 4-tuple notation can represent the BCIN vector in this case (the exact data structure is provided in section IIIA.1). This information is updated only when there is any change in LA dimensions.

A. Assumptions

As introduced already, we assume that cells are regular hexagons in shape, and MTs execute random walk, moving radially outward from the centre of cells. An MT performs an LU only when it crosses the boundary of an LA \([2],[3]\). \textbf{Cell residence time} \(t_m\) is assumed to be an independent and identically distributed (iid) random variable, which follows Gamma distribution with mean \(1/\lambda_m\). Laplace-Stieltjes transform \(F_m^*(s)\) and shaping factor \(\gamma\). Call arrival to an MT is a Poisson process with
mean $\lambda$. Hence, inter-call arrival time $t_c$, also an iid random variable, is exponentially distributed. The location co-ordinate of an MT or in a broader sense, the PA, where the last call was delivered, is known to the system, as the call was connected successfully. Hence, the network exactly knows the PA, where the last call was delivered.

**B. LA Topology**

The $i^{th}$ LA topology consists of $nr_{LA_i}$ number of rings of cells with center cell (which is the center of mass of the LA) representing the innermost or $0^{th}$ ring (R0), six neighboring hexagonal cells around the center cell representing $1^{st}$ ring (R1), and so on. For a given center cell, we denote $r_i (i \geq 0)$ to be the set of cells in the $i^{th}$ ring. We measure all the distances in terms of the number of rings from the center cell such that the distance of any of the cells belonging to $r_i$ is $i$ rings. For example, the distance of each cell in ring 3 (R3 in Figure 1) from the center cell (R0) is 3 rings.

![LA topology network](image)

*Figure 1: An example two-LA topology network (arrow indicating the movement direction of an MT)*

**C. LA Partitioning Scheme**

Since we consider movement-based LU [3] to compliment RBPS, the step variable $d$ indicates that, after executing $d$ number of discrete steps, an MT performs an LU. Naturally, $d$ depends upon the LA size and the average mobility rate of MTs. We partition the $i^{th}$ LA in $\ell = \min(\varphi, nr_{LA_i})$ number of PAs, where
\( \phi \) denotes the maximum allowable delay, in such a way that all PAs consist of approximately same number of rings. Here \( i^{th} \) PA is denoted by \( PA_i \), where, \( 0 \leq i < \ell \). LA partitioning scheme is given below [16]:

**Step 1:** Minimum number of rings of cells per PA in \( i^{th} \) LA is determined as
\[
g = \left\lfloor \frac{nr_{LA_i}}{\ell} \right\rfloor \tag{1}
\]

**Step 2:** The variable \( \kappa \) is determined as
\[
\kappa = nr_{LA_i} - g^* \ell \tag{2}
\]

**Step 3:** For partitioning the LA we start with the center cell of the LA, which denotes 0th ring. The series of PAs is determined as \( PA_0, PA_1, \ldots, PA_{\ell-1} \), where \( g \) rings are allocated to each of the first \((\ell-\kappa)\) PAs and \((g+1)\) rings are assigned to each of the remaining \( \kappa \) PAs. Hence, \( PA_0 \) consists of ring 0 and next \((g-1)\) rings of cells; \( PA_1 \) consists of rings ranging from \( g \) to \( 2g-1 \) and so on. Hence, the following 2-tuple notation completely describes the structure of PAs as: \( (PA_0, \{0, \ldots, g-1\}) \), \( (PA_1, \{g, \ldots, 2g-1\}) \), \( \ldots \), \( (PA_{\ell-1}, \{(\ell-\kappa-1)g, \ldots, (\ell-\kappa)g-1\}) \), \( (PA_{\ell-1}, \{(\ell-\kappa)g, \ldots, (\ell-\kappa+1)g\}) \), \( \ldots \)

To exemplify, let us consider an LA of 16 rings of cells, and let the maximum allowed delay be 5. Then, as per the algorithm, \( \ell = \phi = 5 \); \( g = \left\lfloor \frac{nr_{LA}}{\phi} \right\rfloor = 3 \), and \( \kappa = nr_{LA} - g^* \phi = 1 \). So, the first \((\phi-\kappa)\) i.e., 4 PAs have 3 rings of cells, and the last or the fifth PA has 4 rings of cells. Hence, \( PA_0 \) contains ring 0 (i.e., the center cell), ring 1 and ring 2, \( PA_1 \) contains rings 3, 4 and 5, and so on, and the last PA or \( PA_4 \) contains rings 12, 13, 14 and 15. Once partitioning of an LA has been completed to form PAs, the BCIN vector of every PA is stored in the LR database. Hence, referring the LR database, PAs can be determined easily.

