A Caching Strategy to Reduce Network Impacts of PCS

Ravi Jain, *Member, IEEE,* Yi-Bing Lin, Charles Lo, and Seshadri Mohan, *Member, IEEE*

***Abstract*— We propose an auxiliary strategy, called *per-user caching,* for locating users who move from place to place while using Personal Communications Services (PCS). The caching strategy augments the basic location strategy proposed in existing standards such as GSM and IS-41, with the objective of reducing network signaling and database loads in exchange for increased CPU processing and memory costs. Since technology trends are driving the latter costs down, the auxiliary strategy will become increasingly attractive.**

The idea behind caching is to reuse the information about a called user's location for subsequent calls to that user, and is useful for those users who receive calls frequently relative to the rate at which they change registration areas. This idea attempts to explqit the spatial and temporal locality in calls received by users, similar to the idea of exploiting locality of file access in computer systems.

We use a reference PCS architecture and the notion of a user's *local call-to-mobility ratio* (LCMR) to quantify the costs and benefits of using caching and classes of users for whom it would be beneficial. We also present two simple algorithms for estimating users' LCMR and the situation in which each is preferable. We show that under a variety of assumptions caching is likely to yield significant net benefits in terms of reduced signaling network traffic and database loads.

I. Introduction

We consider the problem of locating users who move from place to place while using Personal Communications Services (PCS). Previous studies [11], [14], [10], [12] have shown that, with predicted levels of PCS users, there will be significant loads upon the signaling network and network databases, and that these loads are dependent upon the data management strategies adopted. We present a user location strategy which has the potential to reduce these loads significantly. The strategy we discuss here is an *auxiliary* strategy, in that it augments the *basic* user location strategies proposed in standards such as the North American IS-41 cellular standard [3] and the European GSM standard for mobile communications [13], [17]. For a survey of various user location strategies in PCS systems, see [7] and [16].

The strategy we present is the use of *per-user caching.* This strategy, like other auxiliary strategies [7], attempts to reduce the network signaling and database loads of the basic strategies in exchange for increased CPU processing and memory costs. Since technology trends are driving the latter costs down, deploying the caching strategy on a system-wide basis will become increasingly attractive. Once deployed, whether the caching strategy should be invoked for a particular user is a function of the user's mobility and communications patterns, as discussed below.

The *basic* user location strategies proposed in the IS-41 [3] and GSM [13], [17] standards are *two-level* strategies in that they use a two-tier system of home and visited databases. In two-level strategies, as in most user location strategies, the same user location procedure has to be invoked for every call to a PCS user. The key observation we make is that, in many cases, it should be possible to re-use the information about the user's location obtained during the previous call to that user. This information will be useful for those users who receive calls frequently relative to the rate at which they change registration areas. This idea attempts to exploit the spatial and temporal locality in calls received by users, similar to the idea of exploiting locality of file access in computer systems [19], and is thus essentially a form of *caching.*

The outline of this paper is as follows. In Section II, we describe the PCS network architecture that we assume for the presentation and analysis of the basic and caching location strategies. In Section III, we describe the caching strategy. A feature of the caching location strategy is that it is useful only for certain classes of PCS users, those meeting certain call and mobility criteria. We encapsulate this notion in the definition of the user's call-to-mobility ratio (CMR), and Local CMR (LCMR), in Section IV. We then use this definition and our PCS network reference architecture to quantify the costs and benefits of caching and the threshold LCMR for which caching is beneficial, thus characterizing the classes of users for which caching should be applied. In Section V we describe two methods for estimating users' LCMR and compare their effectiveness when call and mobility patterns are fairly stable as well as when they may be variable. In Section VI we briefly discuss alternative architectures and implementation issues of the strategy proposed, including cache management issues, and mention other auxiliary strategies which can be designed. Section VII provides some conclusions and discussion of future work.

It should be stressed at the outset that the assumptions we have used in this paper constitute one reference set of assumptions. A number of variations in the assumptions could be considered, including variations in the network architecture and in the auxiliary location strategy. In most cases these variations may alter our analysis in relatively minor ways

Manuscript received February 18, 1994.

R. Jain, Y.-B. Lin, and S. Mohan are with Bell Communications Research, Morristown, NJ 07692.

C. Lo is with AirTouch Communications, Walnut Creek, CA 94596 USA.

IEEE Log Number 9403391.

and may not significantly affect the qualitative conclusions we draw. The intent of this paper is to present the key ideas behind the caching auxiliary strategy and to develop a method for quantifying its costs and benefits. This method can then be applied to specific architectures and deployment scenarios as needed.

II. PCS Network Architecture

PCS users receive calls via either wireless or wire-line access. In general, calls may deliver voice, data, text, facsimile or video information. For our purposes, we define the location of a PCS user, as known by the wire-line network, as the *registration area* (RA) in which the user is located. For users attached directly to a wire-line network, the RA is defined as the point of attachment. For users attached via wireless links, the situation is described as follows. In order to deliver calls by wireless links, the geographical region covered by a PCS network is divided into radio port coverage areas, or *cells.* Each cell is primarily served by one radio *base station,* although a base station may serve one or more cells. The base station locates a user, and delivers calls to and from the user, by means of paging within the cell(s) it serves. Base stations are connected to the rest of the wire-line network by wire-line links. An RA consists of an aggregation of cells, forming a contiguous geographical region.

