Detecting Location Patterns In Mobile Environment Based On User Movements

Mr. Mohammad Waseem, Mrs. R. R. Shelke

***Abstract*—This electronic Now a day's user request various kinds of services by mobile devices at anytime and anywhere, due to the advancements in wireless and web technologies. Thus making the information effectively available to the users is an important issue in the mobile computing systems. Detecting the user behavior can highly benefit the enhancements on system performance and quality of services. Based on changeable user location behavior patterns, mobile service systems have the capability of effectively mining a special request from abundant data. In this paper user location behavior patterns, are studied in terms of the problem of mining matching mobile access patterns based on joining the following four kinds of characteristics, L, T, and S, where U is the mobile user, L is the location, T is the dwell time in the timestamp, and Sis the service request. By introducing standard graph-matching algorithms along with the primitives of a database management system, which comprises grouping, sorting, and joining, these joint operations are defined. Finally, performance studies are conducted to show that, in terms of execution efficiency and scalability, the proposed procedures produced excellent performance results.**

***Index Terms*—mobile access patterns, standard graph matching algorithm, mobile computing.**

I. INTRODUCTION

Mobility management in mobile computing environments covers the methods for storing and updating the location information of mobile users who are served by the system. A hot topic in mobility management research field is mobility prediction. Mobility prediction can be defined as the prediction of a mobile users next movement where the mobile user is traveling between the cells of a PCS (Personal Communication Systems) or GSM network. The predicted movement can then be used to increase the efficiency of PCSs. In this paper, we propose a new algorithm for predicting the locations of a mobile user in a Personal Communication Systems network. In our algorithm, user mobility patterns are mined from the history of mobile user trajectories, mobility rules are extracted from these patterns, and then mobility predictions are accomplished by using these rules. Effective allocation of resources to mobile users would improve resource utilization and reduce the latency in accessing the resources.

Mr. Mohammad Waseem is with HVPM College of Engg. & Tech, Amravati, India
Mrs.R.R.Shelke. B. is with HVPM College of Engg. & Tech, Amravati, India

Broadcast program generation can also benefit from predicted mobility patterns, since the data items can be broadcast to the cell where the users are moving.

Accurate prediction of location information is also crucial in processing location-dependent queries of mobile users. When a user submits a location-dependent query, the answer to the query will depend on the current location of the user. Many application areas including health care, bioscience, hotel management, and the military benefit from efficient processing of location-dependent queries. With effective prediction of location, it may also be possible to answer the queries that refer to the future positions of users.

II. PROBLEM DEFINITION

In our work, we assume that the mobile users move in a wireless PCS network, which has architecture similar to those used in EIA/TIA IS41and GSM standards[12]. The coverage area of the PCS network is partitioned into smaller areas which are called cells. In each cell in the PCS network, there is a base station (BS) which has the capability of broadcasting and receiving information. The base stations are connected to each other via a fixed wired network. Mobile users use radio channels to communicate with base stations. The coverage area consists of a number of location areas. Each location area may consist of one or more cells but in our work we assume that each location area consists of only one cell. Base stations regularly broadcast the ID of the cell in which they are located. Therefore, the mobile users which are currently in this cell and listening to the broadcast channel will know in which cell they are now.

The movement of a mobile user from his current cell to another cell will be re-corded in a database which is called home location register(HLR). In addition, every base station keeps a database in which the profiles of the users located in this cell are recorded. This database is called visitor location register(VLR). Therefore, in our system it is possible to get the movement history of a mobile user from the logs on its home location register. Since mobile users may initiate calls to other users or receive incoming calls while moving in the coverage region, the ongoing calls should be transferred from one cell to another without call dropping. To avoid call dropping due to insufficient resources at the destination cell, apriori resource allocation could be employed at that cell. In our work, we collect the movement trajectories of a user in the form of T=h(id1 ,t1), (id2,t2),.. ., (idk,tk)i. Here id1denotes the ID number of the cell to which the user enters at time t1. In this record it is clear that two consecutive ID numbers must be the ID numbers of two neighbor cells in the network. After the movement history of a user is collected in a predefined time interval in the above format, this record is partitioned into subsequences. This procedure is accomplished as follows: If the mobile user stays in a cell idi more than a threshold value, before moving to another one idi+1atti+1, we assume that his trajectory up until now hid1,...,idiiends here, and at idi+1a new trajectory is started. Therefore, the first subsequence ishid1,.. .,idii. By continuing in this manner the record is partitioned into subsequences, and these subsequences are recorded to be used in our algorithm. We name the trajectories obtained by the above procedure as user actual paths (UAPs). We consider the UAPs as a valuable source of information because the mobility of the users contains both regular and random patterns[8].

