An Information-Theoretic Framework for Optimal Location Tracking in Multi-System $4G$ Wireless Networks

Archan Misra
IBM T. J. Watson Research Center
Hawthorne, New York
Email: archan@us.ibm.com

Abhishek Roy and Sajal K. Das
CReWMaN Laboratory
Department of Compute Science and Engineering
University of Texas at Arlington
Email: {aroy, das}@cse.uta.edu

***Abstract*— An information-theoretic framework is developed for optimal location management in multi-system, fourth generation ($4G$) wireless networks. The framework envisions that each individual sub-system operates fairly independently, and does not require public knowledge of individual sub-network topologies. To capture the variation in paging and location update costs in this heterogeneous environment, the location management problem is formulated in terms of a new concept of weighted entropy. The update process is based on the Lempel-Ziv compression algorithms, which are applied to a vector-valued sequence consisting of both the mobile's movement pattern and its session activity state. Three different tracking strategies which differ in their degrees of centralized control and provide trade off between the location update and paging costs, are proposed and evaluated. While both the proposed centralized and distributed location management strategies are endowed with optimal update capability, the proposed selective location management heuristic also offers a practical trade off between update and paging costs. Simulation experiments demonstrate that our proposed schemes can result in more than 50% savings in both update and paging costs, in comparison with the basic movement-based, multi-system location management strategy. These update strategies can be realized with only modest amounts of memory (12–15 Kbytes) on the mobile.**

Keywords: Location management, update, paging, multi-system cellular networks, information theory, LZ compression.

I. Introduction

The idea of an *integrated wireless infrastructure*, encompassing multiple wireless access technologies [4], [16], is clearly a key part of the vision of packet-based, next-generation *4G* networks. This vision leverages the co-existence of multiple cellular access systems, with possibly significant differences in their individual characteristics, like, coverage area (indoor/outdoor), transmission range and link bandwidth. By using multiple, physical or software-defined radio interfaces, mobile nodes can seamlessly switch between these technologies, thus achieving the ultimate goal of global roaming [17]. The $4G$ network can thus be conceptually visualized as a collection of multiple independent access *sub-networks*, each of which has its own cellular layout and separate location tracking technique. For universal location tracking, these sub-systems are connected to one another through the Internet and share a common signaling framework.

The problem of location tracking in an individual sub-network has been extensively studied in the literature. The integrated $4G$ wireless environment, however, offers a new set of opportunities and challenges that has not yet been significantly addressed. We assume that the integrated environment will only be *loosely coupled, with no mandatory hierarchical or tiering relationship* among the component sub-networks, each of which may be managed by different operators. Most importantly, our proposed location management architecture *does not require the existence of a topology translation database, which can translate the location coordinates of a mobile node from one sub-network to another*. Additionally, we do not focus solely on non-overlapping sub-networks, where inter-system (vertical) hand-offs occur only at the edge of two sub-networks, but allow for arbitrary overlap (including total containment) between arbitrarily-sized cells of different, heterogeneous sub-networks. In emerging smart, ubiquitous computing environments, the number of such different sub-networks accessed by a single user can be far greater than the number of distinct access technologies. For example, the user can connect to different 802.11 wireless LAN (WLAN) providers while at home, in the gym or at the shopping mall. Similarly, even though multiple wide-area cellular service providers may use the same access technology, the user can dynamically switch between different providers to obtain different sets of customized services.

We focus exclusively on the *location management* problem; the complementary problem of *seamless hand-offs* (for ongoing sessions) is beyond the scope of this paper. Based on a general assumption that the movement patterns of a mobile node (MN) are piece-wise stationary, the LeZi-Update [5] strategy uses the Lempel-Ziv ($LZ78$) compression algorithm [18] to reduce the location update cost in a *single-system network*, to an asymptotically optimal value. Our multi-system location management framework is also based on this $LZ78$ algorithm, and thus allows the network to optimize the location management cost (especially the update cost) for each individual MN, *without any prior assumption of its movement pattern.* However, we shall point out that the multi-system framework has several fundamental differences with the existing LeZi-Update scheme. Our contribution lies in explaining how significant enhancements in the location update and paging strategies are necessary for developing a practical, information-theoretic mobility management solution for heterogeneous $4G$ networks.

To begin with, we shall show why the location update decisions for a multi-system network must consider the MN's associated *activity state* (active or idle), in addition to its movement pattern. In a single-system network, location tracking is employed only for an idle MN (one with no active sessions). In the multi-system environment, the MN's activity state is, however, vector-valued, since the MN can be active in one sub-network and simultaneously idle in another. Moreover, the optimal paging sequence depends on the relationship between the movement and the session arrival patterns.

We shall extend the information-theoretic framework of LeZi-Update [5] to introduce the concept of *weighted entropy*, where the location uncertainty of the MN in a particular sub-network is weighed in proportion to the signaling costs in that sub-network. This provides a more accurate model for measuring location uncertainty in heterogeneous $4G$ networks, where different access technologies might have very different communication costs. For example, the higher uplink transmission-power of a satellite-based cellular network results in a higher location update cost for an MN's satellite radio interface than it's local area network (e.g., Bluetooth-based) interface.

We shall provide three different architectural alternatives for information-theoretic multi-system location management, each requiring a different level of coordination among the individual sub-networks.

- In the first *centralized* scheme, the entire location update and movement history for a particular MN is assumed to be stored at a central coordinating server, which can then coordinate the paging activities in each individual sub-network. This architecture provides optimal location tracking at the expense of requiring closer coordination among the component sub-networks, and possible signaling bottlenecks at the central coordinator.
- In the second *distributed* alternative, each individual sub-network possesses complete information about the mobile's combined movement-and-calling sequence in all sub-networks. By eliminating the central coordinator, it allows the location management scheme to be completely distributed, at the expense of higher update cost on the MN.
- The third alternative *heuristic* scheme operates as a compromise between these two extremes. In this approach, a central coordinator stores only partial (and aggregate) information about the MN's location uncertainty in each sub-network, rather than the complete history of the MN. An MN may now issue location updates selectively to individual sub-networks, thereby reducing its location update cost at the expense of higher paging overheads.

We have used mathematical analysis and simulation results to demonstrate how these architectural alternatives can provide integrated location management with much lower cost, and with quite moderate memory requirements.

By operating in *symbolic space*, our location management framework is truly universal and is also capable of supporting sub-networks having geometric location information. Indeed, each sub-network is completely free to possess its own internal location management technology. Our framework never requires (even in the case of a centralized coordinator) individual sub-networks to publicly share their topological information. We believe that this is an extremely attractive feature, since individual sub-network operators are usually extremely reluctant to expose their topological layout to potential competitors. *Accordingly, our system is designed to provide seamless mobility tracking in a truly multi-operator 4G environment, where the different radio interfaces on the MN may be associated with different service providers.*

The rest of the paper is organized as follows. In Section II we survey related work on multi-system location management and explain the key differences in our approach. Section III presents the system description and mathematical model of *weighted entropy* for estimating the location uncertainty in multi-system $4G$ networks. Section IV presents the optimal paging and update schemes for the centralized framework. The distributed analogue of this location management approach is presented in Section V. Section VI then proposes the selective location update strategy. Simulations studies in Section VII are used to evaluate the performance of our three location management schemes. Finally, Section VIII concludes the paper.

II. Related Work on Location Management

Given the large body of work on location tracking in a single cellular access system (e.g., personal communication systems or PCS), we survey only the broad categories of location management solutions. PCS networks typically cluster groups of cells into *registration areas* (RA), such that the location uncertainty of an MN is confined to its last reported RA. In this approach, an MN performs proactive location updates only when it changes its current RA, and not on every cell-change. In general, most location update strategies can be classified into three groups: distance-based, movement-based, and time-based. The relative performance of these schemes for different movement patterns has been analyzed in [3]. Alternative proposals for location-update algorithms include the reporting center strategy [2], where an MN updates its location only when entering designated cells; and the user mobility pattern scheme [7], where each MN has an RA that adapts in response to its movement patterns.

The problem of optimal paging in a single system in terms of cell residency probabilities, both with and without constraints on the acceptable paging latency, has been discussed in [12]. An intuitive, but fundamentally useful, result in [12] states that, in the absence of constraints, the expected paging cost is minimum when the cells are paged sequentially in the descending order of occupancy probabilities. Other paging schemes, like directional-based paging [3], [6] or profile-based paging [11] have been proposed to modify the cell residency probabilities, based on the MN's recent or historical movement patterns.