We assume that a signaling network is used to set up calls which is distinct from the network used to actually transport the information contents of the calls. Specifically, we assume a Common Channel Signaling (CCS) network is used to set up calls which uses the Signaling System No. 7 (SS7) protocols (see [15] for a tutorial).

Fig. 1 illustrates the reference signaling network assumed in this study. This architecture is meant as a reference architecture for a hypothetical geographical region, and is not necessarily the architecture corresponding to any particular implementation. The cells of the geographical region are served by base stations and are aggregated into RA's. The base stations of an RA are connected via a wire-line network to an end-office switch, or Service Switching Point (SSP). Each SSP serves a single RA. SSP's of different RA's are in turn connected to a two-level hierarchy of Signaling Transfer Points (STP's), comprised of a Regional STP (RSTP) connected to all Local STPs (LSTP's) in the region, which perform message routing and other SS7 functions. (In practice each STP actually consists of two STP's in a mated-pair configuration or redundancy [15]; for simplicity Fig. 1 only shows one of the two STP's of each mated pair.) The RSTP is also connected to a Service Control Point (SCP), which is assumed to contain the functionality associated with a Home Location Register (HLR) database.

For simplicity, we assume that in functional terms the MSC is collocated with the SSP. In addition, a distinct Visitor Location Register (VLR) database is associated with each MSC. These assumptions are not unreasonable. It is already anticipated that the VLR will be collocated with the MSC, and the MSC/VLR combination is likely to evolve to be SS7 compatible. (In the rest of this paper, the terms switch or SSP will be used interchangeably, depending on the context.) Each switch is assumed to serve exactly one RA, which, in turn, may be comprised of one or more cells. This assumption is used to simplify the ensuing analysis, although in practice, each VLR may serve a number of RA's.

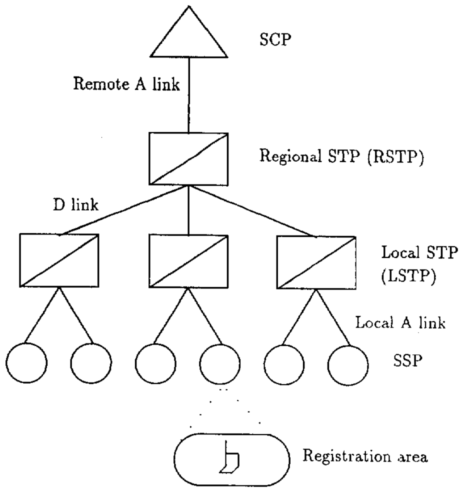


Fig. 1. Reference CCS network architecture.

In this paper we do not address issues relating to the content of messages and other information transfer (i.e., for billing, etc.) which may occur during a call. For simplicity it is assumed that message sizes are equal for different types of transactions (e.g., location, request, registration, and deregistration), for both query and update invocations as well as their associated response messages. Since we only perform a comparative analysis of the basic strategy with and without the auxiliary caching strategy, the conclusions will not be affected by this simplification.

III. Per-User Location Caching

The basic idea behind per-user location caching is that the volume of SS7 message traffic and database accesses required in locating a called subscriber can be reduced by maintaining local storage, or *cache,* of user location information at a switch. At any switch, location caching for a given user should be employed only if a large number of calls *originate* for that user from that switch, relative to the user's mobility. Note that the cached information is kept at the switch from which calls originate, which may or may not be the switch where the called user is currently registered.

Location caching involves the storage of location *pointers* at the originating switch; these point to the VLR (and the associated switch) where the user is currently registered. We refer to the procedure of locating a PCS user as a *FIND* operation, borrowing the terminology from [1]. We define a basic *FIND,* or *BasicFIND( ),* as one where the following sequence of steps takes place.

1) The call to a PCS user is directed to the switch nearest the caller.
2) Assuming that the called party is not located within the immediate RA, the switch queries the HLR for routing

information.

3) The HLR contains a pointer to the VLR in whose associated RA the called party is currently situated, and launches a query to that VLR.
4) The VLR in turn queries the MSC to determine whether the user terminal is capable of receiving the call (i.e., is idle), and if so, the MSC returns a routable address (referred to as the Temporary Local Directory Number, or TLDN, in IS-41) to the VLR.
5) The VLR relays the routing address back to the originating switch via the HLR.