### III. Rule Based Paging Scheme (RBPS)

#### A. Basic Components of a Rule Based System

**A.1 Facts and Fact base (Methods for acquisition, representation and use of known facts)**

The fact base is a collection of information (represented in the form of a suitable data structure) containing the known facts relevant to the domain of interest. We have used “facts” to provide the persistent knowledge incorporated in the RBPS heuristic. For instance, the location of an MT when it has received the last call successfully (i.e., \( PA_c \)) is the most recent interaction of the MT with the network, and, hence, is considered as a ‘fact’. Moreover, LA topology, PA structure, CIN and BCIN are also facts as they are known beforehand. All these facts about the MT constitute its fact base. In our case, a portion of the LR databases can be treated as the fact base, whose size will depend on the size of the coverage area of the
cellular network as well as the sizes of LAs. For example, one representative data format for the LA topology of Figure 1 could be as follows:

<table>
<thead>
<tr>
<th>Fact Base Index</th>
<th>Fact Base Variable</th>
<th>Representative data format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$L$</td>
<td>Integer</td>
</tr>
<tr>
<td>2</td>
<td>nr_LA, or $n_i$</td>
<td>Integer</td>
</tr>
<tr>
<td>3</td>
<td>$N_{c,r}$</td>
<td>Integer</td>
</tr>
<tr>
<td>4</td>
<td>$\ell$</td>
<td>Integer</td>
</tr>
<tr>
<td>5</td>
<td>$\varphi$</td>
<td>Integer</td>
</tr>
<tr>
<td>6</td>
<td>$P_{A_c}$</td>
<td>Long Integer</td>
</tr>
<tr>
<td>7</td>
<td>$NC_{LA_T}$</td>
<td>Integer</td>
</tr>
<tr>
<td>8</td>
<td>$N_{r,PA_l}$, $0 \leq l \leq \ell$</td>
<td>Integer</td>
</tr>
<tr>
<td>9</td>
<td>$NC_{x_{tot}}$</td>
<td>Integer</td>
</tr>
<tr>
<td>10a</td>
<td>$P_{A_0}$</td>
<td>Integer</td>
</tr>
<tr>
<td>10b</td>
<td>Ring 0</td>
<td>Integer</td>
</tr>
<tr>
<td>10g</td>
<td>Ring $(g-1)$</td>
<td>Integer</td>
</tr>
<tr>
<td>11a</td>
<td>$P_{A_{\ell-k}}$</td>
<td>Integer</td>
</tr>
<tr>
<td>11b</td>
<td>Ring $(\ell-k)g$</td>
<td>Integer</td>
</tr>
<tr>
<td>11g</td>
<td>Ring $(\ell-k+1)g$</td>
<td>Integer</td>
</tr>
<tr>
<td>12a</td>
<td>$LA_i$, $1 \leq i \leq L$</td>
<td>Integer</td>
</tr>
<tr>
<td>12b</td>
<td>CIN of $LA_i$, $1 \leq i \leq L$</td>
<td>Array of Integers</td>
</tr>
<tr>
<td>13a</td>
<td>$LA_c$, $1 \leq c \leq L$</td>
<td>Integer</td>
</tr>
<tr>
<td>13b</td>
<td>$W$, $c \in W$</td>
<td>Array of Integers</td>
</tr>
<tr>
<td>13c</td>
<td>$NC_{LA_{cx}}, 1 \leq x \leq NC_{LA_T}$, $1 \leq c \leq L$</td>
<td>Integer</td>
</tr>
<tr>
<td>13d</td>
<td>$NC_{LA_{cx}}, w \in W and w \neq c, 1 \leq c \leq L$</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Very few variables of the fact base need to be changed frequently. For instance, when a call arrives for an MT, after delivering the call successfully, only the location coordinate of $P_{A_c}$ will have to be updated.

**A.2 Production (or Transition) Rules (Methods for conducting heuristic search)**

A production (or transition) rule is a situation-action couple, which implies that whenever a certain situation (given in the left hand side (LHS) of the rule) is encountered, the action specified on the right hand side (RHS) of the rule is performed. Often the action is making some decision. Since our goal is to find the precise location where the MT is currently residing within some pre-fixed temporal constraint, we call the current location of the MT as the goal state in the heuristic search space [4], [21]-[23].
Some legal steps help in moving forward from the initial state to the goal state following some transitions through a sequence of intermediate states in between. These legal moves can be described with the help of production rules, consisting of two parts: (i) LHS (or antecedent) part that serves as a pattern to be matched against some known facts, such as $PA_c$, mobility pattern of the MT, average inter-call arrival time for the MT etc., and (ii) RHS part that indicates some actions/decisions to be taken, such as PA/PAs to be polled to search for the MT to reach the goal state. These production rules, represented as “IF-THEN” clauses, are often called simply rules. RBPS inference engine tries to apply the rules as a consequence of the facts that it studies i.e., it works in a forward chaining mode.