Some recent work has reported on the problem of location management in a multi-system environment, although mainly for the non-overlapping scenario. It has been shown in [1], [15] that an integrated location management strategy can significantly outperform an independent operation of each sub-system's location management algorithm. For managing transitions across different sub-network domains, The concept of boundary location registers presented in [1], helps to migrate the MN's location information from one sub-system to another. At a protocol level, Mobile IP functionality has been extended in [14] to support movement across multiple access networks. However, the problem of integrated location tracking in a generic multi-system environment is yet to be effectively addressed. In particular, a solution must consider both the non-zero probability that an MN is out of the coverage area of one or more sub-networks, and the possibility that an MN might have different sessions (calling) activity states in each sub-network.

III. System description and Location Uncertainty

Figure 1 shows an example of an *integrated 4G network*, comprising a collection of satellite, PCS and campus area (IEEE 802.11 WLAN and Bluetooth) based independent sub-networks. Clearly, the coverage area of each individual sub-system can be discontinuous (e.g., a set of disconnected islands of 802.11-based hot-spots). Accordingly, the set of sub-networks that can be accessed concurrently by a mobile node is not constant but a function of its current location.

Let the $4G$ network consist of N sub-networks or access technologies $\{S_1, S_2, \cdots, S_N\}$, where each sub-network is a collection of (either partitioned or overlapping) cells, such that C_i^j represent the j^{th} cell in the i^{th} sub-network. Let $|S_i|$ represents the cardinality of (number of cells in) S_i. For our environment, we must consider the case where the MN roams out of the coverage area of an individual sub-network. Indeed, the $4G$ network will often contain islands of local area networks, such as 802.11 LANs, with which the MN is only intermittently connected. For notational convenience, let each sub-network have an additional cell ϕ to capture this disconnected state. Accordingly, if the MN is currently out of range of sub-network S_i, its location vector includes the cell C_i^{ϕ}. To model the multi-system environment, where different sub-networks can have different paging and location update costs per transmitted message, let PG_i represent the cost of transmitting a single paging message in a single cell, and LU_i be the cost of transmitting a location update message, in the i^{th} sub-network S_i.

Similar to LeZi-Update [5], our framework is based on the symbolic interpretation of the user's mobility (movement) in each sub-system. Accordingly, the movement pattern of an MN in each access network can be represented by a sequence of symbols. Our optimal location update strategy does not issue location updates on every movement of the MN, but only on an appropriately determined (entropy-coded) sub-set of this movement sequence. While the LeZi-Update strategy can be applied to time and distance-based strategies as well, without loss of generality, we consider a movement-based location update strategy, where a new symbol is generated only when the MN changes a cell in any one of the different sub-systems. The movement pattern of the MN can then be represented as

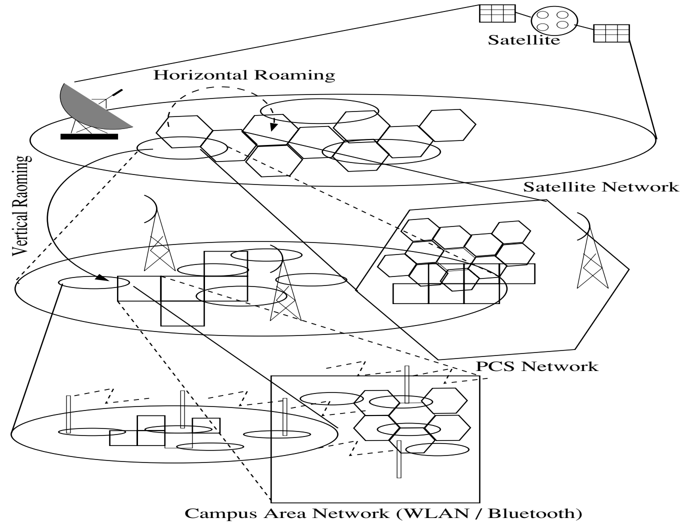

Fig. 1. A Multi-System Heterogeneous Wireless Network

a vector-valued random variable $\bar{X}$ of dimension N, where the i^{th} element of the vector corresponds to the current cell of sub-network S_i. For instance, if $\bar{X}(2) = 4$, the MN is currently located in the 4^{th} cell of sub-network S_2. Clearly, the overall uncertainty in the user's location can be modeled by the joint distribution:

$$\begin{aligned} & Pr\left(\bar{X} = [x_1, x_2, \cdots, x_N]\right) \quad (1) \\ = \; & Prob\Big(\text{MN is located in } C_1^{x_1} \bigwedge C_2^{x_2} \bigwedge .... \bigwedge C_N^{x_N}\Big) \end{aligned}$$

Note that one or more of these elements may have the value ϕ, implying that the MN is outside the coverage area of the corresponding sub-network. Characterizing the mobility as a probabilistic sequence suggests that it can be defined as a stochastic, vector-valued process $\chi = \{\mathcal{X}^n\}$, where the *repetitive* nature of *patterns* adds (piece-wise) *stationarity* as an essential property. Based on this N-valued random vector $\bar{X}$, it is easy to define the *conditional entropy* [8] of the vector-valued random process χ corresponding to the n-element random vector-sequence $\{\mathcal{X}^n\} = \{\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_n\}$. Formally, the conditional entropy $H(\chi)$ is given by:

$$H(\chi) = \lim_{n \to \infty} H(\bar{X}_n | \bar{X}_0, \bar{X}_1, ..., \bar{X}_{n-1}), \quad (2)$$

where $H(\bar{X}) = -\sum_{\bar{x} \in \chi} Pr(\bar{x}) \lg\left[Pr\left(\bar{x}\right)\right]$. Clearly, the stationary stochastic sequence $\chi = \{\mathcal{X}^n\}$ is associated with an entropy, such that the movement-based location update cannot generate an update cost (measured in the *number* of actual update bits) per movement smaller than this fundamental limit.

Since the vector-valued entropy in Equation (2) does not account for the fact that different update/paging messages (corresponding to different sub-systems) have different costs, we propose the concept of *weighted entropy* or the *minimum weighted cost* per movement. Note that, the random vector $\bar{X}_n$ differs from the random vector $\bar{X}_{n-1}$ only in one element, i.e., an MN is not allowed to change its associated cell in two or more sub-systems simultaneously.[1] Thus, if the difference is only in the i^{th} element, the MN has changed its cell in sub-network S_i. We assume that in an idealized setting, the MN would inform its location update to a centralized, location tracking system using any sub-system S_i where it has changed cells, thereby incurring an update cost of LU_i. We thus weigh the cost of conveying an information change in the i^{th} element of the random vector $\bar{X}$ in proportion to the associated update cost LU_i. Accordingly, the *weighted entropy* is given by:

$$\begin{aligned} H_w(\chi) &= \lim_{n\to\infty} \sum_{i=1}^{N} LU_i \times \sum_{j=1}^{|S_i|} [Pr[\bar{X}_n = (\ldots, C_i^j, \ldots)| \\ &\quad \bar{X}_{n-1}, \ldots, \bar{X}_0] \lg(Pr[\bar{X}_n = (\ldots, C_i^j, \ldots)| \\ &\quad \bar{X}_{n-1}, \cdots, \bar{X}_0])], \end{aligned} \tag{3}$$

where the $\ldots$ imply that the corresponding random variable can take any possible value within its range.

A. The Need for Session State Information

While Equation (3) does capture the weighted cost in performing location updates, it does not exploit the correlation between the movement and session arrival patterns, which can be important for location tracking in a multi-system environment. In a single-system environment, the MN is paged only when it is currently idle in that system. In a multi-system environment, paging can also be performed in one sub-network S_i by employing signaling in another sub-network S_j to *trigger* a location update in S_i. For example, if the sub-network S_i is *aware* that the MN is currently active in S_j, then it may make an indirect request to S_j, asking it to direct the MN to provide an update for its location in S_i. The advantage in this case is that, since S_j is aware of the precise location of the MN (which currently has an active session in S_j), this signaling consumes only one transmission unit PG_j in S_j. Clearly, this replaces the possibility of multiple paging messages PG_i being transmitted by sub-system S_i to locate the MN.[2] Note that, S_i may choose to page the mobile node via an alternative sub-network S_k, even if the MN is known to be currently idle in S_k. Intuitively speaking, this approach of indirect (inter-system) paging will result in cost-savings when the location uncertainty in S_k is suitably lower than the location uncertainty in S_i.