At this point, the originating switch can route the call to the destination switch. Alternately, *BasicFIND( )* can be described by pseudocode as follows. (We observe that a more formal method of specifying PCS protocols may be desirable.)

```
BasicFIND( ){
  Call to PCS user is detected at local switch;
  if called party is in same RA then return;
  Switch queries called party's HLR;
  Called party's HLR queries called party's current VLR, V;
  V returns called party's location to HLR;
  HLR returns location to calling switch;
}
```

In the *FIND* procedure involving the use of location caching, or *CacheFIND( )*, each switch contains a local memory (cache) that stores location information for subscribers. When the switch receives a call origination (from either a wire-line or wireless caller) directed to a PCS subscriber, it first checks its cache to see if location information for the called party is maintained. If so, a query is launched to the pointed VLR; if not, *BasicFIND( )*, as described previously, is followed. If a cache entry exists and the pointed VLR is queried, two situations are possible. If the user is still registered at the RA of the pointed VLR (i.e., we have a *cache hit*), the pointed VLR returns the user's routing address. Otherwise, the pointed VLR returns a *cache miss*.

```
CacheFIND( ){
    Call to PCS user is detected at local switch;
    if called is in same RA then return;
    if there is no cache entry for called user
    then invoke BasicFIND( ) and return;
    Switch queries the VLR, V, specified in the cache entry;
    if called is at V, then
        V returns called party's location to calling switch;
    else {
        V returns "miss" to calling switch;
        Calling switch invokes BasicFIND( );
    }
}
```

What when a cache hit occurs we save one query to the HLR (a VLR query is involved in both *CacheFIND( )* and *BasicFIND( )*), and we also save traffic along some of the signaling links; instead of four message transmissions as in *BasicFIND( )*, only two are needed. In steady state operation, the cached pointer for any given user is updated only upon a "miss."

Note that the *BasicFIND( )* procedure differs from that specified for "roaming" subscribers in the IS-41 standard [3]. In the IS-41 standard, the second line in the *BasicFIND( )* procedure is omitted, i.e., every call results in a query of the called user's HLR. Thus, in fact, the procedure specified in the standard will result in an even higher network load than the *BasicFIND( )* procedure specified here. However, in order to make a fair assessment of the benefits of *CacheFIND( )*, we have compared it against *BasicFIND( )*. Thus the benefits of *CacheFIND( )* investigated here depend specifically on the use of caching, and not simply on the availability of user location information at the local VLR.

IV. Caching Threshold Analysis

In this section we investigate the classes of users for which the caching strategy yields net reductions in signaling traffic and database loads. We characterize classes of users by their *call-to-mobility ratio* (CMR). The *call-to-mobility ratio* (CMR) of a user is the average number of calls to a user per unit time, divided by the average number of times the user changes registration areas per unit time. We also define a *local CMR* (LCMR), which is the average number of calls to a user *from a given originating switch* per unit time, divided by the average number of times the user changes registration areas per unit time.

For each user, the amount of savings due to caching is a function of the probability that the cached pointer correctly points to the user's location, and increases with the user's LCMR. In this section we quantify the minimum value of LCMR for caching to be worthwhile. This caching threshold is parameterized with respect to costs of traversing signaling network elements and network databases, and can be used as a guide to select the subset of users to whom caching should be applied. The analysis in this section shows that estimating users' LCMR's, preferably dynamically, is very important in order to apply the caching strategy. The next section will discuss methods for obtaining this estimate.

From the pseudocode for *BasicFIND( )*, the signaling network cost incurred in locating a PCS user in the event of a call is the sum of the cost of querying the HLR (and receiving the response), and the cost of querying the VLR which the HLR points to (and receiving the response). Let

α=Cost of querying the HLR and receiving a response

β=Cost of querying the pointed VLR and receiving a response.

Then, the cost of the *BasicFIND( )* operation is

$$C_B = \alpha + \beta. \tag{1}$$

To quantify this further, assume costs for traversing various network elements as follows.

A_l=Cost of transmitting a location request or response message on A-link between SSP and LSTP

D=Cost of transmitting a location request or response message on D-link

A_r=Cost of transmitting a location request or response message on A-link between RSTP and SCP

L=Cost of processing and routing a location request or response message by LSTP

R=Cost of processing and routing a location request or response message by RSTP

H_Q=Cost of a query to the HLR to obtain the current VLR location

V_Q=Cost of a query to the VLR to obtain the routing address.

Then, using the PCS reference network architecture (Fig. 1),

$$\alpha = 2(A_l + D + A_r + L + R) + H_Q \tag{2}$$

$$\beta = 2(A_l + D + A_r + L + R) + V_Q. \tag{3}$$

From (1), (2), and (3),

$$C_B = 4(A_l + D + A_r + L + R) + H_Q + V_Q. \tag{4}$$

We now calculate the cost of *CacheFIND( )*. We define the *hit ratio* as the relative frequency with which the cached pointer correctly points to the user's location when it is consulted. Let

p=cache hit ratio

C_H=Cost of the *CacheFIND( )* procedure when there is a hit

C_M=Cost of the *CacheFIND( )* procedure when there is a miss.

Then the cost of *CacheFIND( )* is

$$C_C = pC_H + (1 - p)C_M. \tag{5}$$

For *CacheFIND( )*, the signaling network costs incurred in locating a user in the event of an incoming call depend upon the hit ratio as well as the cost of querying the VLR which is stored in the cache; this VLR query may or may not involve traversing the RSTP. In the following, we say a VLR is a *local* VLR if it is served by the same LSTP as the originating switch, and a *remote* VLR otherwise. Let

q=Prob(VLR in originating switch's cache is a local VLR)

δ=Cost of querying a local VLR

ϵ=Cost of querying a remote VLR

η=Cost of updating the cache upon a miss.