![Figure 2: A Timing Diagram (t represents time)](image)

### A.3 Examples

Let us consider a toy scenario (Figure 2), where PQ, QR, and RO represent three consecutive cells. Let us assume that an MT has received its last call when it is at point A (Figure 2) i.e., when it is within the cell PQ. If it moves radially outward from the center of the cell, and if the next paging request arrives somewhere at point B, then the MT can be found in the same cell (PQ), because it is clear from the figure that, within this period of time, the MT cannot leave the cell due to its mobility pattern. Therefore, we can formulate a rule like the following one:

```
IF ($t_c < 0.5t_m$) /* i.e., the MT can be found in the same cell */

THEN the current cell is to be polled to deliver the next call
```

[This is a Fact]

[This is an Action]
Generalizing the above rule for a practical scenario, we can write:

\[ IF \ (t_c < 0.5t_m) \ THEN \ poll \ PA_c \]

Again, going back to the example of Figure 2, if the next call arrives somewhere at C, the MT will be found neither in cell PQ nor in cell QR. In fact, it is to be searched now in cell RO. Therefore, the rule for this exhaustive case can be described in a generic way as:

\[ IF \ (0.5t_m \leq t_c < \hat{U}t_m) \ THEN \ poll \ PA_c \ as \ well \ as \ its \ adjacent \ PAs. \]

The value of \( \hat{U} \) is dependent on the number of rings of cells that form the successive PAs.

Let us next consider a representative PA structure, as shown in Figure 3, to further explain the rules. If we assume that the MT received the last call when it was at the center of the cell in Ring 2 and if \( t_c \) is greater than or equal to 3.5 times \( t_m \), but less than 4.0 times \( t_m \), then the MT could move, at most, to a cell in ring 6 (the worst case when the MT is moving radially outward) which is in PA\(_2\). So, we can conclude that, under this condition, there is a possibility that MT will be either in PA\(_0\) or in PA\(_1\) or in PA\(_2\) and the possibility that the MT can go to PAs beyond PA\(_2\) is zero. Following these considerations, we can develop a general rule base in the form of a decision table (Table I). In our work, we denote PA\(_c\) to be the initial state of the whole dynamic process of paging and concentrate on the state of the MT and the event of occurrence of a call arrival. The probabilities for residing of an MT in any of the states and triggering of each and every rule, following the action to be taken, have been derived in this work.

**B. Decision Table**

A decision table is essentially a look-up table in the form of a matrix, where states are rows and LHS (or antecedent) parts of the rules are columns. The intersection of a row and a column shows the cor-
responding action. **Table I** describes our proposed decision table, where \( n_i \) is the number of rings of cells present in \( \text{PA}_i \). We use the notation \( n_i \) instead of \( nr_{-PA}_i \) that we will use latter for representing number of rings of cells in \( i^{th} \) PA only to present the decision table in a simpler way. As a data structure, decision table can be a matrix with dimension \( \ell \times \ell \). In our case, \( \ell \) is equal to 5. Hence, the dimension of the decision table is merely 5x5. Generally, even for sequential paging, due to temporal constraint, the maximum paging delay is set around 5 (it cannot be very large). Hence, the dimension of the table will be around 5x5 in most of the cases.

**Table I: Decision Table for RBPS**

<table>
<thead>
<tr>
<th>Rules</th>
<th>( t_c &lt; 0.5 t_m )</th>
<th>( 0.5 t_m \leq t_c &lt; (n_1 + 0.5) t_m )</th>
<th>( (n_1 + 0.5) t_m \leq t_c &lt; (n_1 + n_2 + 0.5) t_m )</th>
<th>( (n_1 + \ldots + n_{c-2} + 0.5) t_m \leq t_c &lt; (n_1 + \ldots + n_{c-1} + 0.5) t_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>State ( S = \text{PA}_c )</td>
<td>( \text{PA}_c )</td>
<td>( \text{PA}<em>c + \text{PA}</em>{c+1} )</td>
<td>( \text{PA}<em>c + \text{PA}</em>{c+1} + \text{PA}_{c+2} )</td>
<td>( \text{PA}<em>c + \text{PA}</em>{c+1} + \ldots + \text{PA}_\ell )</td>
</tr>
<tr>
<td>( \text{PA}_0 )</td>
<td>( \text{PA}_c )</td>
<td>( \text{PA}_{c-1} + \text{PA}_c )</td>
<td>( \text{PA}_{c-1} + \text{PA}<em>c + \text{PA}</em>{c+1} )</td>
<td>( \text{PA}_{c-1} + \text{PA}<em>c + \ldots + \text{PA}</em>\ell )</td>
</tr>
<tr>
<td>( \text{PA}_1 )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( \text{PA}_{c-1} )</td>
<td>( \text{PA}_c )</td>
<td>( \text{PA}_{c-1} + \text{PA}_c )</td>
<td>( \text{PA}_{c-1} + \text{PA}<em>c + \text{PA}</em>{c-1} + \text{PA}_c )</td>
<td>( \text{PA}<em>{c-1} + \text{PA}</em>{c-2} + \text{PA}_{c-1} + \text{PA}_c )</td>
</tr>
</tbody>
</table>

If the PA-structure remains the same, obviously the decision table need not be updated. However, if the maximum allowable paging delay \( \varphi \) changes, the PA-structure might change. In that case, only the decision table will be redesigned.