To capture the MN's session activity state, we define another vector-valued random variable $\bar{Z}$, with its i^{th} element corresponding to the activity state of the MN in S_i. Moreover, each element in $\bar{Z}$ is binary-valued: such that,

$$\bar{Z}_i = \begin{cases} 0, \text{ if MN is idle in } S_i \\ 1, \text{ if MN is active in } S_i. \end{cases} \tag{4}$$

[1]In general, an MN can change its cell association with two sub-systems, say S_i and S_j, simultaneously (if the cells in the two sub-systems exactly overlap). To apply our formulation in this case, we merely need to randomly order these transitions, and treat them as a distinct set of ordered transitions.

[2]In general, it does not follow that an indirect registration request through an alternative sub-network S_j, where the MN is known to be active, is always better than paging the MN in S_i. The correct decision depends on the relative paging costs, and the degree of uncertainty about the MN's location in S_i.

To exploit the correlation between $\bar{X}$ and $\bar{Z}$, we now define a $2N$-valued random vector $\bar{Y}$, consisting of alternating elements of $\bar{X}$ and $\bar{Z}$. Accordingly, if $N = 3$, $\bar{X} = (C_1^5, \phi, C_3^4)$ and $\bar{Z} = (1, 0, 0)$, $\bar{Y} = (C_1^5, 1, \phi, 0, C_3^4, 0)$. To provide an intelligent paging scheme, the MN must keep the network informed of *appropriate patterns* of the random process ν corresponding to the n-element random vector-sequence $\{\mathcal{Y}^n\} = \{\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_n\}$.[3] Under this definition, the cost entropy associated with the MN's combined calling and movement pattern is given by:

$$\begin{aligned} H_w(\nu) &= \lim_{n\to\infty} \sum_{i=1}^{N} LU_i \times \Big[\sum_{j=1}^{|S_i|} Pr(\bar{Y}_n = (\ldots, C_i^j, 0, \ldots) \\ &\quad \bar{Y}_{n-1}, \ldots, \bar{Y}_0) \lg(Pr(\bar{Y}_n = (\ldots, C_i^j, 0, \ldots)|\bar{Y}_{n-1}, \\ &\quad \cdots, \bar{Y}_0)) + \sum_{j=1}^{|S_i|} Pr(\bar{Y}_n = (\ldots, C_i^j, 1, \ldots)|\bar{Y}_{n-1} \ldots, \\ &\quad \bar{Y}_0) \lg(Pr(\bar{Y}_n = (\ldots, C_i^j, 1, \ldots)|\bar{Y}_{n-1}, \cdots, \bar{Y}_0))\Big] \end{aligned} \tag{5}$$

Result *1: For a given stationary stochastic process ν, no location tracking scheme can predict the precise location of the MN (across all sub-networks) with a smaller average update cost, per change in state, than $H_w(\nu)$.*

The weighted entropy of any user's movement and activity pattern thus provides a lower bound against which the performance of any practical location management scheme may be measured. Of course, computing this theoretical bound a priori requires exact knowledge of the statistics of the corresponding random process, a requirement that is clearly not practical. In the next few sections, we shall instead focus on feasible location update strategies that employ on-line learning to asymptotically achieving this performance bound (without knowing ahead of time what the exact bound is), and paging strategies that exploit these location updates.

IV. Centralized Optimal Location Tracking Strategy

In this section, we consider a model where a centralized location coordinator tracks the location of an MN across all sub-networks. In this approach, the coordinator is connected to each of the individual sub-networks. Every location update from the MN to any sub-network is relayed by the corresponding sub-network to the coordinator, which is thus *aware of the entire update pattern of the MN*. The coordinator does not, however, need to know the topological layout of each sub-network. When attempting to locate the MN, an individual sub-network must consult the coordinator to determine the optimal paging sequence, *across all sub-networks.* This approach clearly imposes the highest load on the coordinator, since it

[3]Each MN updates the location management system of its call state *at the time of the location update.* The location tracking system does not, however, have exact knowledge of the MN's activity state at other instants, including between location updates.

must receive and process every update. Moreover it has to compute the optimal paging sequence for every paging activity. Of course, the coordinators themselves may be distributed, with each coordinator responsible for tracking the location of a collection of mobile nodes. A separate protocol (for example, similar to Mobile IP) may be used to associate each MN with its unique coordinator.

A. The Optimal Location Update Algorithm

The location update strategy of the MN is based on the Lempel-Ziv (LZ) compression algorithm. Unlike the existing LeZi-Update scheme [5], our LZ-compression technique is applied to the entire $2N$-valued vector sequence $\{\mathcal{Y}^n\} = (\bar{Y}_1, \ldots, \bar{Y}_n)$, rather than on a single valued random sequence for each individual sub-network. The algorithm for generating updates (at the MN) and for processing them (at the coordinator) is described in Figures 2 and 3, respectively. Whenever the mobile changes its state $\bar{Y}_n$ to a distinct new value $\bar{Y}_{n+1}$, it checks if the sequence $\bar{Y}_{k+1}, \ldots, \bar{Y}_{n+1}$ (where $\bar{Y}_k$ resulted in the previous updated) has been encountered before. If so, it does not update; otherwise it reports the new sequence to the centralized coordinator in a compressed form. Essentially MN acts as the multi-variate encoder and the coordinator plays the role of a multi-variate decoder. Thus, not only the movement, but the state-history "$\bar{Y}_1\bar{Y}_2\bar{Y}_3\ldots$" reaches the central repository as a sequence "$\mathcal{C}(\bar{w}_1)\mathcal{C}(\bar{w}_2)\mathcal{C}(\bar{w}_3)\ldots$," where $\bar{w}_i$s are non-overlapping, distinct segments of the string "$\bar{Y}_1\bar{Y}_2\bar{Y}_3\ldots$" and $\mathcal{C}(\bar{w})$ is the encoding for segment $\bar{w}$. For example, the input string "$\bar{a}\bar{j}\bar{l}\bar{l}\bar{o}\bar{o}\bar{j}\bar{h}\bar{h}\bar{a}\bar{a}\bar{j}\bar{l}\bar{l}\bar{o}\bar{o}\bar{j}\bar{a}\bar{a}\bar{j}\bar{l}\bar{l}\bar{o}\bar{o}\bar{j}\bar{a}\bar{a}\bar{j}\bar{l}\bar{m}\ldots$" (where each symbol is $2N$-valued) is parsed as distinct substrings (phrases): "$\bar{a}$, $\bar{j}$, $\bar{l}$, $\bar{l}\bar{o}$, $\bar{o}$, $\bar{j}\bar{h}$, $\bar{h}$, $\bar{a}\bar{a}$, $\bar{j}\bar{l}$, $\bar{l}\bar{o}\bar{o}$, $\bar{j}\bar{a}$, $\bar{a}\bar{j}$, $\bar{l}\bar{l}$, $\bar{o}\bar{o}$, $\bar{j}\bar{a}\bar{a}$, $\bar{j}\bar{l}\bar{m}$, $\ldots$,". Such a symbol wise context model is efficiently stored in a dictionary implemented as a search trie. Figure 4 shows these different phrases with their frequencies, where the frequency of every symbol is incremented for *every prefix of every suffix* of each phrase [5]. The incremental parsing accumulates larger and larger phrases in the dictionary, thereby accruing estimate of entropy of all possible orders. Essentially, the algorithm approaches optimality for stationary, ergodic sources [18].