Then,

$$\delta = 4A_l + 2L + V_Q \tag{6}$$

$$\epsilon = 4(A_l + D + L) + 2R + V_Q \tag{7}$$

$$C_H = q\delta + (1 - q)\epsilon. \tag{8}$$

Since updating the cache involves an operation to a fast local memory rather than a database operation, we shall assume in the following that $\eta = 0$. Then,

$$C_M = C_H + C_B = q\delta + (1 - q)\epsilon + \alpha + \beta. \tag{9}$$

From (5), (8), and (9) we have

$$C_C = \alpha + \beta + \epsilon - p(\alpha + \beta) + q(\delta - \epsilon). \tag{10}$$

TABLE I
Minimum Hit Ratios and LCMR's for Various Individual Dominant Signaling Network Cost Terms

Dominant Cost Term	Hit Ratio Threshold, p_T	LCMR Threshold, $LCMR_T$	LCMR Threshold, ($q = 0.043$)	LCMR Threshold, ($q = 0.25$)
A_l	1	∞	∞	∞
A_r	0	0	0	0
D	$1 - q$	$1/q - 1$	22	3
L	$1 - q/2$	$2/q - 1$	45	7
R	$1 - q/2$	$2/q - 1$	45	7
H_Q	0	0	0	0
V_Q	1	∞	∞	∞

For net cost savings we require $C_C < C_B$, or that the hit ratio exceeds a *hit ratio threshold*, p_T, derived using (5), (8), and (1):

$$p > p_T = \frac{C_H}{C_B} = \frac{\epsilon + q(\delta - \epsilon)}{\alpha + \beta} \tag{11}$$

$$= \frac{4A_l + 4D + 4L + 2R + V_Q - q(4D + 2L + 2R)}{4A_l + 4D + 4A_r + 4L + 4R + H_Q + V_Q}. \tag{12}$$

Equation (12) specifies the hit ratio threshold for a user, evaluated at a given switch, for which local maintenance of a cached location entry produces cost savings. As pointed out previously, a given user's hit ratio may be location dependent, since the rates of calls destined for that user may vary widely across switches.

The hit ratio threshold in (12) is comprised of heterogeneous cost terms, i.e., transmission link utilization, packet switch processing, and database access costs. Therefore, numerical evaluation of the hit ratio threshold requires either detailed knowledge of these individual quantities or some form of simplifying assumptions. Based on the latter approach, two possible methods of evaluation may be employed:

1) assume one or more cost terms dominate, and simplify (12) by setting the remaining terms to zero, or
2) establish a common unit of measure for all cost terms, for example *time delay*. In this case, A_l, A_r, and D may represent transmission delays of fixed transmission speed (e.g., 56 kb/s) signaling links, L and R may constitute the sum of queueing and service delays of packet switches (i.e., STP's), and H_Q and V_Q the transaction delays for database queries.

We adopt the first method in this section and evaluate (12) assuming a single term dominates. (In Section V we present results using the second method). Table I shows the hit ratio threshold required to obtain net cost savings, for each case in which one of the cost terms is dominant.

In Table I we see that if the cost of querying a VLR or of traversing a local A-link is the dominant cost, caching for users who may move is never worthwhile, regardless of users' call reception and mobility patterns. This is because the caching strategy essentially distributes the functionality of the HLR to the VLR's. Thus the load on the VLR and the local A-link is always increased, since any move by a user results in a cache miss. On the other hand, for a fixed user (or telephone), caching is *always* worthwhile. We also observe that if the remote A-links or HLR querying are the bottlenecks caching is worthwhile even for users with very low hit ratios.

As a simple average-case calculation, consider the net network benefit of caching when HLR access and update is the performance bottleneck. Consider a scenario where $u = 50\%$ of PCS users receive $c = 80\%$ of their calls from $s = 5$ RA's where their hit ratio $p > 0$, and $s' = 4$ of the SSP's at those RA's contain sufficiently large caches. Assume that caching is applied only to this subset of users, and to no other users. Suppose that the average hit ratio for these users is $p = 80\%$, so that 80% of the HLR accesses for calls to these users from these RA are avoided. Then the net saving in the accesses to the system's HLR is $H = (u c s' p)/s = 25\%$.

We discuss other quantities in Table I below. It is first useful to relate the cache hit ratio to users' calling and mobility patterns directly via the LCMR. Doing so requires making assumptions about the distribution of the user's calls and moves. We consider the steady state where the incoming call stream from an SSP to a user has a mean arrival rate λ, and the time that the user resides in an RA has mean $1/\mu$. Thus

$$\text{LCMR} = \frac{\lambda}{\mu}. \tag{13}$$

Let t be the time interval between two consecutive calls from the SSP to the user, and t_1 be the time interval between the first call and the time when the user moves to a new RA. From the random observer property of the arrival call stream [4], if call arrivals are a Poisson and $F(t)$ is an exponential distribution, the hit ratio is

$$p = \Pr[t < t_1] = \int_{t=0}^{\infty} \lambda e^{-\lambda t} \int_{t_1=t}^{\infty} \mu[1 - F(t_1)]\, dt_1\, dt.$$

Then

$$p = \frac{\lambda}{\lambda + \mu} \tag{14}$$

and we can derive the *LCMR threshold*, the minimum LCMR required for caching to be beneficial assuming incoming calls are a Poisson process and inter-move times are exponentially distributed,

$$\text{LCMR}_T = \frac{p_T}{1 - p_T}. \tag{15}$$

Equation 15 is used to derive LCMR thresholds assuming various dominant costs terms, as shown in Table I.