Definition 1—[Mobile User]:U={u1,u2,u3,...,ui} is the set of mobile users. Each mobile user represents a physical person who carries a mobile device that has the capability of receiving services from the mobile environment, and is capable of being identified and tracked.

Definition 2—[Location]:Two special locations used to identify the regularities of the visiting locations are the generic location and the interesting location. The generic location is a collective term for one or more interesting locations, and the interesting location is a subset of the generic location. The generic location can be defined as L={l1,l2,l3,...,lj}, where each element lj represents a generic location.

The inter-esting location can be defined as the useruistaying at a location l is longer than the maximum duration, which will be defined in

Definition 3—[Timestamp and Maximum Duration]:The timestamp Tm, as defined in Table II, is assumed to have an equal period and a uniform unit. The maximum duration is considered 30 min, in general.

Definition 4—[Service]:S={s1,s2,s3,...,sn}is the set of services requested by mobile users. Each element represents an individual service ID. In addition, an optimum time is set for each service requested. If the mobile users use the acquired service longer than the optimum time, the service is regarded as an interesting or useful information service.

III. RELATED WORK

The sequential pattern mining problem was discussed in[5]. For our domain, the mobile users are assumed to be moving between the cells of a PCS network. The algorithms proposed in[5] cannot be applied directly to our domain for mining mobility patterns, because these algorithms do not take into account the network topology while generating the candidate patterns. This weakness of the proposed algorithms gives rise to generation of candidate patterns, which cannot exist as mobility patterns on the corresponding network, since only the sequence of neighboring cells of the network can be considered as a mobility pattern. Therefore, the number of candidates generated can be extremely high, and this factor can dramatically reduce the performance of the mining algorithm. In[2,3], sequential pattern mining is applied to the domain of predictive Web prefetching. Web prefetching can be defined as deriving users future requests for Web documents based on their previous requests. For effectively predicting the users future requests, user access patterns are mined from the Web logs of users previous requests and then these patterns are used for prefetch-ing. The method presented in [2,3] extends existing algorithms for mining sequential patterns in order to take the graph structure of the corresponding Web site into account during support counting, candidate generation and pruning. As we describe in Section 3.1, in the first phase of our mobility prediction algorithm, we generalize the method presented in[2,3] to be able to mine mobility patterns of users in mobile computing environments. In the latter stages of our algorithm, mobility rules are extracted from the mobility patterns, and by using these rules, user move-ments are predicted. There has been a considerable amount of research in mobility prediction, as well. It is reported in[7]that as the random movements of the user increase the performance of MMP decreases linearly.

In[1], Aljadhai and Znati use a first-order autoregressive filter in order to determine the direction of movement of a user. It is claimed in that work that the pro-posed method guarantees that the predicted mobile direction is not affected by small deviations in the mobile users direction. In the work[10], for location prediction cell-to-cell transition probabilities of a mobile user is calculated by the help of the previous inter-cell movements of the user, and then recorded to a atrix. Based on this, resource allocation is done at thekmost probable cells that are in the neighborhood of the current cell. Herekis a user-defined parameter. This method is calledMobil-ity Prediction based on Transition Matrix (TM). An adaptive algorithm for location management is proposed in[13]. By building and maintaining a dictionary of individual users path updates, the proposed on-line algorithm can learn mobile users mobility patterns. However, some serious shortcomings of the algorithm make it impracti-cal. First of all, it is very sensitive to noisy (random) user movements. Moreover, the algorithm is not scalable for huge numbers of mobile clients since the used data structure—the trie—can grow to unmanageable size so as to be used in an on-line fashion. In some of the other works such as[11,9], data mining methods such as clustering and associ-ation rule mining are used for exploring mobility patterns. In[11], a new location tracking method calledbehavior-based strategy(BBS) is presented. The aim of that work is designing a better paging area for each mobile user for each time region. The moving behavior of each mobile user is mined from long-term collection of the users moving logs. Next, time varying probability of each mobile user is estimated by using users moving behavior, and then optimal paging area of each time region is derived.. In addition, while designing our algorithm, our purpose was accurately predicting the next inter-cell movement of the mobile users which enables the system to allocate the network resources effectively, while the method proposed in[11] aims to design a better paging area. In[9], a method called dynamic clustering based prediction(DCP) of mobile user movements is presented. In that work, DCP is used for discovering user mobility patterns from collections of recorded mobile trajectories, and then these patterns are used for the prediction of movements and dynamic allocation of resources At each iteration of the clustering algorithm, the two most similar clusters (i.e., clusters that are closest in terms of weighted edit distance) are merged to form a new cluster. main difference between that work and ours is the method used for mining the UMPs. In[9], UAPs are clustered in order to mine the UMPs, while sequential pattern mining is used for the same purpose in our work. Moreover, the UMPs are used in different ways for mobility prediction in both works. Another difference is that only the next inter-cell movement of a mobile user is predicted in our method, while the complete trajectory of a mobile user is predicted in [9].