**B.1 Declarative Knowledge Base**

The dynamic behavior of the system can be described by the declarative knowledge \( \hat{C} \). Our proposed knowledge based system is a 3-tuple structure, such that \( \hat{C} = \langle S(\delta), T(\zeta), A \rangle \), where \( S(\delta) = \)
\{S_0(\delta_0), S_1(\delta_1), …, S_{\ell-1}(\delta_{\ell-1})\} is a set of states with associated probability \(\delta\) and \(\ell \geq 0\). The probability the system can be at any state \(S_i\) is \(\delta_i\).\( T(\zeta) = \{T_1(\zeta_1), T_2(\zeta_2), …, T_\beta(\zeta_\beta)\}\) is a set of rules and \(\beta \geq 0\). The probability of any rule \(T_j\) being fired is \(\zeta_j\). \(A = \{A_{01}, A_{02}, …, A_{ik}, …, A_{1-1\beta}\}\) is a set of actions to be taken when a rule \(T_k\) has been fired and the system is at some state \(\beta \geq k \geq 1\), \(\ell \geq i \geq 0\).

Rules are derived from the joint probability of occurrence of the arrival of a call within a specified time interval of cell residence time of an MT. Let us consider a member of \(\hat{C}\) as \((S_2(\delta_2), T_2(\zeta_2), A_{22})\). It implies that, if the MT be in PA\(_2\) with a probability \(\delta_2\) while receiving the previous call, and if the event \(T_2(\zeta_2)\) occurs, the action that has to be taken is \(A_{22}\). To be more explicit, we explain the members of \(\hat{C}\), using the following rule base.

**FACT:** The MT has been in PA\(_2\) with a probability \(\delta_2\), when it has received the last call: (State)

**IF:** The time elapsed since then till the next paging request arrives for the MT is greater than or equal to 0.5\(t_m\) and less than \((n_1+0.5)\ t_m\): (Antecedent)

**THEN:** The PAs that have to be polled for the MT are PA\(_1\), PA\(_2\) and PA\(_3\) (Action)

The rules form the key to RBPS, and, hence, must be derived carefully. For instance, the rules mentioned above, are formulated based on the following rationale:

1. The MT can move to either PA\(_1\) (PA\(_{c-1}\)) with some transition probability or to PA\(_3\) (PA\(_{c+1}\)) with another transition probability in this interval; else it can still remain in PA\(_2\) (PA\(_c\)) with some other transition probability (these transition probabilities can be derived from 2-D random walk state transition matrix).

2. For sequential paging, in this particular case, since PA\(_2\) is the current location, it has to be polled first. If the MT is found in PA\(_2\), search procedure stops. Else in the next polling cycle, the network will poll the other PAs (PA\(_1\) or PA\(_3\)) according to the decreasing order of probabilities of finding the MT in
these PAs. On the contrary, for blanket paging, all the cells in the PAs, viz., PA₁, PA₂ and PA₃, are polled simultaneously.

3. For the rest of the work, we assume $nr_{LA} > \phi$. Only for a very small LA, the situation could be reverse.

B.2 Inference Engine

Whenever an event occurs (for e.g., a call arrives for an MT), within a specified interval of time, depending on its mobility pattern, a rule will suggest an external event. The Inference Engine will invoke that program (the action part of the rule) and will resume execution of the appropriate rule, which will finally generate the action from the network end (i.e., firing of the rule). In this context, the external event is polling or searching the PA/PAs (the RHS of the rule found from the decision table) either using SDF or Blanket paging. For simulation testing, we have designed the Inference Engine in MATLAB version 6.

IV. Analytical Model and Cost Calculation

Derivation of the stochastic model for RBPS is based on the probability distributions of: (i) the cell-, PA- and LA-boundary crossings by an MT between two consecutive call arrivals, and (ii) the variation of $t_c$ within a given interval of $t_m$. The probability $a(K)$ of $K$ number of cell boundary crossings between two successive call arrivals has been derived using the analysis as given in [3]. Assuming that the MTs are evenly distributed within an LA, the probability $PR_{CB_{ij}}$ that an MT will make a transition from LAᵢ to LAⱼ, in a single step, is given by:

$$PR_{CB_{ij}} = \sum_{i=1}^{NC_{LAc}} \frac{PR_{CB_{ej}}}{NC_{LAi}} \quad (3)$$

Detailed mathematical derivation is given in Appendix I.

A. PA Transition Probability

The efficacy of RBPS relies upon the partitioning of LAs into PAs. Once the LA is divided into PAs, the next step is to find out the transition probability of an MT from one PA to another PA within a given pe-
period of time depending on its mobility pattern. The number of rings in each PA depends upon the total number of rings that could be identified in an LA and the PA partitioning scheme. Hence, to find out PA transition probabilities, one should find out ring-to-ring transition probabilities, which can be easily computed by simplifying 2-D random walk into 1-D random walk with one barrier state [3], as shown in Figure 4. The ring-to-ring state transition as per [3] is shown in Equation 4. The MT is said to be in state $i$ if it is currently residing in a ring-$i$ cell. After $K$ moves, the MT can, at most, move to ring-$K$ [3]. Thus, we can modify the 1-D random walk model in Figure 4 such that states zero to $K-1$ are transient states and state $K$ is an absorbing state. The $K \times K$ transition matrix for the random walk is $P_K$ (the matrix is given below, where an element $p_{i,j,K}$ is the probability that an MT moves from a ring-$i$ cell to a ring-$j$ cell in one step [3]).

