A Dynamic Location Management Scheme for Wirless Networks Using Cascaded Correlation Neural Network

Amar Pratap Singh.J and Karnan.M

Abstract—**In this paper we propose a novel intelligent technique for reducing the location update and paging cost. This paper focuses on the location management procedure, which is the process that allows the network to identify the exact location of a mobile terminal (MT) for a call delivery. Location management occurs in two stages: locations updates and call delivery. When a mobile terminal moves from one location area to another it performs a location update procedure that provides the network with its location information. On the other hand, call delivery means that the network is queried for location information on the called MT. In order to actually deliver the call, the network needs to identify the exact location of the called MT using a paging procedure. In this paper we present a user pattern learning strategy using intelligent algorithm called Cascaded Correlation Neural Network to find the called MT's current cell, within its registered Location Area, in the most accurate and efficient way possible. We compare the proposed algorithm with cayrici's strategy and standard UMTS procedure. The search cost and update cost are used for performance analysis. The experimental results show that the proposed technique is more efficient than the other existing techniques.**

Index Terms—**Cascaded Correlation Neural Network, Location, Mobile terminal, Mobile User, Paging, Update.**

I. Introduction

Diverse mobile services and development in wireless networks have stimulated an enormous number of people to employ mobile devices such as cellular phones and portable laptops as their communications means. In a mobile system the user must be able to access the services while roaming from one location to another location. The major feature of wireless networks is mobility support, which enables mobile users to communicate with others regardless of location. It is also the very source of many challenging issues, relating to the mobility and service patterns of mobile terminals (MTs), namely, user mobility profile (UMP). For each mobile user, a UMP consists of detailed information of service requirements and mobility models that is essential to quality of service (QoS) and roaming support. User mobility profile (UMP) is a combination of historic records and predictive patterns of mobile terminals, which serve as fundamental information for mobility management and enhancement of quality of service (QoS) in wireless multimedia networks.

In second generation mobile communication system such as thc Global System for Mobile Communications (GSM), location tracking (paging) [6] [7] is achieved as follows. A network is divided into geographical areas, called Location Areas (LA) and the location management system keeps track of the current LA of an mobile host (MH). An LA may contain one or more cells. The location information is stored in the network database for location management. To maintain the consistency of location information, an update process is triggered whenever an MH crosses LA boundaries. When a call arrives, a query to the location database is conducted 10 obtain the location information of the called MH. Then the location management system pages all cells in the corresponding LA simultaneously.

Since Mobile hosts are fast growing, the load will be too heavy to handle the location management, especially in densely populated areas. There are several methods for location management: they are categorized into two one is memory based and the other is memory less. Memory based system relies on user's statistics to reflect the users behavior. In mobile networks, most users pursue standard routines during business hours, residing mostly at their place of work. For these users, it is possible to predict with significant accuracy their location at a particular time of day. With the help of user pattern learning location prediction schemes location and paging requests can be reduced to a considerable amount. Mobile users can be classified into three different categories namely: Predictable User, Expected User and Random User, depending on the predictability of their daily routine: users who have a very high probability of being where the system expects them to be (Predictable User), users who have a certain likelihood of being where the system expects them to be (Expected User), and users whose position at a given moment is unpredictable (Random User).

The idea is, the user pattern learning strategy links with each user a list of LA's where the mobile user is most likely to be in a particular time interval. So, when a call arrives the location with in the list is being paged first in a sequence, until the MT is found. When a user moves within the list no update is required. The list is stored at an intermediate location database (ILD) associated with a Mobile Switching Centers (MSC) as well as within the user's MT If the user follows the expected behavior then the cost for location update will be reduced. The proposed scheme reduces the control-signaling traffic due to location management by several orders of magnitude comparing to the time-based [9]

Amar Pratap Singh.J is with the Research Scholar, Computer Science and Engineering, nna University,Coimbatore(e-mail: japsindia@yahoo.com).
Karnan.M is the Prof & Head, Computer Science and Engineering,Tamil Nadu College of Engineering, imbatore (email:kamanme@yahoo.com).