A. MOBILITY PREDICTION BASED ON MOBILITY RULES

Our algorithm consists of three phases: user mobility pattern (UMP) mining, generation of mobility rules using the mined UMPs, and the mobility prediction. The next inter-cell movement of mobile users is predicted based on the mobility rules in the last phase. We examine each phase in detail in the following subsections.

1.Mining user mobility patterns from graph traversals

We define a user mobility pattern (UMP) as a sequence of neighboring cells in the coverage region network. The consecutive cells of a UMP should be neighbors because the users cannot travel between non neighbor cells. Indeed, UMPs correspond to the expected regularities of the user ac-tual paths. In order to mine the UMPs from user actual paths (UAPs),sequential pattern mining [5] can be used. Sequential pattern mining has been previously used and examined in various re-search domains. One such work has been performed in the domain of web log mining[2,3]. In that work, sequential pattern mining is used to mine the access patterns of a user while he is visiting the pages of web sites. This method assumes the web pages to be the nodes and the links between these pages to be the edges of an unweighted directed graph,G. Then, sequential pattern mining is ap-plied to web logs by consideringG. We design a new method that is convenient for our domain, by generalizing the method of[2,3] and applying it for UMP mining. This new method employs • a different definition of the graphG, and • a new method for support counting, which generalizes the method presented in[2,3]. In our method, we use a directed graphG, where the cells in the coverage region are considered to be the vertices ofG. hese edges demonstrate the fact that a user can move from A to B or B to A directly. In Fig. 1, an example coverage region and the corresponding graph G is presented. The algorithm we have developed for UMP mining is presented inFig. 2. To understand how the UMP mining algorithm works, assume that the set of candidate pat-terns each includingkcells is found in the (k1)st run of the while loop and this set is not empty (line 4, inFig. 2). The set of these patterns, denoted byCk, is called length-k candidate patterns. Returning to the execution of our algorithm, from line 5 to line 12, first all the length-k subse-quences of all UAPs are generated and these subsequences are used to count the supports of the length-k candidate patterns. In order to be more precise, the subsequence definition is given below. Definition 1.Assume that we have two UAPs,A=ha1,a2,...,aniandB=hb1,b2,.. .,bmi. Bis a subsequence of A, iff there exists integers 16i1< <im6nsuch thatbk ¼aik , for all k, where 16k6m. In other words,B is a subsequence of A, iff all cells in B also exist in Awhile keeping their order in B(but they do not need to be consecutive in A). Let us give an example by using the coverage region given inFig. 1: assume A=hc3 ,c4 , c0,c1,c6,c5i, then B=hc4,c5iwill be a length-2 subsequence of A. In other words, the UAP B is contained by the UAPA.

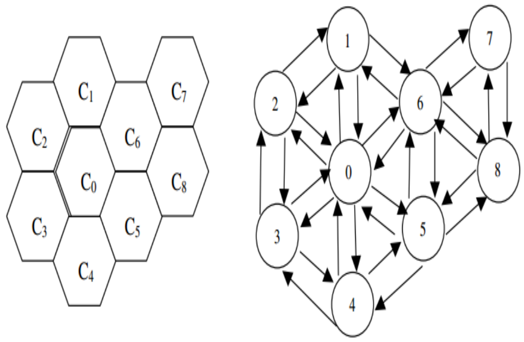

Fig. 1. An example coverage region (a) and the corresponding graph G (b).