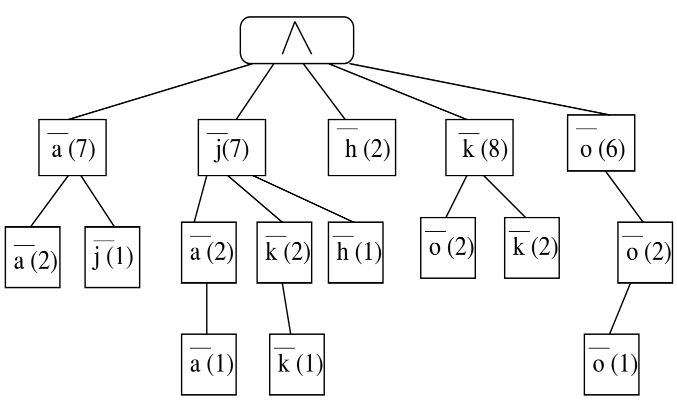

Fig. 4. Decoder Trie in the Multi-System $4G$ Networks

Result 2: *The multi-variate LeZi-Update algorithm asymptotically approaches a location update cost, per change in sequence $\{\mathcal{Y}^n\}$, defined by $H(\nu)$.*

B. The Normalized Probabilistic Paging Algorithm

The centralized strategy essentially implements the paging by performing a *trie-traversal* to compute the unconditional residence probabilities of the MN at each cell of every sub-system that is present in the trie. Let ϱ_i^j represent the unconditional residence probability of the MN in cell C_i^j. Before describing how this unconditional residence probability is computed, we first prove how the system can utilize this probabilities and the differential paging cost $PG_j, \ \forall j \in \{1, 2, \ldots, N\}$, associated with the cells in different sub-networks to compute the optimal paging sequence (at least, the optimal sequence that relies purely on the unconditional marginal residence probabilities):

Result 3: *In the absence of any delay bounds on the paging scheme, the optimal (minimum expected cost) paging algorithm is based on sequentially paging all cells (except the dummy cells C_i^{ϕ} $\forall i$) in the descending order of their* normalized residency factor $r_i^j, \ \forall i \in \{1, 2, \ldots, N\}, \ \forall j \in |S_i|, \ j \neq \phi$. *The normalized residency factor for cell C_i^j is given by $r_i^j = \frac{\varrho_i^j}{PG_i}$.*

Proof: The proof of the optimal paging sequence is similar to the analogous proof for the conventional paging sequence (with equal weights for the paging cost in all cells) presented in [12]. We prove this proposition by contradiction. Assume that the optimal paging schedule A is given by the cell-sequence $C(1), C(2), C(3), \ldots$, i.e., the cells are indexed according to their paging order. (Note that the symbols $C(k)$ and C_i^j are different. While $C(k)$ represents k^{th} cell in the optimal paging order, C_i^j denotes the j^{th} cell in i^{th} sub-network). Similarly, let $PG(l)$ denote the paging cost and $\varrho(l)$ denote the unconditional occupancy probability for the cell $C(l)$, so that the normalized residency factor for cell $C(l)$ is given by $r(l) = \frac{\varrho(l)}{PG(l)}$. If the MN is indeed in cell $C(k)$, then the algorithm pages all cells $C(1), C(2), \ldots, C(k)$, thereby incurring a total paging cost $Q_k = \sum_{l=1}^{k} PG(l)$. Accordingly, the expected paging cost for schedule A equals $E[P(A)] = \sum_{l=1}^{N} Q_l \times \varrho(l)$.

Now, suppose that their exist two cells $C(m)$ and $C(n)$ in schedule A, such that $r(m) < r(n)$ and $m < n$, i.e., m^{th} cell is paged before n^{th}. We can then construct another sequence A', simply by exchanging the places of $C(m)$ and $C(n)$. It is then easy to see that the difference between the expected costs $E[P(A)]$ and $E[P(A')]$ is given by:

$$\begin{aligned}
E[P(A)] - E(P(A')] &= PG(m)\,\varrho(n) - PG(n)\,\varrho(m) \\
&= PG(m)\,\varrho(n)\left(1 - \frac{PG(n)\,\varrho(m)}{PG(m)\,\varrho(n)}\right) \\
&= PG(m)\,\varrho(n)\left(1 - \frac{r(m)}{r(n)}\right) \qquad (6) \\
&\geq 0 \text{ (by assumption).}
\end{aligned}$$

```
initialize dictionary := NULL
initialize phrase-vector w̄:= NULL
loop
  wait for next vector-symbol Ȳ
  if(w̄.Ȳ in dictionary)
    w̄ := w̄.Ȳ
  else
    encode <index(w̄),Ȳ>
    add w̄.Ȳ to dictionary
    w̄ := NULL
  endif
forever
```

Fig. 2. Encoder at the mobile

But, this clearly violates our assumption of optimality of A, thereby proving our theorem. Note, however, that using joint residence probabilities, instead of purely marginal probabilities, may allow us to exploit additional correlation in the residence patterns (e.g, if the MN always simultaneously resides in C_1^4 and C_2^1). We defer the investigation of such algorithms to future research.

The unconditional residence probabilities are computed using the PPM (prediction by partial match) style blending techniques discussed in [18]. In general, each symbol in the trie is a cell-activity tuple, i.e., the trie stores separate symbols for the probabilities $\varrho_i^j(0)$, imp;ying MN is in cell C_i^j and is idle in S_i; and $\varrho_i^j(1)$, implying MN is in cell C_i^j and has an active session in S_i. The residence probability computation algorithm, shown in Figure 5 starts from the highest order of context (leaves in the trie) and *escapes* to lower order, until order-0 (the root) is reached. If $\bar{\xi}$, $\mathcal{N}(\bar{\omega})$, $\mathcal{L}(\bar{\omega})$, $\mathcal{S}^k(\bar{\omega})$ and $\mathcal{P}(\bar{\omega})$ denote the last updated phrase, number of occurrences of a phrase $\bar{\omega}$, its length, k-th suffix, and prefix respectively, the probability of any phrase $\bar{\psi}$ can be estimated by the recursive formula:

$$Pr\left[\bar{\psi}\right] = \frac{\mathcal{N}(\bar{\psi} \mid \mathcal{P}(\mathcal{S}^k(\bar{\xi})))}{\sum_{\bar{\omega}} \mathcal{N}(\bar{\omega} \mid \mathcal{P}(\mathcal{S}^k(\bar{\xi}))} + \frac{\mathcal{N}(\Lambda \mid \mathcal{P}(\mathcal{S}^k(\bar{\xi})))}{\sum_{\bar{\omega}} \mathcal{N}(\bar{\omega} \mid \mathcal{P}(\mathcal{S}^k(\bar{\xi})))} \times \Pr\left[\mathcal{S}^1(\mathcal{P}(\mathcal{S}^k(\bar{\xi})))\right] \quad (7)$$

for all k, where $1 \le k \le \mathcal{L}(\bar{\xi})$. Thus, considering "$jll$" as the latest state-update reported in the centralized-server, the usable contexts are "jl" (order-2), "j" (order-1) and "Λ" (order-0). A list of all predictable state-sequence with frequencies at this context are shown in Table I. Subsequently, the probabilities associated with all such segments are shown in Table II.

TABLE I
PHRASES AND THEIR FREQUENCIES AT DIFFERENT ORDERS

$\bar{j}\bar{l}$ (order-2)	$\bar{j}$ (order-1)	Λ (order-0)		
$\bar{l}\mid\bar{j}\bar{l}(1)$	$\bar{a}\mid\bar{j}(1)$	$\bar{a}(4)$	$\bar{a}\bar{a}(2)$	$\bar{a}\bar{j}(1)$
$\Lambda\mid\bar{j}\bar{l}(1)$	$\bar{a}\bar{a}\mid\bar{j}(1)$	$\bar{j}(2)$	$\bar{j}\bar{a}(1)$	$\bar{j}\bar{a}\bar{a}(1)$
	$\bar{l}\mid\bar{j}(1)$	$\bar{j}\bar{l}(1)$	$\bar{j}\bar{h}(1)$	$\bar{l}(4)$
	$\bar{l}\bar{l}\mid\bar{j}(1)$	$\bar{l}\bar{o}(1)$	$\bar{l}\bar{o}\bar{o}(1)$	$\bar{l}\bar{l}(2)$
	$\bar{h}\mid\bar{j}(1)$	$\bar{o}(4)$	$\bar{o}\bar{o}(2)$	$\bar{h}(2)$
	$\Lambda\mid\bar{j}(2)$	$\Lambda(1)$		

```
initialize dictionary := NULL
loop
  wait for next codeword
  decode phrase-vector
  add phrase-vector to dictionary
  increment frequency for every prefix
            of every suffix of phrase-vector
forever
```

Fig. 3. Decoder at the system

```
Initialize i := 0, Pr[ψ̄] := 0, h:=highest order
Initialize escape probability Pr_h^(e) := 1
While (i ≤ h)
      Search for ψ̄ at order h − i
      If (ψ̄ found)
         Compute its (h − i)^th order
         probability (Pr_{h−i}[ψ̄])
      else
           Pr_{h−i}[ψ̄] := 0
      End-if
      Compute the escape probability (Pr_{h−i}^(e))
      to order (h − i)
      Estimate the combined probability as
      Pr[ψ̄] := Pr[ψ̄] + Π_{j=h}^{h−i} Pr_j^(e) × Pr_{h−i}[ψ̄]
      i := i + 1
End-while
```