[13] and movement-based [6] [9] [12] [7] location update, and the blanket, selective, and velocity [9] paging schemes.

In our strategy we make use of an intelligent network to learn about the various users. The scheme requires to phases: knowledge representation and application. First the network is trained with the training examples. Then it is subjected to application.

The ability to learn is a fundamental trait of intelligence. Although a precise definition of learning is difficult to formulate, a learning process in the neural network context can be viewed as the problem of updating network architecture and connection weights (an elementary structure and functional unit between two neurons) so that a network can efficiently perform a specific task. The network usually must learn the connection weights from available training patterns. Performance is improved over time by iteratively updating the weights in the network. Neural network's ability to automatically learn from examples makes them attractive. Instead of following a set of rules specified by human experts, Neural network's appear to learn underlying rules (like input-output relationships) from the given collection of representative examples. This is one of the major advantages of neural networks over traditional expert systems. Neural networks derive their computing power through their ability to learn and then generalize; generalization refers to the ability of the neural network to produce reasonable outputs for inputs not encountered during training. It is this quality that we utilize to predict the movement of mobile users so that we can predict the position of a user in advance and reduce the paging cost based on the predicted destination cell.

The rest of this paper is organized as follows: In Section 2, we introduce the User Pattern Learning Strategy. Then, in Section 3, describes the proposed cascaded correlation algorithm and in section 4 cascaded correlation neural network architecture. Section 5 describes the simulation results and the different scenarios used for cost comparison, finally, the conclusion in Section 6.

II. User Pattern Learning Strategy

Mobile subscribers usually follow a limited number of mobility patterns in their daily lives. For example, people generally take almost the same path and same time to go to work every day. In the UPL scheme, the data related to these patterns is stored in an intermediate database called intermediate location database. This database is located in the same level as Mobile Switching Center (MSC) is placed.

A UPL is a list of cells expected to be visited starting from a given time according to the mobility history of a mobile. The main motivation to use an intelligent system is their ability to learn relationships with complex data that cannot be made by humans. The basic element of an neural network (NN) system is called a neuron. The neuron accepts one input x, which may actually be a sum of multiple inputs, and produces an output value y based on a nonlinear function F(x) [4], [8], [10]. In general, NN systems are capable of "learning" trends in a given data set and establishing input-output relationships based strictly on a "test" set of data. It is desirable for the "test" data that the system "learns" from to be as representative of the complete data set as possible; trends not seen in the test data set will not be "learned" by the neural network system [5]. For example, one may go to work passing through a cell in the morning. The user may come back home passing through the same cell in the evening. Moreover, the same person may go somewhere else through the same cell at noon in the weekends.

Cascaded correlation neural network (CCNN), is similar to an ANN system. CCNN is organized into several layers and each layer is organized with elementary units called neurons. The learning phase of an CCNN is based on algorithms (e.g., backpropagation) able to set the weight connections by training the net with a known data set until a certain (small) error is achieved.

The main advantage of using CCNN over ANN is as follows,

- CCNN is much faster than the ANN
- Design of ANN is not an easy task because the number of hidden layers and the number of neurons in the hidden layers should be specified in advance. In CCNN the network will automatically add the hidden layer based on the network performance.
- In CCNN, the input layer is connected to the hidden layer and also to the output layer which makes the learning process more effective.