Several values for LCMR_T in Table I involve the term q, the probability that the pointed VLR is a local VLR. These values may be numerically evaluated by simplifying assumptions. For example, assume that all the SSP's in the network are uniformly distributed amongst l LSTP's. Also, assume that all the PCS subscribers are uniformly distributed in location across all SSP's, and that each subscriber exhibits the same incoming call rate at every SSP. Under these conditions, q is simply $1/l$. Consider the case of the public switched telephone network. Given that there are a total of 160 Local Access Transport Area (LATA) across the 7 Regional Bell Operating Company (RBOC) regions [2], the average number of LATA's, or l, is 160/7 or 23. Table I shows the results with $q = 1/l$ in this case.

We observe that the assumption that all users receive calls uniformly from all switches in the network is extremely conservative. In practice, we expect that user call reception patterns would display significantly more locality, so that q would be larger and the LCMR thresholds required to make caching worthwhile would be smaller. It is also worthwhile to consider the case of a RBOC region with PCS deployed in a few LATA only, a likely initial scenario, say 4 LATA's. In either case the value of q would be significantly higher; Table I shows the LCMR threshold when $q = 0.25$.

It is possible to quantify the net costs and benefits of caching in terms of signaling network impacts in this way, and determine the hit ratio and LCMR threshold above which users should have the caching strategy applied. Applying caching to users whose hit ratio and LCMR is below this threshold results in net increases in network impacts. It is thus important to estimate users' LCMR's accurately. The next section discusses how to do so.

V. Techniques for Estimating Users' LCMR

Here we sketch some methods of estimating users' LCMR. A simple and attractive policy is to not estimate these quantities on a per-user basis at all. For instance, if the average LCMR over all users in a PCS system is high enough (and from Table I, it need not be high depending upon which network elements are the dominant costs), then caching could be used at *every* SSP to yield net system-wide benefits. Alternatively, if it is known that, at any *given* SSP, the average LCMR over all users is high enough, a cache can be installed at that SSP. Other variations can be designed.

One possibility for deciding about caching on a per-user basis is to maintain information about a user's calling and mobility pattern at the HLR, and download it periodically to selected SSP's during off-peak hours. It is easy to envision numerous variations on this idea.

In this section we investigate two possible techniques for estimating LCMR on a per-user basis when caching is to be deployed. The first algorithm, called the *running average* algorithm, simply maintains a running average of the hit ratio for each user. The second algorithm, called the *reset-K* algorithm, attempts to obtain a measure of the hit ratio over the "recent" history of the user's movements. We describe the two algorithms below, and evaluate their effectiveness using a stochastic analysis taking into account user calling and mobility patterns.

A. The Running Average Algorithm

The running average algorithm maintains, for every user that has a cache entry, the running average of the hit ratio. A running count is kept of the number of calls to a given user, and, regardless of the *FIND* procedure used to locate the user, a running count of the number of times that the user was at the same location for any two consecutive calls; the ratio of these numbers provides the *measured* running average of the hit ratio. We denote the measured running average of the hit

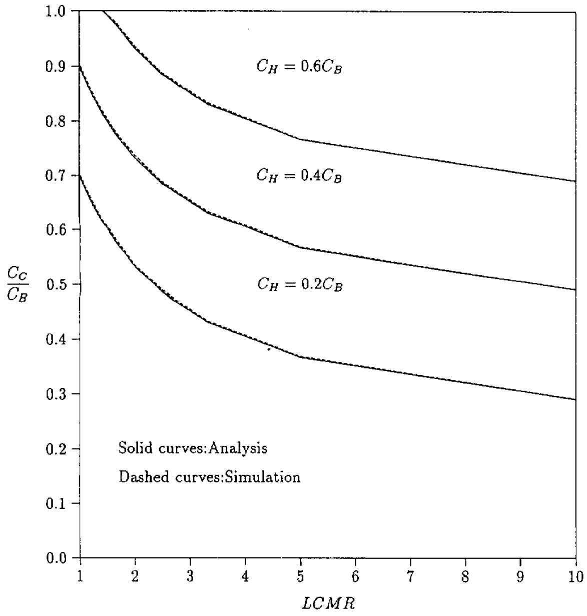


Fig. 2. The location tracking cost for the running average algorithm.

ratio by p_M; in steady state, we expect that $p_M = p$. The user's previous location as stored in the cache entry is used only if the running average of the hit ratio, p_M, is greater than the cache hit threshold p_T. Recall that the cache scheme outperforms the basic scheme if $p > p_T = C_H/C_B$. Thus in steady state, the running average algorithm will outperform the basic scheme when $p_M > p_T$.