\[
P_K = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
1/6 & 1/3 & 1/2 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 1/4 & 1/3 & 5/12 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 5/18 & 1/3 & 7/18 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & 5/18 & 1/3 & 7/18 & 0 & \ldots & 0 & 0 \\
0 & 0 & 0 & 0 & 5/18 & 1/3 & 7/18 & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & 5/18 & 1/3 & 7/18 & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & \frac{2(K-1)}{6(K-1)} & \frac{1}{1} & \frac{2(K-1)+1}{6(K-1)} \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
\end{bmatrix}
\]

\textbf{Figure 4: State Diagram for hexagonal random walk model in 1-D plan}
\[
P_k^{(a)} = \begin{cases} 
P_k & a = 1 \\
P_k \times P_k^{(a-1)} & a > 1 \end{cases}
\]  

Let us now explore how we can obtain the PA transition probability from the above-mentioned ring-to-ring transition probability. The probability for an MT to move from PA\(_i\) to PA\(_j\) depends on where it exactly resides in PA\(_i\). It is obvious that an MT, residing in the inner-most ring of a PA, has lesser probability of moving to an adjacent PA, compared to the one, which resides in the outer ring of the previous PA, having the same mobility rate.

From Figure 5, we simply demonstrate that the PA transition probability is nothing but the ring-to-ring transition probability, which only needs a little re-calculation. For example, the probability of an MT to move from PA\(_0\) to PA\(_1\) is the probability of transition from any one of the rings (0, 1, 2) in PA\(_0\) to any one of the rings (3, 4, 5) in PA\(_1\), given that the MT was previously residing in either in ring 0 or in ring 1 or in ring 2 in PA\(_0\). Here, we assume that the probability for an MT to reside in a particular ring is the ratio of the number of cells in that ring to the total number of cells in that PA. This again implies that the probability that an MT could reside in any of the cells within an LA is equally likely, since it does not perform any LU while roaming within an LA.

![Figure 5: Partitioning LA into PAs](image)

Translating these conditional probabilities into vector-matrix notation, we can obtain PA transition probabilities. The \(\ell \times \ell\) PA transition matrix is as follows [19]:

\[
\begin{pmatrix}
P_{00} & P_{01} & P_{02} & \cdots & P_{0\ell} \\
P_{10} & P_{11} & P_{12} & \cdots & P_{1\ell} \\
P_{20} & P_{21} & P_{22} & \cdots & P_{2\ell} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P_{\ell 0} & P_{\ell 1} & P_{\ell 2} & \cdots & P_{\ell \ell}
\end{pmatrix}
\]
where, an element $p_{a_{i,j}}$ in $PA_{tran}$ is the probability for an MT to move from $PA_i$ to $PA_j$ in a single step. In general, $p_{a_{i,j}}^{(K)}$ in $PA_{tran}^{(K)}$ is the probability for an MT to move from $PA_i$ to $PA_j$ after exactly $K$ number of cell boundary crossings.

**B. Probability of Firing a Rule**

Let us assume that $t_c$ and $t_m$ are iid random variables [17] with probability density functions $f_m(t_m)$ and $f_c(t_c)$ respectively. Probability that $\chi t_m \leq t_c < \psi t_m$ can be calculated as,

$$
\xi_{\psi \chi} = \int_{t_m=0}^{\infty} f_m(t_m) \int_{t_c=\chi t_m}^{\psi t_m} f_c(t_c) dt_c dt_m
$$

If $t_m$ follows Gamma distribution and $t_c$ follows exponential distribution, for any $\chi$ and $\psi$, and a given $\gamma$,

$$
\xi_{\psi \chi} = \gamma^\gamma \left[ (\gamma + \theta \psi)^\gamma - (\gamma + \theta \chi)^\gamma \right] / (\gamma + \theta \chi)^\gamma (\gamma + \theta \psi)^\gamma 
$$

The detailed derivation is provided in Appendix II.

**C. Calculation of Paging Cost**

Let $pr_{PAC_x}$ be the probability that the $x^{th}$ PA is $PA_x$, where the last call was delivered successfully and the network is fully aware of that [18]. Then, as per [17],

$$
pr_{PAC_x} = nc_{-}PA_x/NC_{-}LA_{x_t}
$$

Let, $pr_{TC_j}$ be the probability that the $j^{th}$ rule will fire. We can obtain this from $\xi_{\psi \chi}$. Let $\rho_{i,j,x}$ be the probability for an MT to move from $i^{th}$ PA to $x^{th}$ PA when rule $j$ fires.
Then,

\[
\rho_{i,x,j} = \begin{cases} 
1 & j = 1 \\
\sum_{l=2}^{j} \sum_{K=0}^{\infty} \alpha(K)p_{i,x}^{(K)} & j = 2 \text{ to } \ell-1 \\
\sum_{K=0}^{\infty} \alpha(K)p_{i,x}^{(K)} & j = \ell 
\end{cases}
\]  

(10)

Then,

\[
nc_{PA_{x,j}} = \begin{cases} 
1 & j = 1 \\
nc_{PA_{x}} & j = 2 \text{ to } \ell 
\end{cases}
\]  

(11)

The equation (11) implies the rule, \( t_c < 0.5t_m \), would fire only when the MT receives the next call before crossing the cell boundary, where it received the previous call. Hence, it is logical that the network should search for the terminal in that particular cell only. This indicates that the number of cells that has to be polled under such situation in one. The other situations depicted in equation (11) are self-explanatory.