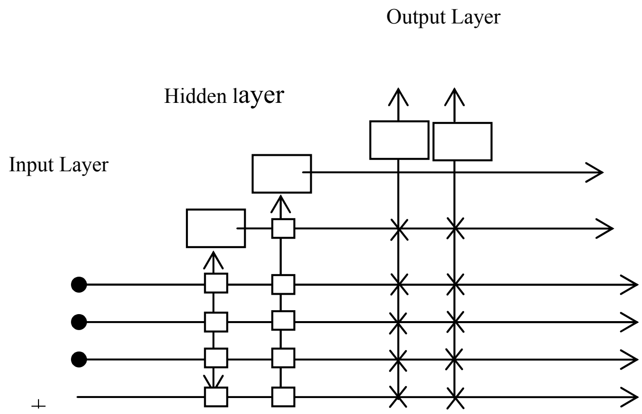

Figure 1. Cascaded Correlation Architecture Model

The architecture of CCNN is shown in fig 1. The number of nodes used in the input layer usually depends on the type and amount of input data; several hundred input nodes may be used in large applications.The number of nodes in the hidden layer determines, in general, the ability of the network to learn complex relationships. There may be multiple hidden layers to increase the network's ability to learn. In our model, with the BP algorithm, there are three layers in the Neural Networks, input layer, hidden layer, and output layer. The role of the hidden layer is to remap the inputs and results of previous layers to achieve a more separable or classifiable representation of the data and allow attachment of semantics to certain combinations of layer inputs.

CCNNs perform their calculations using nonlinear functions and simple multiplying factors, called weights, which are associated with a pathway between any two nodes. While the functions remain constant for any given application, the weights are updated in such a manner that the complete network "learns" to produce a specific output for a specific input. The process of adjusting the weights to achieve a specified accuracy level is referred to as "training." The backpropagation (BP) training algorithm is a method for iteratively adjusting the weighting factors until the desired accuracy level is achieved. This algorithm is based on a gradient-search optimization method applied to an error function (i.e., the sum of squared error). In our approach, we use the learning process to derive a list from which we can find, with high accuracy, the exact cell in which the MT resides at any time of every day. The learning process is able to derive such a list after observing (learning) the behavior of a mobile user for a certain period of time. By observing the mobile user's daily behavior, we use the BP algorithm to learn the behavior. With some useful data from observation of the mobile user as the input nodes, we can obtain the output as the result we want, which is the cell information of the mobile user on observation, that is to say, the cell list for a mobile user. For every mobile user there is a user pattern learning process associated to it. We may classify the users into three different categories depending the predictability of their daily routine: users who have a very high probability of being where the system expects them to be (Predictable Users), users who have a certain likelihood of being where the system expects them to be (expected users), and users whose position at a given moment is unpredictable (random users), similar to the classification proposed in [12]. The certainty of predictable and Expected users can be used by the system to reduce the number of location update operations. So, after the learning process completes, we get the mobile user's behavior associated with known location areas. Then, a profile is built for the mobile user table 1 When a call arrives for a mobile, it is paged sequentially in each location within the list. When a user moves between location areas in this list, no location updates are required. The list is stored at the HLR in the information database (ID) as well as in the user's mobile terminal. The cost reduction depends on the behavior of each class of user. It can be assumed that, when the user follows its expected behavior, the location update cost is reduced, even if accesses to HLR are minimized when calls are received from relatively close areas.

Our strategy increases the intelligence of the location update procedure and utilizes replication and locality to reduce the cost incurred from the paging procedure. An Intermediate Location Database (ILD) is added to the UPL scheme. This database is located on the same architectural level as the MSC and contains the profile of each user. In Table 1, we show an example UPL training examples.

TABLE I. AN EXAMPLE TRAINING SET.

Examples	Day	Time	Cell Id	Probability(%)
E1	Monday	02.15	5,1,6	90,50,10
E2	Monday	11.45	5,2,1	95,50,10
E3	Sunday	12.00	5,6,3	40,90,15
E4	Thursday	14.15	5,2,1	95,50,10
E5	Thursday	05.30	5,1,6	90,50,10

In the above user profile behavior the first field is the example number, there can be an number of examples. The next to fields are the day and time when a mobile terminal has entered a LA. The next field is the cell ID, where a mobile terminal is expected to be and the next filed is the probability of the mobile terminal to be in a particular LA

III. CASCADED CORRELATION ALGORITHM

1. Initial configuration: The algorithm begins with a simple perceptron with N input units and M output units. The number of inputs (N) and outputs (M) is dictated by the problem and by the I/O representation the experimenter has chosen. Every input is connected to every output by a connection with an adjustable weight. There is also a bias input, permanently set to +1. The output units just reproduce a linear sum of their weighted inputs, or they may employ some non-linear activation function. In the experiments we have run so far, we use symmetric sigmoidal activation function (hyperbolic tangent) whose output range is -1.0 to +1.0.