Fig. 5. Residence Probability Computation Algorithm

TABLE II
UNCONDITIONAL RESIDENCE PROBABILITIES

Phrase	$Pr[Phrase]$	Phrase	$Pr[Phrase]$
$\bar{l}$	0.5905	$\bar{o}\bar{o}$	0.0095
$\bar{l}\bar{l}$	0.0809	$\bar{h}$	0.0809
$\bar{l}\bar{o}$	0.0048	$\bar{j}$	0.0095
$\bar{l}\bar{o}\bar{o}$	0.0048	$\bar{j}\bar{h}$	0.0048
$\bar{o}$	0.0195	$\bar{a}$	0.0905
$\bar{j}\bar{a}$	0.0048	$\bar{j}\bar{a}\bar{a}$	0.0048
$\bar{j}\bar{l}$	0.0048	$\bar{a}\bar{a}$	0.0809
$\bar{a}\bar{j}$	0.0048		

Following the principle of *insufficient reason* [10], the phrase probability mass is distributed according to its *type* or *composition*. The probability of an individual symbol ($2N$-valued cell-activity vector), is thus essentially computed based on the relative weights of symbols in these phrases. Formally, the probability $\varrho^{\bar{Y}_k}$ of each state (location and call-activity) can then be obtained as

$$\varrho^{\bar{Y}_k} = \sum_{\bar{\psi}} \frac{\mathcal{N}(\bar{Y}_k)}{\mathcal{L}(\bar{\psi})} \times \Pr\left[\bar{\psi}\right]. \quad (8)$$

Note that, $\varrho^{\bar{Y}_k}$ is different from ϱ_i^j. While ϱ_i^j represents the unconditional residence probability at cell C_i^j, $\varrho^{\bar{Y}_k}$ denotes the unconditional probability of vector $\bar{Y}_k$, containing the N cells. After obtaining the unconditional vector probabilities, we translate these into the unconditional (cell, state) probabilities for each cell within its own sub-system. To obtain these probabilities, we essentially treat each element pair of the vector-valued symbols as having the corresponding probability mass, and then normalize these values over the element pairs for each system. For example, consider the case where the trie-based traversal results in the following probability assignments: ($[C_1^2, 0, C_2^1, 0] = 0.3$, $[C_1^2, 0, C_2^3, 0] = 0.15$, $[C_1^2, 0, \phi, 0] = 0.2$, and $[C_1^3, 1, C_2^3, 0] = 0.35$). We then obtain the individual (cell, activity) probabilities as: $\rho_1^2(0) = \frac{0.65}{1}$, $\rho_1^3(1) = \frac{0.35}{1}$, $\rho_2^1(0) = \frac{0.3}{1}$, $\rho_2^3(0) = \frac{0.5}{1}$, and $\rho_2^{\phi} = \frac{0.2}{1}$. Finally, the unconditional residence probability ϱ_i^j of each cell j in ith sub-system is computed by adding the unconditional probabilities $\varrho_i^j(1)$ and $\varrho_i^j(0)$. The paging algorithm then pages each cell in the decreasing order of their normalized residency factor.

An interesting question relates to the complexity of this algorithm (Figure 5). Clearly, as the length of the sequences generated by the LZ compression scheme increases, so does the complexity of escaping from the leaf of a trie towards its root. We can bound the complexity of computing sub-phrases from the phrase by noting the following result [13] on the maximum length sequence (depth) of the Lempel-Ziv trie:

Result *4: If $h_1 = \lim_{n\to\infty} \frac{\lg(1/min\{Pr(\mathcal{Y}^n)\})}{n}$ and $h_2 = \lim_{n\to\infty} \frac{\lg(E\{Pr(\mathcal{Y}^n)\})^{-1}}{2n}$, respectively denotes Rényi entropy of order $-\infty$ and order 0, then for a stationary, ergodic sequence $\{\mathcal{Y}^n\}$ of length n, the depth (ℓ) of this Lempel-Ziv trie is almost surely oscillating between $(\lg n)/h_1$ and $(\lg n)/h_2$. More precisely,*

$$\lim_{n\to\infty} \inf_n \ \ell = \frac{\lg n}{h_1}, \qquad \lim_{n\to\infty} \sup_n \ \ell = \frac{\lg n}{h_2}, \tag{9}$$

where $0 < h_1 < H(\nu) < h_2 < \infty$. Let ℓ_1 denote the upper bound $(\lg n)/h_1$, and ℓ_2 the lower bound $(\lg n)/h_2$.

Now, if the total cost for computing the probability of every sub-path of a path of length ℓ is denoted as $Cost(\ell)$, we can see that $Cost(.)$ satisfies the recurrence relation:

$$Cost(\ell) = Cost(\ell-1) + Cost(\ell-2) + \ldots, +Cost(1),$$

with $Cost(1) = 1$. Accordingly, we have $Cost(\ell) = O(\ell^2)$. Moreover, since the asymptotic update rate for Lempel-Ziv is $H(\nu)$ bits/symbol, it follows that the number of unique phrases in the LZ trie for the sequence $\{\mathcal{Y}^n\}$ is asymptotically bounded by $2^{nH(\nu)}$. Accordingly,

Result *5: The complexity, $PG_{complex}$, of computing the residence probabilities from the LZ trie is bounded by:*

$$PG_{complex} \approx O(\ell^2) \times 2^{nH(\nu)}, \tag{10}$$

where ℓ is given by Equation (9). Clearly, as n increases, the update sequences will get longer, resulting in an increase in the memory overhead of the trie, and its associated computing cost. At some point, the trie must be flushed or reset.

V. De-centralized Location Tracking Scheme

The location management strategy in Section IV can cause performance bottlenecks at the centralized coordinator, which must process all the location updates, and also generate all the paging sequences. In this section, we present the other extreme, a purely distributed strategy, where each sub-network independently computes its own paging schedule. The solution works by maintaining a copy of the trie in each sub-network. We note that the computation of the paging sequence is based entirely on the contents of the trie (and the knowledge of the paging costs for each sub-system). Accordingly, if the MN were to transmit every issued update (the encoded sequence $\{\mathcal{C}(\bar{w}_i)\}$) to all the sub-networks, each sub-network would store an identical copy of the MN's update history in its trie, and would thus be able to implement the optimal paging strategy independently. This scheme assumes that the weighted cost factors LU_i and PG_i for each sub-network S_i $i \in \{1, 2, \ldots, N\}$, are well known constants that are globally available.

A. The Location Update Strategy

The location update strategy requires the MN to transmit every encoded update to *all* sub-systems. This requirement can be understood by realizing that the paging algorithm essentially considers only those symbols that occur at least once in the trie. Accordingly, if the MN skipped some updates, a particular sub-network, by mistake, might estimate a zero residence probability in some cell, causing the paging process to fail. For example, in the sequence $\bar{a}\bar{b}\bar{a}\bar{c}\bar{a}\bar{c}$, the MN would normally issue the updates $\{\bar{a}\}, \{\bar{b}\}, \{\bar{a}\bar{c}\}$ and suspend location update until it moves to a new symbol, thereby generating a previously unseen pattern $\{\bar{a}\bar{c}*\}$. If a sub-system does not get the last update $\{\bar{a}\bar{c}\}$, it would assign a 0 probability mass to symbol $\bar{c}$ and never page it. The resulting location update and paging strategy is then an extension of the centralized scheme. Since each update is now sent to every sub-network, it incurs a cost of $\sum_{i=1}^{N} LU_i$, thus leading to the following result:

Result *6: The "all-sub-network" (de-centralized) location update strategy asymptotically incurs an update cost of $H(\bar{Y}) \times \sum_{i=1}^{N} LU_i$, where $H(\bar{Y})$ is the un-weighted entropy of the random sequence $\{\bar{\mathcal{Y}}_n\}$.*

In the distributed sub-network case, the location update strategy must also account for the possibility of extended disconnections from a particular sub-network. Clearly, as long as the MN is beyond the coverage area of sub-network S_i, the trie for S_i does not get updated since it does not receive any new updates from the MN. To ensure consistency across the various tries, the MN must keep track of the missed updates for each disconnected sub-network, and must issue *delayed* updates whenever it re-attaches to the corresponding sub-network.