UMPMining()
Input: All the UAPs in the database, D
Minimum value for support, $supp_{min}$
Coverage Region Graph, G
Output: User mobility patterns (UMPs), L

1. $C_1 \leftarrow$ the patterns which have a length of one
2. $k = 1$
3. $L = \emptyset$ // Initially the set of large patterns is empty
4. while $C_k \neq \emptyset$ {
5. foreach UAP $a \in D$ {
6. $S = \{ s \mid s \in C_k$ and s is a subsequence of a }
7. // S is the set of candidate length-k patterns which are also
8. // subsequences of UAP a
9. foreach $s \in S$ {
10. s.count = s.count + s.suppInc //increment the support of c and s
11. }
12. }
13. // choose the candidates which have enough support
14. $L_k = \{ s \mid s \in C_k, s.count \geq supp_{min} \}$
15. $L = L \cup L_k$ // add these length-k large patterns to the set of all large patterns
16. // Generate length-(k+1) candidate patterns
17. $C_{k+1} \leftarrow$ CandidateGeneration(L_k, G), $\forall c \in C_{k+1}$ $c.count = 0$
18. $k = k+1$
19. }
20. return L

Fig. 2. User mobility pattern mining algorithm.

In line 10 of the mining algorithm, we see that every candidates has acountvalue and this value is incremented bys.suppInc value. The count value of a candidate keeps the support given to this candidate by the UAPs. This is the point where our algorithm extends the method pre-sented in[2,3]. The method presented in that work, increments the count value of a candidate by 1 if this candidate iscontainedby a UAP. These paths can be characterized as noise and the UAPs con-taining noise are called corrupted. If the number of corrupted UAPs in the data is high, then a pattern may not have adequate support and it will be missed.

Example.An example database of UAPs is given inTable 1.

InTables 2–6, the execution of the UMP mining algorithm with suppmin= 1.33 (which corre-sponds to 33.25%) and graphGwhich is given inFig. 1is illustrated on an example using the data-base of UAPs which is given inTable 1.InTable 2, set of length-1 candidate patterns (C1) and set of length-1 large patterns (L1) are given.

Table 1
Database of user actual paths (UAPs)

UAP ID	UAP
1	⟨5,6,0,4,5⟩
2	⟨3,4,5,0⟩
3	⟨1,2,3,4,0,5⟩
4	⟨3,2,0⟩

Table 2
Length-1 candidate patterns (C_1) and length-1 large patterns (L_1)

C_1		L_1	
CAND	SUPP	PATTERN	SUPP
⟨0⟩	4	⟨0⟩	4
⟨1⟩	1	⟨2⟩	2
⟨2⟩	2	⟨3⟩	3
⟨3⟩	3	⟨4⟩	3
⟨4⟩	3	⟨5⟩	3
⟨5⟩	3		
⟨6⟩	1		
⟨7⟩	0		
⟨8⟩	0		

Table3
Length-2 candidate patterns (C_2) and length-2 large patterns (L_2)

C_2				L_2	
CAND	SUPP	CAND	SUPP	PATTERN	SUPP
⟨0,1⟩	0	⟨3,2⟩	1	⟨0,5⟩	1.5
⟨0,2⟩	0	⟨3,4⟩	2	⟨2,0⟩	1.33
⟨0,3⟩	0	⟨4,0⟩	1.5	⟨3,0⟩	1.33
⟨0,4⟩	1	⟨4,3⟩	0	⟨3,4⟩	2
⟨0,5⟩	1.5	⟨4,5⟩	2.5	⟨4,0⟩	1.5
⟨0,6⟩	0	⟨5,8⟩	0	⟨4,5⟩	2.5
⟨2,0⟩	1.33	⟨5,0⟩	1.5	⟨5,0⟩	1.5
⟨2,1⟩	0	⟨5,4⟩	0.33		
⟨2,3⟩	1	⟨5,6⟩	1		
⟨3,0⟩	1.33				

Next,C2is generated by using the candidate generation algorithm given in Fig. 3and,L1is used in this process. Then, the supports of these candidates are counted and the patterns which have a support value larger thansuppminare assigned to setL2. The sets C2andL2are presented in Table 3. HavingL2, C3is generated using CandidateGeneration( ) function, and then the large patterns in C3are assigned to the setL3. These sets are shown in Table 4. The set of length-4 candidate patterns,C4, is illustrated in Table 5.