Now, intuitively one can say that the expected paging cost will depend on the following factors, which are the state of the system (PA_c), the rule which is being fired at that instant, and, depending on these two, how many cells in PA/PAs are to be polled for finding the MT. Combining these, we get the following expression for the expected paging cost per call arrival, denoted by \( C_v \), as

\[
C_v = V * \sum_{i=1}^{\lambda} pr_{-PAC_{i}} \sum_{j=1}^{\lambda} pr_{-TC_{j}} \min(i + j -1, \lambda) \sum_{x=\max(i-j+1,1)}^{\infty} \rho_{i,x,j} * nc_{PA_{x,j}}
\]  

(12)

D. Calculation of LU Cost

As indicated above, we have assumed that one LU takes place whenever an MT crosses an LA boundary.

Let \( C_u \) be the expected LU cost per call arrival. Then, as explained in [3],

\[
C_u = U * \sum_{K=1}^{\infty} \alpha(K) * p_{ij}^{(K)} \quad, \quad i \neq j
\]  

(13)

E. Objective Function
The expected total cost for LU and terminal paging per call arrival is given by:

\[ C_T = C_u + C_v \] (14)

Hence, the objective function can be written as:

Solve \( Z_p: \quad C_T = C_u + C_v \), (15)

subject to the constraint imposed by \( \phi \). \( Z_p \) can be solved by any standard method [17].

V. Performance Analysis

To evaluate the performance of the developed analytical model and the proposed paging scheme, we compare the total cost and the paging cost as obtained from RBPS with those obtained from SDF scheme [3] for \( U=10 \) and \( V=1 \). To demonstrate the effect of varying mobility patterns of the MTs and call arrival rate, two CMR values, namely 0.1 and 10, are considered. If \( \lambda_c \) is assumed to be constant, then a low CMR value represents fast users with high mobility rate and vice versa. So, we have taken constant value for \( \lambda_c \), equal to one and varied \( \lambda_m \) accordingly.

![Figure 6: Total cost per call arrival with \( \phi = 1 \) and \( \gamma = 1 \)](image)

Figures 6 and 7 compare the performances of SDF and RBPS, for \( \gamma \) equal to 1, in terms of total cost and paging cost separately. We have used SDF for delay equal to both 1 and 5, because SDF with
delay equal to 1 is nothing but GSM-adopted Blanket Paging scheme. The paging cost incurred in SDF is simply equal to the number of cells, as all the cells will have to be polled simultaneously. In case of RBPS, generally a selective portion of an LA, as dictated by the joint probability distribution of $t_c$ and $t_m$, will have to be polled simultaneously and only in some extreme situations the entire LA will have to be polled (as per Table 1). In both cases, typically, for larger networks, the paging cost as well as the total cost is higher for SDF, as compared to those obtained from RBPS (Figures 6 and 7). For slow users, RBPS results in even lower cost as the probability that an MT would traverse a large distance between two call arrivals is quite negligible. Consequently, as per RBPS, a small region around the cell, where the MT received the last call, would have to be searched for. A case in point is, for slow user (CMR=10), for large network (484 number of cells), the paging cost as per RBPS is about one-fifth of that obtained by SDF. Hereafter, for the sake of brevity, we have presented the comparative results of paging cost only.

With delay bound equal to 5, for fast users (CMR=0.1) our scheme does perform better than SDF scheme, which is evident from Figure 8. For slow users (CMR=10), in small sized networks comprising 11 cells to 49 cells, SDF scheme shows slightly better performance. However, for larger coverage area of a wireless cellular network, RBPS outperforms SDF by a significant margin. For example, in a network with 484 cells RBPS reduces the paging cost by 37% as compared to SDF (Figure 8).

![Figure 7: Paging cost per call arrival with $\varphi = 1$ and $\gamma = 1$](image-url)
Referring back to sequential paging, irrespective of the network size and speed of the MT, RBPS offers superior results compared to SDF. For instance, for a network with 484 cells, RBPS reduces the paging cost by around 98% for slow user and around 15% for fast user (Figure 9).

Moreover, in sequential paging, for all the cases presented here, depending upon the rule that would fire, either one single PA would to be polled or in worst case, all the five PAs would be polled. Hence, the average paging latency is reduced to 3.5.

Figure 8: Paging cost per call arrival with $\phi = 5$ and $\gamma = 1$

Figure 9: Paging cost per call arrival with $\phi = 1$ and $\gamma = 100$
Figure 10 presents the performance of SDF-adopted and RBPS-adopted blanket paging with \( \gamma \) equal to 100. The high value of \( \gamma \) signifies that, cell residence time, closely, varies around its mean value. This can be interpreted as the number of cell boundary crossings remains close to its mean value. It is apparent from Figure 10 that RBPS offers superior results compared to SDF scheme for CMR equal to 0.1. For larger network paging cost is significantly high for slow users as per SDF scheme. Consequently, RBPS offers better result. To summarize the fact, the cost incurred in SDF is more than three times for slow users in large network with 400 cells whereas in case of a coverage area with 11 cells, the cost incurred in SDF for CMR equal to 0.1 is more than six times compared to our scheme.