B. The Constrained Paging Scheme

In general, each sub-network can simply implement the paging strategy of Section IV-B. Given the distributed nature of the system, we can, however, impose additional constraints on the paging sequence, and develop a corresponding strategy under these constraints. The centralized case assumed that the central coordinator could define a purely cell-based paging sequence, i.e., the paging sequence could interleave cells from different sub-networks in any random order. In a practical scenario, such intermittent paging requests from one sub-network to another might be prohibited. A sub-system S_i might then be constrained to issue only a single paging request for any other sub-network S_j: this paging request would specify the entire collection of cells in that S_j with non-zero residence probability. (Of course, S_j would still page its own cells in descending order of residence probability.) Moreover, we allow a sub-network to issue two distinct types of paging requests to a target sub-network: (a) an "active" paging request, that asks the target sub-network to request the MN for a proactive registration, only if it is currently active in the target sub-network, and (b) an "idle" paging request, where the target sub-system invokes its own paging mechanism. Clearly, if MN is active in sub-network S_i, an "active paging request" would incur a signaling cost of only PG_i, since S_i would already be aware of the MN's precise cell of attachment. On the other hand, suppose an "idle" paging request is issued for the cells $C_i^1, C_i^2, \cdots, C_i^{\eta}$, where the η cells are indexed in decreasing order of the residence probability. Then, the expected number of paging attempts in that sub-network equals: $E[P(S_i)] = \sum_{j=1}^{|\eta-1|} j \times \varrho_i^j(0) + \eta \times (1 - \sum_{j=1}^{\eta-1} \varrho_i^j(0))$. Accordingly, the expected idle paging cost in S_i is given by $PG_i \times E[P(S_i)]$. Under this constraint (where each sub-system can be issued at most one active and one idle paging request), Figure 6 describes the optimal paging strategy.

1) Compute the unconditional idle residency probability $\varrho_i^j(0)$ by using the paging algorithm presented in Section IV-B.

2) For each sub-system S_i, $i \in \{1, 2, \ldots, N\}$, compute the active residency probability $\varrho_i^{act} = \sum_{j=1}^{|S_i|} \varrho_i^j(1)$. Then, compute the *normalized active residency* factor

$$\hat{\varrho}_i^{act} = \frac{\varrho_i^{act}}{PG_i}.$$

3) For each sub-system S_i, $i \in \{1, 2, \ldots, N\}$, compute the idle residence probability $\varrho_i^{idle} = \sum_{j=1}^{|N_i(0)|} \varrho_i^j(0)$, where $N_i(0)$ denotes the number of cells in S_i (not including the virtual cell C_i^{ϕ}) where the idle residence probability ($\varrho(0)$) is non-zero.

4) Sort the cells in descending order of residence probabilities $C_i^1, C_i^2, \cdots, C_i^l$, and compute the expected paging cost *per sub-network*:

$$E[P_i^{idle}] = PG_i \times \{ \sum_{j=1}^{|N_i(0)-1|} j \times \varrho_i^j(0) + N_i(0) \times (1 - \sum_{j=1}^{N_i(0)-1} \varrho_i^j(0))\}. \quad (11)$$

5) Compute the *normalized idle residency* factor

$$\hat{\varrho}_i^{idle} = \frac{\varrho_i^{idle}}{E[P_i^{idle}]}.$$

6) Issue the active and idle paging requests to different sub-networks in the descending order of the **normalized residency factors**.

Fig. 6. Constrained Paging in De-centralized System

VI. Selective Location Update Heuristic

The decentralized scheme essentially sends each (location, state) update vector to every sub-network, thus guaranteeing that all sub-networks have identical tries. Besides the high location update cost, the scheme also requires the MN to store all update messages during a period of disconnection from the sub-network, so that it can issue delayed updates upon re-attachment. In cases where the period of disconnection is fairly long, the MN might suffer the burden of storing potentially long sequences (per disconnected sub-system). In this section, we investigate an alternative location management heuristic that tries to reduce the location update cost, at the possible expense of additional paging cost.

In our proposed approach, the MN does not necessarily send each update to every sub-network, but selectively updates every sub-network by informing it of the mobile's (MN) profile belonging to that particular sub-network only. On the MN's side, this is equivalent to having the MN run the conventional LeZi-Update algorithm independently for each sub-network. As a consequence of this approach, the decoder trie in any sub-network now stores only a subset of the mobility profiles, corresponding to the movement patterns in that sub-network. From the network perspective, the paging process is, however, not completely independent, but utilizes a central coordinator to improve the efficiency of the paging process. Unlike the centralized scheme of Section IV, this coordinator does not store the detailed update sequence of the MN, *but only periodically refreshes estimates of the paging costs in each individual sub-network.* Every sub-network is thus responsible for intermittently informing this central *shared paging-repository* of its current estimate of the cost for paging the MN within its constituent cells. While initiating a search for the MN, each sub-network will consult this central coordinator to determine the relative paging costs for the MN advertised by other sub-networks, and accordingly decide on whether to page the MN directly within its own cells, or to request another sub-network to perform the paging on its behalf.

A. Proposed Heuristic

The selective location update heuristic is based on the realization that every sub-network is cognizant of the mobile's movement patterns belonging to that particular sub-network only. Thus, none of these sub-networks possesses complete knowledge of MN's mobility profiles. In addition, since each sub-network receives updates only when the MN is in an idle state in that sub-network, this heuristic is unable is exploit any correlation between the MN's call activity state and its movement pattern. Although the location update scheme for each sub-network is independently optimal, this heuristic fails to optimally exploit the correlation between the MN's movement and activity patterns *across sub-networks.* Accordingly, this approach focuses on each of the N independent random processes that are part of the vector-valued stochastic sequence χ used to define Equation (2). Intuitively, this independent operation actually increases the overall location uncertainty of the mobile node. Mathematically, this can be observed from the fact that *conditioning reduces entropy* [8], i.e.,

Result *7: For a set of random variables X_1, X_2, ..., X_n, with distribution $Pr(X_1 = x_1,\, X_2 = x_2,\, \ldots,\, X_n = x_n)$,*

$$\begin{aligned} H(\chi) &= H(X_1, X_2, \ldots, X_n) \qquad (12)\\ &= \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1)\\ &\leq \sum_{i=1}^{n} H(X_i) \end{aligned}$$

The resulting update cost $\sum_{i=1}^{N} H(X_i)$ of this heuristic should thus be larger than the joint-update cost $H(X_1, X_2, \ldots, X_N)$. However, this update cost will usually be smaller than the weighted entropy $H_w(\nu)$ of Equation (5), since the random process ν encodes additional information about the MN's session activity state as well. Clearly, by ignoring the additional state vector $\bar{Z}$, our heuristic update strategy can obtain lower update cost than the centralized optimal strategy, but only at the expense of increased uncertainty. The increased location uncertainty will reflect in higher paging costs, and eventually in larger combined (paging + update) costs as well. To minimize the increase in the paging overhead, each sub-network informs the central coordinator (repository) of its current estimate of the normalized paging cost for that sub-network. We employ a *trigger-based update strategy*, whereby a sub-network informs the central coordinator whenever the current estimate of the normalized paging cost deviates from the previously reported paging cost by a certain fraction.

B. Profile-based Paging on Selective Update

The profile-based paging approach essentially requires each sub-network to independently evaluate the "apparently best" paging sequence. Since each sub-network trie only provides the residence probabilities for cells in that sub-network, the paging algorithm cannot exploit information about the precise movement pattern in other sub-networks and is thus, no longer optimal. The paging process for sub-network S_i essentially consists of the following steps:

i) First, compute the normalized residency factor in its own sub-network, using the Equation (11) to compute $E[P_i^{idle}]$:

$$\hat{\varrho}_i = \frac{\sum_{j=1}^{|N_i|} \varrho_i^j}{PG_i\{\varrho_i^j(0)\sum_{j=1}^{|N_i-1|} j + N_i(1 - \sum_{j=1}^{N_i-1} \varrho_i^j(0))\}}, \qquad (13)$$

where the cells are sorted in decreasing order of residency probabilities. Here N_i denotes the number of cells with non-zero residence probabilities, except C_i^{ϕ}, in S_i, and essentially bounds the maximum number of paging attempts in sub-network S_i.

ii) Consult the central coordinator to determine the set θ of sub-networks (sorted in decreasing order), whose reported normalized residency factors are *greater* than $\hat{\varrho}_i$. If θ is not-empty, issue paging requests successively to each sub-network $j \in \theta$, until the set is exhausted or the MN is successfully contacted.

iii) If the MN cannot be reached by any sub-network in θ, or if θ is empty, page the MN within the current sub-network S_i.