We consider, as before, the steady state where the incoming call stream from an SSP to a user is a Poisson process with arrival rate λ, and the time that the user resides in an RA has an exponential distribution with mean $1/\mu$. Thus LCMR = λ/μ [eq. (13)] and the location tracking cost at steady state is

$$C_C = \begin{cases} p_M C_H + (1 - p_M)C_B, & p_M > p_T \\ C_B, & \text{otherwise.} \end{cases} \tag{16}$$

Fig. 2 plots the cost ratio C_C/C_B from (16) against LCMR. (This corresponds to assigning uniform units to all cost terms in (12), i.e., the second evaluation method as discussed in Section IV. Thus the ratio C_C/C_B may represent the percentage reduction in user location time with the caching strategy compared to the basic strategy.) The figure indicates that in the steady state, the caching strategy with the running average algorithm for estimating LCMR can significantly outperform the basic scheme if LCMR is sufficiently large. For instance with LCMR $\sim$ 5, caching can lead to cost savings of 20–60% over the basic strategy.

Equation (16) (cf., solid curves in Fig. 2) is validated against a simple Monte Carlo simulation (cf., dashed curves in Fig. 2). In the simulation, the confidence interval for the 95% confidence level of the output measure C_C/C_B is within 3% of the mean value. This simulation model will later be used to study the running average algorithm when the mean of the movement distribution changes from time to time [which cannot be modeled by using (16)].

One problem with the running average algorithm is that the parameter p is measured from the entire past history of the user's movement, and the algorithm may not be sufficiently dynamic to adequately reflect the recent history of the user's mobility patterns.

B. *The Reset-K Algorithm*

We may modify the running average algorithm such that p is measured from the "recent" history. Define every K incoming calls as a *cycle*. The modified algorithm, which is referred to as the *reset-K* algorithm, counts the number of cache hits n in a cycle. If the measured hit ratio for a user, $p_M = n/K \geq p_T$, then the cache is *enabled* for that user, and the cached information is always used to locate the user in the next cycle. Otherwise, the cache is *disabled* for that user and the basic scheme is used. At the beginning of a cycle, the cache hit count is reset, and a new p_M value is measured during the cycle.

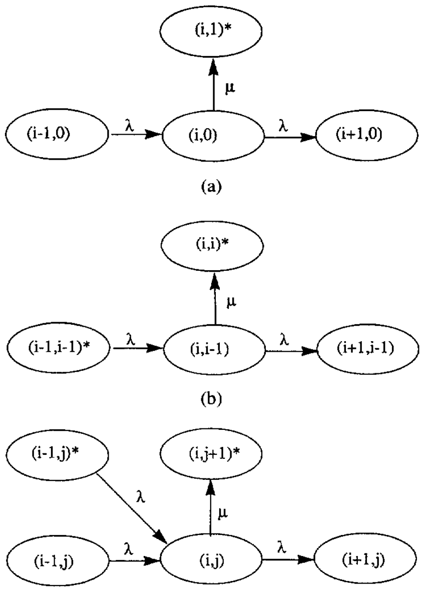
To study the performance of the reset-K algorithm, we model the number of cache misses in a cycle by a Markov process. Assume as before that the call arrivals are a Poisson process with arrival rate λ and the time period the user resides in an RA has an exponential distribution with mean $1/\mu$. A pair (i, j), where $i > j$, represents the state that there are j cache misses before the first i incoming phone calls in a cycle. A pair $(i, j)^*$, where $i \geq j \geq 1$, represents the state that there are $j - 1$ cache misses before the first i incoming phone calls in a cycle, and the user moves between the ith and the $i + 1$st phone calls. The difference between (i, j) and $(i, j)^*$ is that if the Markov process is in the state (i, j) and the user moves, then the process moves into the state $(i, j + 1)^*$. On the other hand, if the process is in state $(i, j)^*$ when the user moves, the process remains in $(i, j)^*$ because at most one cache miss occurs between two consecutive phone calls.

Fig. 3(a) illustrates the transitions for state $(i, 0)$ where $2 < i < K + 1$. The Markov process moves from $(i - 1, 0)$ to $(i, 0)$ if a phone call arrives before the user moves out. The rate is λ. The process moves from $(i, 0)$ to $(i, 1)^*$ if the user moves to another RA before the $i+1$st call arrival. Let $\pi(i, j)$ denote the probability of the process being in state (i, j). Then the transition equation is

$$\pi(i, 0) = \frac{\lambda}{\lambda + \mu}\pi(i - 1, 0), \qquad 2 < i < K + 1. \tag{17}$$

Fig. 3(b) illustrates the transitions for state $(i, i - 1)$ where $1 < i < K + 1$. The only transition into the state $(i, i - 1)$ is from $(i - 1, i - 1)^*$, which means that the user always moves to another RA after a phone call. (Note that there can be no state $(i - 1, i - 1)$ by definition, and hence no transition from such a state.) The transition rate is λ. The process moves from $(i, i - 1)$ to $(i, i)^*$ with rate μ, and moves to $(i + 1, i - 1)$ with rate λ. Let $\pi^*(i, j)$ denote the probability of the process being in state $(i, j)^*$. Then the transition equation is

$$\pi(i, i - 1) = \frac{\lambda}{\lambda + \mu}\pi^*(i - 1, i - 1), \qquad 1 < i < K + 1. \tag{18}$$



(a)

(b)

(c)

Fig. 3. State transitions. (a) Transistions for state $(i, 0)(2 < i < K + 1)$. (b) Transitions for state $(i, i - 1)(1 < i < K + 1)$. (c) Transitions for state $(i, j)(2 < i < K + 1, 0 < j < i - 1)$.