![Figure 10: Paging cost per call arrival with \( \phi = 5 \) and \( \gamma = 100 \)](image)

**VI. Conclusions**

In this work, we have presented a novel rule-based paging strategy, which requires a minimal amount of paging related knowledge on per user basis. RBPS translates disparate data items that have some statistical relations to appropriate paging area(s) thereby significantly reducing the signaling load. Considering the non-deterministic natures of the events (like call arrival and mobility pattern) associated with the system, we have developed a stochastic model to analyze the performance of the scheme. When compared with the popular SDF scheme [3], RBPS performs impressively in most of the cases. It reduces not only
the paging cost but also the average paging latency. The results show that RBPS is better suited than SDF for low CMR value, because, for high CMR, the cost for RBPS, though less, quickly approaches that for SDF. Moreover, no processing is needed at the terminal end. The scheme can be extended to accommodate a finite population size, and the processing overhead at the MSC end is minimal.

References
Appendix I: Location Area (LA) Transition Probability Estimation

AI.1 Derivation of LA Transition Probability

Let us assume that the MT currently resides in cell $x$ of $LA_i$, which is the current LA or $LA_c$, $c \in \mathcal{L}$. The number of cells which are immediate neighbors of cell $x$ in $LA_i$ is say, $NC_{LA_{cx}}$, $1 \leq x \leq NC_{LA_Ti}$, $1 \leq c \leq \mathcal{L}$. In this context, we define $W$ as the index set of adjacent LAs of $LA_c$ and $c \notin W$ and $W \subset \mathcal{L}$. Suppose, there are $\bar{n} \in W$, number of LAs ($LA_1, \ldots, LA_{\bar{n}}$), which are adjacent to $LA_c$ and the MT can move to any of these $\bar{n}$ LAs after performing a single cell boundary crossing and the events are equally likely.

After solving CSA problem, we can easily identify the number of neighboring cells of cell $x$ in $LA_c$ in any adjacent LA, say $LA_w$, $w \in W$ and $w \neq c$, $1 \leq c \leq \mathcal{L}$. Next, let us define $NC_{LA_{cw}}$ as the number of neighboring cells of cell $x$ in $LA_c$ in the adjacent LAs, $w \in W$ and $w \neq c$, $1 \leq c \leq \mathcal{L}$

Hence, the total number of neighboring cells, $NC_{xc_{tot}}$, around cell $x$ in any LA within the total coverage area of a PCSN, can be calculated as,

$$NC_{xc_{tot}} = \sum_{w \in W} NC_{LA_{cw}} + NC_{LA_{cx}} \quad (AI.1)$$

Next, we define $PR_{CB_{cxj}}$ as the probability that an MT in cell $x$ in $LA_c$ would cross the LA boundary in a single step and would make a transition to $LA_j$, $1 \leq x \leq NC_{LA_Ti}$, $1 \leq c \leq \mathcal{L}$, $1 \leq j \leq \mathcal{L}$ and $j \neq c$. The probability $PR_{CB_{cxj}}$ can simply be calculated as

$$PR_{CB_{cxj}} = \frac{NC_{LA_{cj}}}{\sum_{w \in W} NC_{LA_{cw}} + NC_{LA_{cx}}} \quad (AI.2)$$

where, $LA_j$ is assumed to be the destination LA ($j \in \mathcal{L}$), and $NC_{LA_{cj}}$ denotes total number of cells in $LA_j$, which are immediate neighbors to the cell $x$ in $LA_c$.

Hence, $PR_{CB_{ij}} = \frac{\sum_{x=1}^{NC_{LA}} PR_{CB_{cxj}}}{NC_{LA_i}} \quad (AI.3)$

$PR_{CB_{ij}}$ gives the probability that an MT will make a transition from $LA_i$ to $LA_j$, in single step.
For a two LA topology, we can compute a single-step 2x2 LA-transition probability matrix, which is as follows:

$$P_{LA} = \begin{bmatrix} p_{l_{11}} & p_{l_{12}} \\ p_{l_{21}} & p_{l_{22}} \end{bmatrix}$$  \hspace{1cm} (AI.4)

where $p_{l_{ij}}$, an element of $P_{LA}$, is the probability for an MT to move from $LA_i$ to $LA_j$ in single step. The generic notation for $p_{l_{ij}}$ that we have used to derive LA transition probability matrix is $PR_{CB_{ij}}$. For an L-LA topology, $LxL$ LA transition matrix can be similarly computed. For an $\hat{L}$-LA topology, $\hat{L}x\hat{L}$ LA transition matrix can be similarly computed.