VII. Performance Evaluation

We now present simulation results obtained using a discrete-event simulation framework that we developed for studying the movement of a mobile user in a multi-system environment. A multi-system heterogeneous network topology, with different cell sizes for each sub-network, constitutes the heart of the simulation environment. Synthetic traces of user's activities are dynamically generated and fed in. The mobile user is assumed to have a set of deterministic and statistical patterns reflecting his/her life-style. The activity of the user is broadly classified into mobility and communication events. While mobility is instantiated by *idle* and *move* events, the communication is managed by *session-start* and *session-term* events. In order to consume memory economically, while keeping the scheme fast and efficient, every trie is implemented as a hash-table. The table is periodically flushed, when it becomes full. An open addressing scheme in a double-hashing is used to get random permutations. The hash function is of the form

$$\zeta(\kappa) = \zeta_1(\kappa) + i \times \zeta_2(\kappa)\, mod\, M, \qquad (14)$$

where M is a prime number, i is the number collisions, κ (hash-key) is obtained from any phrase $\bar{w}$ and symbol $\bar{Y}$ using the relation: $\kappa = |\bar{w}| \times |\nu| + |\bar{Y}|$. Further more, $\zeta_1(\kappa)$ and $\zeta_2(\kappa)$ are auxiliary hash functions of the form: $\zeta_1(\kappa) = \lfloor M \times \frac{\kappa}{2^{|\bar{w}|}} \rfloor$, and $\zeta_2 = 2\,[\kappa\; mod\,(M/2+1)] + 1$. Before discussing the simulation results, we describe the various parameters used in our study.

- The $4G$ network is assumed to consist of 3 distinct sub-network types: satellite, PCS and WLAN. The satellite (largest cell size) and PCS networks (intermediate cell sizes) have cellular structure, with each cell having 4–8 neighbors (6 on the average). The 802.11 wireless LANs are generated as multiple (often disconnected) hot-spots, each with its own cellular layout. Assuming that PCS networks have the lowest update cost, the relative update costs for wireless LANs and satellite networks are respectively set to 3 and 10 times that of the PCS network. Similarly, assuming that wireless LANs have the lowest paging cost, the relative paging costs associated for the PCS and satellite networks are set to 4 and 9 times of the wireless LANs.

- The user has a home and a work-place randomly chosen from the multi-system network. This home and work-place both span multiple cells (in different sub-networks). The time spent by the user at every cell is normally distributed. The movement pattern of the user is assumed to differ between weekdays and weekends/holidays.
- The call arrival process is Markov-modulated, with three distinct states, each with its own Poisson arrival rate λ and normally distributed holding holding times with mean μ and variance σ. The states represent weekday daytimes ($\lambda = 0.2$ calls/hr, $\mu = 10$ min, $\sigma = 3$ min), weekday evenings ($\lambda = 0.3$ calls/hr, $\mu = 20$ min, $\sigma = 5$ min), and weekends ($\lambda = 0.5$ calls/hr, $\mu = 30$ min, $\sigma = 7$ min).
- The entire simulation results are based on an observation for a period of 12 weeks.

We present the results of a typical study to compare the *update* and *paging* costs for the three strategies – centralized, distributed and heuristic – presented earlier. For comparison, we also studied the performance of an independent location management strategy (each sub-network operated independently) with movement-based updates and cluster paging. For cellular PCS networks, it has been demonstrated in [3] that, as the movement threshold (the number of moves between updates) increases, the update cost decreases but the paging cost increases. More precisely, using an average threshold of 3 was seen to achieve a reasonably fair compromise. Thus, in our experiments we have used a movement threshold of 1 for satellite-networks (highest level of hierarchy), 3 for cellular PCS networks (middle level) and 5 for wireless LANs (lowest level of hierarchy).

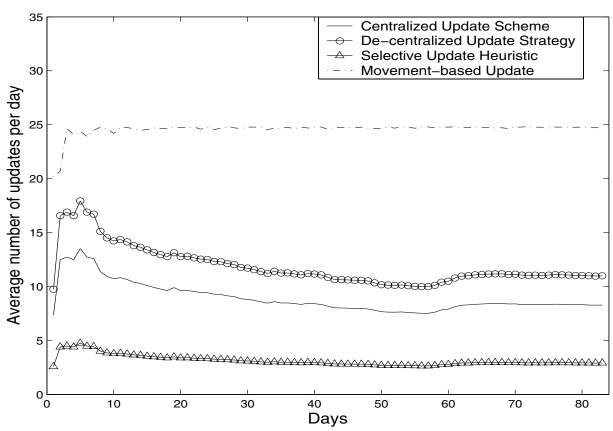

Fig. 7. Number of Daily Updates with Different Strategies

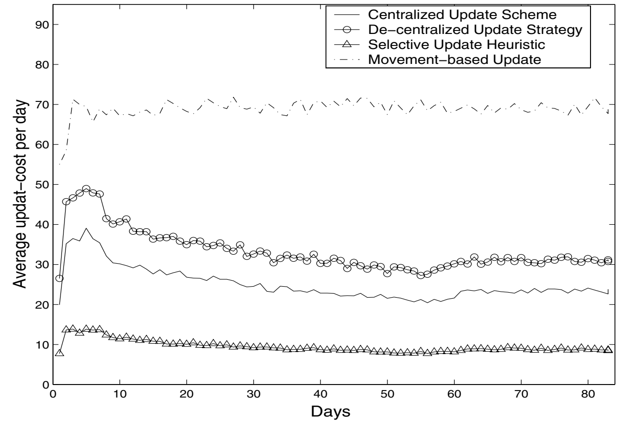

Fig. 8. Comparison of Update Cost in Multi-System Networks

Figure 7 shows that all the three proposed location management schemes exhibit significant performance gains over the independent movement-based update strategy. While the proposed selective location management heuristic results in lowest number of updates (almost $\frac{1}{8}$ of movement-based updates), the centralized and de-centralized schemes also achieve a savings of 50%–60% in the average *number* of daily updates. Figure 8 provides a comparison of daily average update *cost* in such a multi-system environment, where the updates are weighed with the corresponding location update cost. The centralized, de-centralized and selective update schemes respectively incur $\sim \frac{2}{7}$, $\sim \frac{3}{7}$ and $\sim \frac{1}{7}$ of the average daily update costs of the movement-based strategy. Note that the heuristic update strategy (where each sub-network operates its independent LeZi-Update algorithm) has the lowest update cost because both the centralized and decentralized mechanisms additionally consider the call activity vector, which the heuristic mechanism does not. In our simulations, the randomness of the call arrival patterns causes the centralized scheme to generate more updates – for other more predictable call patterns, the update cost of the heuristic strategy may however be higher than that of the centralized scheme.