Table 4
Length-3 candidate patterns (C_3) and length-3 large patterns (L_3)

C_3								L_3	
CAND	SUPP	CAND	SUPP	CAND	SUPP	CAND	SUPP	PATTERN	SUPP
(0,5,8)	0	(2,0,6)	0	(3,4,5)	1.5	(4,5,4)	0	(3,4,0)	1.5
(0,5,0)	0	(3,0,1)	0	(4,0,1)	0	(4,5,6)	0	(3,4,5)	1.5
(0,5,4)	0	(3,0,2)	0	(4,0,2)	0	(5,0,1)	0		
(0,5,6)	0	(3,0,3)	0	(4,0,3)	0	(5,0,2)	0		
(2,0,1)	0	(3,0,4)	0	(4,0,4)	0	(5,0,3)	0		
(2,0,2)	0	(3,0,5)	0.5	(4,0,5)	1	(5,0,4)	0.5		
(2,0,3)	0	(3,0,6)	0	(4,0,6)	0	(5,0,5)	0.33		
(2,0,4)	0	(3,4,0)	1.5	(4,5,8)	0	(5,0,6)	0		
(2,0,5)	0.33	(3,4,3)	0	(4,5,0)	1				

Table 5
Length-4 candidate patterns (C_4)

C_4			
PATTERN	SUPP	PATTERN	SUPP
(3,4,0,1)	0	(3,4,0,6)	0
(3,4,0,2)	0	(3,4,5,8)	0
(3,4,0,3)	0	(3,4,5,0)	1
(3,4,0,4)	0	(3,4,5,4)	0
(3,4,0,5)	1	(3,4,5,6)	0

Table 6
The set of all large patterns

L			
PATTERN	SUPP	PATTERN	SUPP
(0)	4	(3,0)	1.33
(2)	2	(3,4)	2
(3)	3	(4,0)	1.5
(4)	3	(4,5)	2.5
(5)	3	(5,0)	1.5
(0,5)	1.5	(3,4,0)	1.5
(2,0)	1.33	(3,4,5)	1.5

Unfortunately, none of these patterns have a support larger than suppmin which indicates that L4does not contain any patterns. Therefore, the UMP mining algorithm terminates with the set of large candidates, L, which is shown in Table 6.

IV. EXPERIMENTAL RESULTS

For simulation, we have adapted the simulation model which is presented in our earlier work [9]. In this model, it is assumed that a mobile user travels on a 15 by 15 hexagonal shaped network which gives a total of 225 base stations. In order to generate the user actual paths(UAPs), first a number of user mobility patterns (UMPs) is generated. The length of a UMP is determined by a uniform distribution with a mean lengthl. Each UMP is taken as a random walk over the hexagonal network. There are two types of UAPs generated. The first type consists of UAPs that follow a UMP and the sec-ond type consists ofoutliers(i.e., those which do not follow a pattern). The ratio of the number of outliers to the number of UAPs that follow a UMP is denoted by 0. For each new UAP we decide whether it is going to be an outlier or not, according to the value 0.. We insert random cells between the consec-utive cells of the UMP. In order to accomplish this, we define a corruption ratioc, which de-notes the ratio of the number of such random cells to the number of cells in the corresponding UMP. The total number of generated UAPs is 10,000 and from these, we construct the training and test sets. The number of UAPs in training set is 9000 and the number of UAPs in test set is 1000. UMPs are mined from the UAPs in the training set and then the mobility rules that will be used in prediction are generated by using these UMPs. The UAPs in the test set are used for evaluating the prediction accuracy of our algorithm. There are three possible outcomes for the location prediction, when compared to the actual location:

- The predictor correctly identified the location of the next move.
- The predictor incorrectly identified the location of the next move.
- The predictor returned ‘‘no prediction’’.