2. Initial training: The learning algorithm starts with no hidden units. The direst input-output are trained as well as possible over the entire training set. The perceptron is trained on the entire training set {(Vp,Tp) | p = 1, . . . , P }, until the performance of the network is as good as possible. If the desired performance is obtained, the algorithm stops. Otherwise: Start adding hidden units to the network, one by one.

3. Training of candidates: A pool of candidates for a new hidden unit is generated. This pool emulates a stochastic search in the weight space, which will decrease the risk of inserting a candidate stranded in a local minimum with high error. Each node in the pool of candidates is connected to all input nodes and all previously inserted hidden units. Each of the candidates is trained with the purpose of maximizing some measure of"goodness" of the candidate.

4. Inserting a new hidden unit: To create a new hidden unit, we begin with a candidate unit that receives trainable input connections from all of the network's external input and from all pre-existing hidden units. The candidate with the highest score is inserted "for real" in the network as a new hidden unit. The incoming weights to the new hidden unit are then frozen, i.e. they are not to be changed anymore. The new hidden unit is connected to all output nodes with random weights.

5. Retraining the network: All the incoming weights to the output units are retrained in order to adjust the weights from the newly inserted hidden unit. If the performance of the network is satisfied after retraining, the algorithm stops. Otherwise: Go to 3.

IV. CASCADED CORRELATION NEURAL NETWORK ARCHITECTURE

Cascade correlation network starts with a minimal topology, consisting only of the required input and output units (and a bias input that is always equals to 1). This net is trained until no further improvement is obtained. The error for each output until is then computed (summed over all training patterns). Next, one hidden unit is added to the net in a two-step process. During the first step, a candidate unit is connected to each of the input units, but is not connected to the output units.

The weights on the connections from the input units to the candidate unit are adjusted to maximize the correlation between the candidate's output and the residual error at the output units. The residual error is the difference between the target and the computed output, multiplied by the derivative of the output unit's activation function, i.e., the quantity that would be propagated back from the output units in the back propagation algorithm. When this training is completed, the weights are frozen and the candidate unit becomes a hidden unit in the net. The second step in which the new unit is added to the net now begins.

The new hidden unit is then connected to the output units, and the weights on the connections being adjustable. Now all connections to the output units are trained. (Here the connections from the input units are trained again, and the new connections from the hidden unit are trained for the first time.) A second hidden unit is then added using the same process. However, this unit receives an input signal from the both input units and the previous hidden unit. All weights on these connections are adjusted and then frozen. The connections to the output units are then established and trained.

The process of adding a new unit, training its weights from the input units and the previously added hidden units, and then freezing the weights, followed by training all connections to the output units, is continued until the error reaches an acceptable level or the maximum number of epochs (or hidden units) is reached.

The purpose of inserting a new unit is to reduce the total error of the network. The way the CCA does this is to train the candidate unit so the correlation between the residual error and the output from the candidate is maximized. Let X and Y be two stochastic variables.

Then the correlation between X and Y is defined as:

$$corr[x,y] = \mathrm{cov}[x,y] / \sqrt{\mathrm{var}(x)\,\mathrm{var}(y)} \quad (1)$$

V. Simulation Results

A. Analytical Model

We assume that link costs and database access costs are defined by message transmission delays and updating or query delays, respectively. For each mobile terminal, we define the following quantities:

A: average number of calls (i.e., voice or data) to a target MT per time unit;

fi: average number of times the user changes LA per time unit;

UT: average total cost of the location update procedure;

ST: average total cost of the location search procedure; and

CG: average total cost per time unit for the location update and the location search.

The total cost per time unit for the location search and location update procedures of the proposed scheme is

$$C_G = \mu U_T + \lambda S_T$$

B. Cost Comparison

Cayirci and Akyildiz [15] proposed a strategy for location management, which we call Cayirci, while the GPRS/ UMTS standard proposed another strategy, which we call UMTS, for solving the same problem.