Fig. 3(c) illustrates the transitions for state (i, j) where $2 < i < K + 1, 0 < j < i - 1$. The process may move into state (i, j) from two states $(i - 1, j)$ and $(i - 1, j)^*$ with rate λ, respectively. The process moves from (i, j) to $(i, j + 1)^*$ or $(i + 1, j)$ with rates μ and λ, respectively. The transition equation is

$$\pi(i,j) = \frac{\lambda}{\lambda + \mu}[\pi(i-1,j) + \pi^*(i-1,j)], \qquad 2 < i < K+1, 0 < j < i-1. \tag{19}$$

Fig. 4(a) illustrates the transitions for state $(K + 1, j)$ where $0 < j < K + 1$. Note that if a phone call arrives when the process is in (K, j) or $(K, j)^*$, the system enters a new cycle (with rate λ), and we could represent the new state as (1, 0). In our model, we introduce the state $(K + 1, j)$ instead of (1, 0), where $\Sigma_{0 \le j \le K}\, \pi(K+1, j) = \pi(1, 0)$, so that the hit ratio, and thus the location tracking cost, can be derived [see (24)]. The process moves from $(K + 1, j)$ [i.e., (1, 0)] to (1, 1)* with rate μ if the user moves before the next call arrives. Otherwise, the process moves to (2, 0) with rate λ. The transition equation is

$$\pi(K+1,j) = \frac{\lambda}{\lambda + \mu}[\pi(K,j) + \pi^*(K,j)], \qquad 0 < j < K+1. \tag{20}$$

For $j = 0$, the transition from $(K, j)^*$ to $(K + 1, 0)$ should be removed in Fig. 4(a) because the state $(K, 0)^*$ does not exist. The transition equation for $(K + 1, 0)$ is given in (17). Fig. 4(b) illustrates the transitions for state $(i, j)^*$ where $0 < j \le i, 1 < i < K + 1$. The process can only move to $(i, j)^*$ from $(i, j - 1)$ (with rate μ). From the definition of $(i, j)^*$, if the user moves when the process is in $(i, j)^*$, the process remains in $(i, j)^*$ (with rate μ). Otherwise, the process moves to $(i + 1, j)$ with rate λ. The transition equation is

$$\pi^*(i,j) = \frac{\mu}{\lambda}\pi(i, j-1), \qquad 0 < j \le i, 1 < i < K+1, i \ge 2. \tag{21}$$

The transitions for (2, 0) are similar to the transitions for $(i, 0)$ except that the transition from (1, 0) is replaced by $(K + 1, 0), \cdots, (K + 1, K)$ [cf., Fig. 4(c)]. The transition equation is

$$\pi(2,0) = \frac{\lambda}{\lambda + \mu}\left[\sum_{0 \le j \le K} \pi(K+1,j)\right]. \tag{22}$$

Finally, the transition for (1, 1)* is similar to the transitions for $(i, j)^*$ except that the transition from (1, 0) is replaced by $(K + 1, 0), \cdots, (K + 1, K)$ [cf., Fig. 4(d)]. The transition equation is

$$\pi^*(1,1) = \frac{\mu}{\lambda}\left[\sum_{0 \le j \le K} \pi(K+1,j)\right]. \tag{23}$$

Suppose that at the beginning of a cycle, the process is in state $(K + 1, j)$, then it implies that there are j cache misses in the previous cycle. The cache is enabled if and only if

$$p_M \le p_T = \frac{C_H}{C_B} \Rightarrow 1 - \frac{j}{K} \ge \frac{C_H}{C_B}$$
$$\Rightarrow 0 \le j \le \left\lceil K\left(1 - \frac{C_H}{C_B}\right)\right\rceil.$$

Thus, the probability that the measured hit ratio $p_M < p_T$ in the previous cycle is

$$\Pr[p_M < p_T] = \frac{\displaystyle\sum_{\lceil K(1-(C_H/C_B))\rceil < j \le K} \pi(K+1,j)}{\displaystyle\sum_{0 \le j \le K} \pi(K+1,j)}$$

and the location tracking cost for the reset-K algorithm is

$$C_C = C_B \Pr[p_M < p_T] + (1 - \Pr[p_M < p_T]) \cdot \left\{ \sum_{0 \le j \le K} \left(\frac{(K-j)C_H}{K} + \frac{j(C_H + C_B)}{K} \right) \cdot \left[\frac{\pi(K+1,j)}{\sum_{0 \le j \le K} \pi(K+1,i)} \right] \right\}. \tag{24}$$

The first term in (24) represents the cost incurred when caching is disabled because the hit ratio threshold exceeds the hit ratio measured in the previous cycle. The second term is the cost when the cache is enabled, and consists of two parts, corresponding to calls during which hits occur and calls during which misses occur. The ratio in square brackets is the conditional probability of being in state $\pi(K + 1, j)$ during the current cycle.