Table 8
Symbol table for the parameters used in our experiment

Symbol	Definition	Default values
m	Maximum number of predictions made each time	2
l	Average length of UAPs	5
c	Corruption factor	0.4
o	Outlier percentage	30%
$supp_{min}$	Minimum support percentage	0.1%
$conf_{min}$	Minimum confidence percentage	80%

All predictors encounter situations in which they are unable to make a prediction; in particular, all realistic predictors will have no prediction for the first location of each user trace.

We use two performance measures for the evaluation of the proposed algorithm:

- Recall: the number of correctly predicted cells divided by the total number of requests (i.e., the total number of inter-cell movements that the user makes). Thus, the recall counts the ‘‘no-pre-diction’’ case as an incorrect prediction.
- Precision: the number of correctly predicted cells divided by the total number of predictions made. This metric is appropriate for applications that may prefer no prediction to a wild guess. The parameters used in the experiments and their default values are given inTable 8. The de-fault values ofl, cand 0 are adapted from[9].

V. CONCLUSION

In this paper, we proposed a data mining method for mining UMBPs. We showed that the simple approach of computing by applying standard graph-matching algorithms and the DBMS primitives of grouping, sorting, and joining could be utilized to yield efficient match join operations. Moreover, a novel mining scheme was proposed to mine associated trees so that we can locate user behavior patterns .

ACKNOWLEDGMENT

It gives me, Mohammad Waseem great pleasure on bringing out paper entitled “Detecting Location Patterns in Mobile Environment based on User Movements.”

I express my deep sense of gratitude and sincere regards to my Guide Prof. Mrs. R.R.Shelke for her timely guidance and helpful discussions has supported me immensely in selecting a new proposal and completing paper work.

REFERENCES

[1] A. Aljadhai, T. Znati, Predictive mobility support for QoS provisioning in mobile wireless environments, IEEE J. Select. Area Commun. 19 (10) (2001) 1915–1930.

[2] A. Nanopoulos, D. Katsaros, Y. Manolopoulos, Effective prediction of web user accesses: a data mining approach, in: Proceedings of the WebKDD Workshop (WebKDD01), 2001.

[3] A. Nanopoulos, D. Katsaros, Y. Manolopoulos, A data mining algorithm for generalized web prefetching, IEEE Trans. Knowl. Data Eng. 15 (5) (2003) 1155–1169.

[4] Tzung-Shi Chen, Yen-Ssu Chou and Tzung-Cheng Chen, “Mining User Movement Behavior Patterns in a Mobile Service Environment,” IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 42, No. 1, January2012.

[5] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the IEEE Conference on Data Engineering (ICDE95), 1995, pp. 3–14.

[6] B. Liang, Z. Haas, Predictive distance-based mobility management for PCS networks, in: Proceedings of the IEEE Conference on Computer and Communications (IEEE INFOCOM99), 1999, pp. 1377–1384.

[7] G.Y. Liu, M.Q. Gerald, A predictive mobility management algorithm for wireless mobile computing and communications, in: Proceedings of the IEEE International Conference on Universal Personal Communications, 1995, pp. 268–272.

[8] T. Liu, P. Bahl, I. Chlamtac, Mobility modeling, location tracking, and trajectory prediction in wireless ATM networks, IEEE J. Select. Area Commun. 16 (6) (1998) 922–936.

[9] D. Katsaros, A. Nanopoulos, M. Karakaya, G. Yavas, O. Ulusoy, Y. Manolopoulos, Clustering mobile trajectories for resource allocation in mobile environments, in: Intelligent Data Analysis Conference (IDA2003)Lecture Notes in Computer Science, vol. 2810, Springer-Verlag, 2003.

[10] S. Rajagopal, R.B. Srinivasan, R.B. Narayan, X.B.C. Petit, GPS-based predictive resource allocation in cellural networks, in: Proceedings of the IEEE International Conference on Networks (IEEE ICON02), 2002, pp. 229–234.

[11] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid, “Supporting top-k join queries in relational databases,”J. Very Large Databases, vol. 13, no. 3, pp. 207–221, Sep. 2004.

[12] S. Mohan, R. Jain, Two user location strategies for personal communication systems, IEEE Personal Commun. Mag. 1 (1994) 42–50.

[13] A. Bhattacharya, S.K. Das, LeZi—Update: an information-theoretic approach to track mobile users in PCS networks, ACM Wireless Networks 8 (2–3) (2002) 121–135.