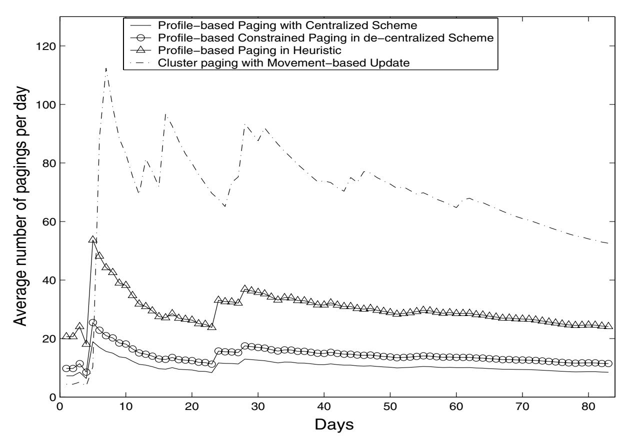

Fig. 9. Number of Daily Paging with Different Schemes

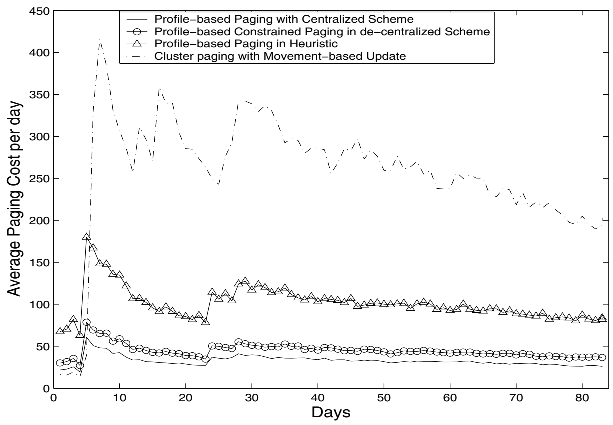

Fig. 10. Comparison of Paging Cost in Multi-System Networks

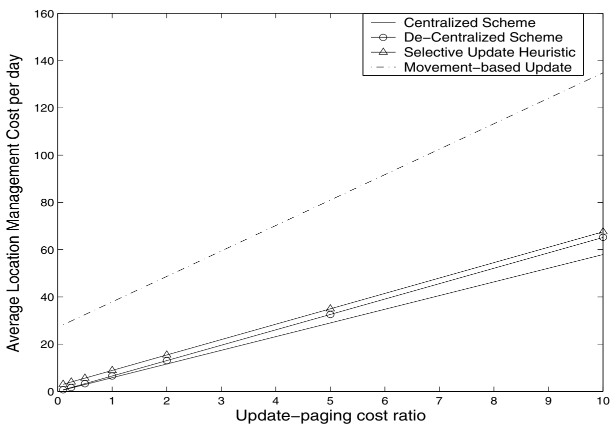

Fig. 11. Location Management Cost with Update:Paging Ratio

Figure 9 demonstrates the advantage of the profile-based probabilistic paging schemes over the existing cluster paging approach. The last known position of the mobile and its neighborhood (where the neighborhood is equal to the respective movement threshold) is searched and paged simultaneously in this cluster paging scheme. The centralized scheme generates the lowest *number* of paging messages ($\leq \frac{1}{7}$ of cluster paging), followed by the de-centralized scheme ($\approx \frac{1}{5}$ of cluster paging) and the heuristic ($\approx \frac{1}{3}$ of cluster paging). Figure 10 shows a similar trend, when we weigh the paging messages by their corresponding cost to compute the overall system paging *cost*. By failing to completely utilize the correlation in the MN's movement pattern across different sub-networks, the heuristic strategy incurs substantially higher paging overhead than the centralized strategy,

To compare the relative gains of all these schemes, we have compared their performance in terms of *total location management cost* for different values of the location update:paging cost ratio. Figure 11 shows that the centralized scheme offers the lowest combined (paging + location update) location management cost, while the de-centralized and the heuristic schemes allow us to trade off between between the update and paging costs. However, all the three schemes result in significant savings over the existing independent location management strategy (movement-based update and cluster paging).

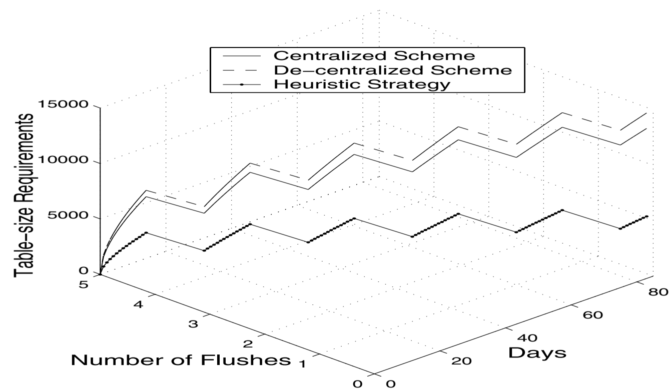

Fig. 12. Table-Size for Proposed Location Management Schemes

Figure 12 compares the evolution of the table-size at the MN's encoder for the three proposed strategies over the entire entire simulation period, and for different number of flushes. Since the selective location update heuristic does not consider the activity state of the MN at all, and also creates an independent trie for each sub-network, it uses the smallest amount of memory. More importantly, even in this multi-system heterogeneous wireless environment, the table sizes for the optimal centralized and de-centralized strategies are bounded by 12–15 Kbytes. The memory requirement is even lower than this amount when the table is flushed for $\sim$ 4-5 times in the entire period of 12 weeks. Such a memory requirement appears to be quite reasonable for future-generation mobile devices.

VIII. Conclusions

In this paper we have proposed an information-theoretic location management strategy for emerging multi-system $4G$ networks. The key benefit of our framework is that it exploits the correlation in movement patterns between individual sub-networks, without assuming the existence of a centralized or publicly available topology database. To exploit the fact that the mobile can be simultaneously active and passive in different sub-networks, we identified the need for combining the MN's activity state with its movement pattern in the location prediction algorithm. We have also developed the concept of *weighted entropy*, as the most fair measure of location uncertainty of the mobile in heterogeneous sub-networks. Based on the LZ-compression algorithm, we identified three alternative approaches for mobility management, each with differing levels of coordination, and update strategies. Among these approaches, the centralized approach has the lowest overall cost, but can lead to processing bottlenecks. In contrast, the de-centralized approach incurs slightly higher costs in both paging and location update. As a practical alternative, the heuristic approach is fairly promising, since it significantly reduces the locate update load on the MN (which is significantly energy-constrained). Simulation results show that the memory overhead on the MN for each of these alternatives is also fairly reasonable. For future work, we shall address the problem of paging under bounded latencies, and the use of joint residency distributions for deriving better paging sequences. We are also currently investigating the use of lossy compression techniques in the update process, to provide reasonable performance even when some update messages are not correctly delivered.

ACKNOWLEDGEMENTS

The work of Abhishek Roy and Sajal K. Das is supported by NSF grants # IIS-0121297, # IIS-0326505 and Texas Telecommunication Engineering Consortium (TXTEC).

REFERENCES

[1] I. F. Akyildiz and W. Wang, "A Dynamic Location Management Scheme for Next-Generation Multitier PCS Systems", *IEEE Transactions on Wireless Communications*, vol. 1, no. 1, pp. 178-189, Jan 2002.

[2] A. Bar-Noy and I. Kessler, "Tracking Mobile Users in Wireless Communication Networks", *IEEE/ACM Trans. on Information Theory*, vol. 39, no. 6, pp. 1877-1886, Nov. 1993.

[3] A. Bar-Noy, I. Kessler and M. Sidi "Mobile Users: To Update or not to Update", *Wireless Networks*, vol. 1, no. 2, pp. 175-185, July 1995.

[4] R. Berezdivin, R. Breinig and R. Topp, "Next-Generation Wireless Communications Concepts and Technologies", *IEEE Communications Magazine*, vol. 40, no. 3, pp. 108-116, March 2002.

[5] A. Bhattacharya and S. K. Das, "LeZi-Update: An Information-theoretic approach for personal mobility tracking in PCS networks," *ACM-Kluwer Wireless Networks* (WINET), vol. 8, no.2, pp.121-137, 2002.

[6] Y. Birk and Y. Nachman, "User Direction and Elapsed-time Information to Reduce the Wireless Cost of Locating Mobile Users in Cellular Networks", *Wireless Networks*, vol. 1, no. 4, pp. 403-412, Dec 1995.

[7] E. Cayirci and I. F. Akyildiz, "User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems", *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, pp. 236-246, July-Sep 2003.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley, 1991.

[9] M. Feder, N. Merhav and M. Gutman, "Universal prediction of individual sequences," *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1258-1270, July 1992.

[10] J. N. Kapur and H. K. Kesavan, *Entropy Optimization Principles with Applications*, Academic Press, 1992.

[11] G. P. Pollini and C.-L. I, "A Profile-based Location Strategy and its Performance, *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 8, pp. 1415-1424, 1997.

[12] C. Rose and R. Yates, "Minimizing the Average Cost of Paging under Delay Constraints", *Wireless Networks*, vol. 1, no. 2, pp. 211-219, July 1995.

[13] W. Spankowski, "Asymptotic Properties of Data Compression and Suffix Trees", *IEEE Trans. on Information Theory*, vol.39, no. 5, pp. 1647-1659, Sept. 1993.

[14] J. H Sun, D. Howie and J. Sauvola, "Mobility Management Techniques for Next Generation Wireless Networks", *Proc. of SPIE, Wireless and Mobile Communications*, vol. 4586, pp. 155-166, Oct. 2001.

[15] W. Wang and I. F. Akyildiz, "A Cost-Efficient Signalling Protocol for Mobility Application Part (MAP) in IMT-2000 Systems", *Proc. of ACM 7th Annual Intl. Conf. on Mobile Computing and Networking* (MobiCom), pp. 345-355, 2001.

[16] G. Wu and M. Mizuno , "MIRAI Architecture for Heterogeneous Network", *IEEE Communications Magazine*, vol. 40, no. 2, pp. 126-134, Feb 2002.

[17] T. B. Zaharidis, K. G. Vaxevanakis, C. P. Tsantilas and N. A. Zervos, "Global Roaming in Next-Generation Networks", *IEEE Communications Magazine*, vol. 40, no. 2, pp. 145-151, Feb 2002.

[18] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp.530-536, September 1978.