**AI.2 An Example**

Let us consider a small network consisting of 11 cells, which are partitioned into two LAs. In this context, we mention that movement of an MT is feasible within the region covered by the cells and the movements are equally probable in each of the feasible direction. With the help for Figure AI.1, we explain how to derive the LA transition probability matrix. As shown in the Figure, LA\(_1\) consists of five cells and LA\(_2\) comprises six cells. If the MT resides in the cells, which are away from the LA boundary (such as cell-1 or cell-5 in LA\(_1\)), the probability to move from LA\(_1\) to LA\(_2\), in a single step is zero. If the MT resides in the boundary cells (such as cell-2, cell-6 or cell-9), the probability to cross the LA boundary in single step

![Figure AI.1 An example network](image-url)
can be derived (equation AI.1-AI.3) easily and is shown in the figure. Since the MT could be anywhere within LA$_1$, so the probability for an MT to move from LA$_1$ to LA$_2$ in a single step is

$$\text{PR}_{\text{CB}12} = pl_{12} = \frac{1}{5}(0 + 0 + 0.33 + 0.167 + 0.25) = .1494 \quad (\text{AI.5})$$

The rest of the elements in LA transition matrix $P_{\text{LA}}$ can be similarly computed. The single-step LA transition matrix, for this particular example, can be computed as

$$P_{\text{LA}} = \begin{bmatrix} 0.1494 & 0.8506 \\ 0.236 & 0.764 \end{bmatrix} \quad (\text{AI.6})$$

$$P_{\text{LA}}^{(K)} = P^K_{\text{LA}} \quad (\text{AI.7})$$

An element $pl_{ij}^{(K)}$ in $P_{\text{LA}}^{(K)}$ is the probability that an MT in LA$_i$ moves to LA$_j$ after $K$ cell boundary crossings.
Appendix II: Derivation of the Probability that a particular Rule Fires from the Rule Based System

Let us assume that call inter arrival time $t_c$ and cell-residence time $t_m$ are iid random variables with probability mass functions $f_c(t_c)$ and $f_m(t_m)$ respectively. Further we have assumed that the call inter arrival time $t_c$ follows the exponential distribution such as

$$f_c(t_c) = \lambda_c e^{-\lambda_c t_c} \quad \text{(AII.1)}$$

and cell residence time $t_m$ follows the Gamma distribution such as

$$F_m^*(s) = \left( \frac{\lambda_m \gamma}{s + \lambda_m \gamma} \right)^\gamma \quad \text{(AII.2)}$$

By taking Inverse Laplace transformation for equation AII.2, we get [17],

$$f_m(t_m) = \frac{(\lambda_m \gamma)^\gamma}{\gamma - 1} t_m^{\gamma - 1} e^{-\lambda_m t_m} \quad \text{(AII.3)}$$

We denote the probability that the $t_c$ is in the interval of $\chi t_m$ and $\psi t_m$ by $\zeta_{\chi \psi}$. We can calculate the $\zeta_{\chi \psi}$ as follows:

$$\zeta_{\chi \psi} = \int_{t_m = 0}^\infty \int_{t_c = \chi t_m}^{\psi t_m} f_c(t_c) dt_c dt_m \quad \text{(AII.4)}$$

By substituting AII.1 and AII.3 in AII.4, we can write

$$\zeta_{\chi \psi} = \int_{t_m = 0}^\infty \int_{t_c = \chi t_m}^{\psi t_m} \lambda_c e^{-\lambda_c t_c} dt_c dt_m \quad \text{(AII.5)}$$

After simplifications, we get

$$\zeta_{\chi \psi} = \frac{(\lambda_m \gamma)^\gamma}{\gamma - 1} \left[ \int_{t_m = 0}^\infty t_m^{\gamma - 1} e^{-(\lambda_m + \lambda_c) t_m} dt_m - \int_{t_m = 0}^\infty t_m^{\gamma - 1} e^{-(\lambda_m + \lambda_c) \chi t_m} dt_m \right] \quad \text{(AII.6)}$$

We can simplify the first part of the equation AII.6, as follows

$$\int_{t_m = 0}^\infty t_m^{\gamma - 1} e^{-(\lambda_m + \lambda_c) t_m} dt_m = \frac{\gamma - 1}{(\lambda_m \gamma + \lambda_c \chi)^\gamma} \quad \text{(AII.7)}$$

Similarly,
\[
\int_0^\infty t^{\gamma-1} e^{-\left(\lambda_m t + \lambda \psi\right) t_m} dt_m = \frac{\gamma - 1}{(\lambda_m \gamma + \lambda \psi)^\gamma}
\]  

(AII.8)

By substituting AII.7 and AII.8 in AII.6, we will get,

\[
\hat{\xi}_{\nu} = (\lambda_m \gamma)^\nu \left[ \frac{\gamma - 1}{(\lambda_m \gamma + \lambda \psi)^\nu} - \frac{\gamma - 1}{(\lambda_m \gamma + \lambda \psi)^\nu} \right]
\]  

(AII.9)

After simplification, we will get,

\[
\hat{\xi}_{\nu} = \gamma^\nu \left[ (\gamma + \theta \psi)^\nu - (\gamma + \theta \chi)^\nu \right] \left[ (\gamma + \theta \psi)^\nu (\gamma + \theta \psi)^\nu \right]
\]  

(AII.10)

where, \( \theta = \frac{\lambda}{\lambda_m} \), call to mobility ratio