To be able to compare our strategy to UMTS, we have to compute the costs for the location update and location search operations. We define the following costs for the UMTS location management procedure:

S: cost for a location update operation according to the GPRS/UMTS standard;

SUMTS: cost for a location search operation according to the GPRS/UMTS standard; and

CUMTS: total cost per time unit for the location search and location update operations.

The total cost per time unit for the location search and location update procedures is given by $C_{UMTS} = \mu U_{UMTS} + \eta S_{UMTS}$

We define the relative cost of the proposed scheme as the ratio of the total cost of our scheme (per time unit) on the total cost of the standard UMTS procedures. Furthermore, this relative cost is a function of the call-to-mobility ratio (CMR):

$$\frac{C_G}{C_{UMTS}} = \frac{\mu U_T + \lambda S_T}{\mu U_{UMTS} + \lambda S_{UMTS}}$$

$$\frac{C_G}{C_{UMTS}} = \mu\left(\frac{U_T + (\lambda/\mu) S_T}{U_{UMTS} + (\lambda/\mu) S_{UMTS}}\right)$$

Where $CMR = \lambda / \mu$

The method proposed by Cayirci and Akyildiz [3] is based on a profile similar to the one used in our scheme. There are some differences between the two, but they are mainly structural differences. For example, the short-term events leading to registration are not reflected as they are in our scheme. Furthermore, our profile is more likely to find the user in fewer trials due to the "next nodes" field that provides information on the next visited areas. Both factors compensate each other. Another important difference is the fact that [3] sets up a list of cells where no updates are performed while the user roams within this set of cells. Otherwise, a new record is created and another classical location update method is used (i.e., IS-41 or standard GPRS/UMTS).

The total cost per time unit for the location search and location update procedures is given by

$$C_{Cayirci} = \mu U_{Cayirci} + \eta S_{Cayirci}$$

As we did for the UMTS standard, we define the relative cost of our scheme compared to Cayirci's scheme as:

$$\frac{C_G}{C_{Cayirci}} = \frac{\mu U_T + \lambda S_T}{\mu U_{Cayirci} + \lambda S_{Cayirci}}$$

$$\frac{C_G}{C_{Cayirci}} = \mu\left(\frac{U_T + (\lambda/\mu) S_T}{U_{Cayirci} + (\lambda/\mu) S_{Cayirci}}\right)$$

Where $CMR = \lambda / \mu$

C. Numerical Results

In this section, we present the numerical results from the performance evaluation of the user pattern learning strategy we proposed and we compare those results with other strategies, namely, Cayirci's strategy and the UMTS location management standard. We define several parameters that will be used for the different simulation scenarios: K is the probability that an MT will be found roaming under one of its likely areas as registered in the profile of a given LSTP (as an average). It will be close to 1 for Predictable users and will vary between 0.5 and 0.8 for Expected users. Random users cannot be assigned a list and, thus, their values are below 0.5, n is the average number of tries needed to page an MT from the ILD where it is roaming in. Its value depends on the probability distribution of the user among the areas in the list and the size of the list itself. Uniform, linear, and exponential models are discussed in [15]. For each of them, an average value of n is given as a function of the size of the list, k is the number of location data tables that must be updated each time the MT changes LSTP. k normally ranges from 4 to 12. If its value is 0, it indicates that no replication (i.e., location data tables) is used. CMR is the independent variable in the simulations as in [15]. All users are classified according to their call-to-mobility ratio (CMR).

Table 2 shows the comparison of relative cost for various CMR values for proposed scheme with Cayirci’s strategy and proposed scheme with UMTS procedure.

TABLE II. RELATIVE COST COMPARISON

Sl.No	**CMR**	**Relative cost with Cayirci’s strategy**	**Relative cost with UMTS Procedure**
1	1	0.4623	0.6097
2	2	0.5121	0.7697
3	3	0.5404	0.8821
4	4	0.5586	0.9654
5	5	0.5713	1.0295
6	6	0.5807	1.0805
7	7	0.5879	1.1219
8	8	0.5936	1.1563
9	9	0.5983	1.1853
10	10	0.6021	1.2100

VI. CONCLUSION

In this paper, we present a user pattern learning strategy using Cascaded Correlation Neural network to reduce location update signaling cost by increasing the intelligence of the location procedure in UMTS. This strategy associates to each user a list of cells where she is likely to be with a given probability in each time interval. The list is ranked from the most likely to the least likely place where a user may be found. When a call arrives for a mobile, it is paged sequentially in each location within the list. When a user moves between location areas in the list, no location updates are required. The results obtained from our performance evaluation confirm the efficiency and the effectiveness of UPL in comparison with the UMTS standard and other well-known strategy. This improvement represents a large reduction in location update and paging signaling costs.

REFERENCE

[1] Akyildiz I.F, Ho J.S.M, and Lin.Y.B : Movement-Based Location Update and Selective Paging for PCS Network IEEE/ACM Trans. *Networking,* vol. 4, pp. 629-638, Aug. 1996.

[2] Bar-Noy.A, Kessler.I, and Sidi.M : Mobile Users: To Update or not to Update, ACM/Baltzer J. *Wireless Networks*, vol. 1, no. 2, pp. 175-186, July 1995

[3] Cayirci.E and Akyildiz.I.F: User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems IEEE *Trans. Mobile Computing,* vol. 1, no. 3, pp. 236-247, July-Sept. 2002.

[4] Go.J, Han.G, Kim.H, and Lee.C, Multigradient: A New Neural Network Learning Algorithm for Pattern Classification IEEE *Trans. Geoscience and Remote Sensing,* vol. 39, no. 5, pp. 986-993, 2001.

[5] Halpin.S and Burch.R: Applicability of Neural Networks to Industrial and Commercial Power Systems: A Tutorial Overview IEEE *Trans. Industry Applications,* vol. 33, no. 5, pp. 1355-1361, 1997

[6] Hang-Wen Hwang, Ming-Feng Chang and Chien-Chao Tseng: A Direction-Based Location Update Scheme with a Line-Paging Strategy for PCS Networks IEEE *Communication Letters*, Vol.4, No.5, pp. 149-151, May 2000. 8, pp. 75-83, 1997.

[7] Ho.J.S.M and Xu.J : History-Based Location Tracking for Personal Communications Networks Proc. IEEE VTC ’98, pp. 244-248, May 1998

[8] Hu.W and Tan.T: A Hierarchical Self-Organizing Approach for Learning the Patterns of Motion Trajectories IEEE *trans. Neural Networks,* vol. 15, no. 1, pp. 135-144, 2004.

[9] Jie Li, Yi Pan and Xiaohua Jia,: Analysis of Dynamic Location Management for PCS Networks, IEEE transaction on *Vehicular Technology*, Vol. 51, No.5 pp. 1109-1119, September 2002.

[10] Lee.C and Landgrebe.D: Decision Boundary Feature Extraction for Neural Networks IEEE *Trans. Neural Networks,* vol. 8, pp 77-83, 1997.

[11] Pollini.G and Chih-Lin.I: A Profile-Based Location Strategy and Its Performance IEEE J. *Selected Areas In Comm.,* vol. 15, no. 8, pp. 1415-1424, 1997.

[12] Li.J, Kameda.H, and Li.K : Optimal Dynamic Mobility Management for PCS Networks IEEE/ACM Trans. *Networking*, vol. 8, pp. 319-327, June 2000.

[13] Rose.C: Minimizing the Average Cost of Paging and Registration: A Timer-Based Method, ACM/Baltzer J. *Wireless Networks*, vol. 2, no. 2, pp. 109-116, June 1996.

[14] Rose C: State-Based Paging/Registration: A Greedy Technique IEEE Trans. *Vehicular Technology,* vol. 48, no. 1, pp. 166-173,1999.

[15] Yuen W.H.A and Wong W.S: A Contention-Free Mobility Management Scheme Based on Probabilistic Paging IEEE Trans. *Vehicular Technology*, vol. 50, no. 1, pp. 48-58, 2001