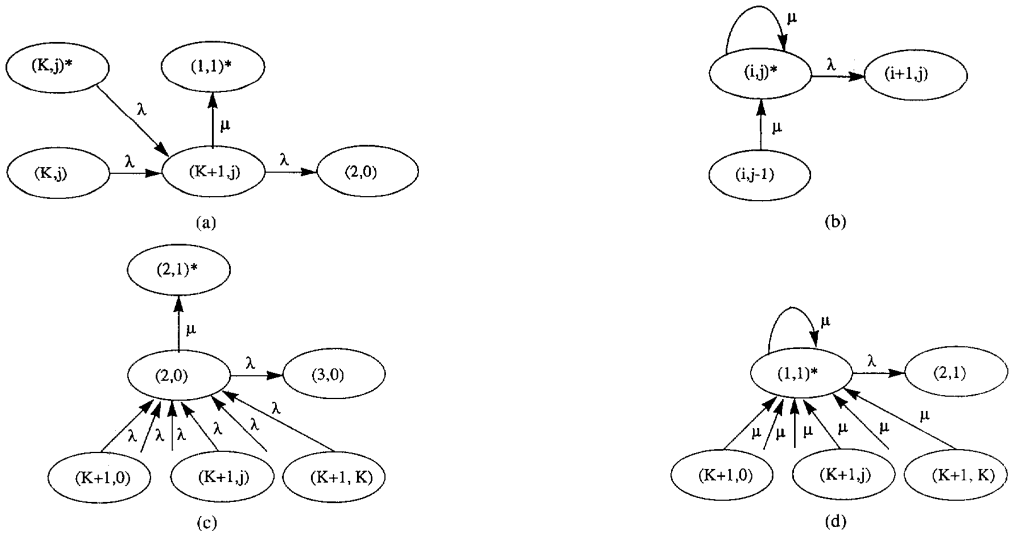


Fig. 4. (a) Transitions for state $(K+1, j)(0 < j < K+1)$. (b) Transitions for state $(i,j)^*(0 < j \leq i, 1 < i < K+1)$. (b) Transitions for state (2, 0). (d) Transitions for state (1, 1)*.

$\pi(K+1, j)$ can be computed numerically as follows. First compute $a_{i,j}$ and $b_{i,j}$ where $\pi(i,j) = a_{i,j}\pi^*(1,1)$ and $\pi^*(i,j) = b_{i,j}\pi^*(1,1)$. Note that $a_{i,j} = 0\,(b_{i,j} = 0)$ if $\pi(i,j)\,(\pi^*(i,j))$ is not defined in (17)–(23). Since $\Sigma_{i,j}\,[\pi(i,j) + \pi^*(i,j)] = 1$ we have

$$\pi^*(1,1) = \frac{1}{\sum\limits_{i,j}(a_{i,j} + b_{i,j})}$$

and $\pi(K+1, j)$ can be computed and the location tracking cost for the reset-K algorithm is obtained using (24).

The analysis is validated by a Monte Carlo simulation. In the simulation, the confidence interval for the 98% confidence level of the output measure C_C/C_B is within 3% of the mean value. Fig. 5 plots curves for (24) (the solid curves) against the simulation experiments (the dashed curves) for $K = 20$ and $C_H = 0.5C_B$ and $0.3C_B$, respectively. The figure indicates that the analysis is consistent with the simulation model.

C. Comparison of the LCMR Estimation Algorithms

If the distributions for the incoming call process and the user movement process never change, then we would expect the running average algorithm to outperform the reset-K algorithm (especially when K is small) because the measured hit ratio p_M in the running average algorithm approaches the true hit ratio value p in the steady state. Surprisingly, the performance for the reset-K algorithm is roughly the same as the running average algorithm even if K is as small as 10. Fig. 6 plots the location tracking costs for the running average algorithm and the reset-K algorithm with different K values.

The figure indicates that in steady state, when the distributions for the incoming call process and the user movement process never change, the running average algorithm outperforms reset-K, and a large value of K outperforms a small K, but the differences are insignificant.

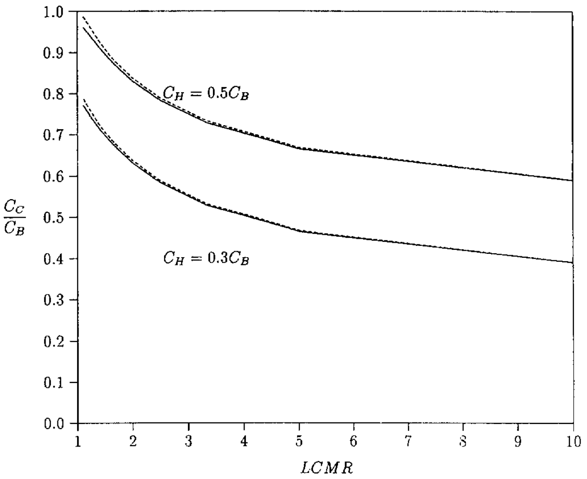


Fig. 5. The location tracking costs for the reset-K algorithm ($K = 20$).

If the distributions for the incoming call process or the user movement process change from time to time, we expect that the reset-K algorithm outperforms the running average algorithm. We have examined this proposition experimentally. In the experiments, 4000 incoming calls are simulated. The call arrival rate changes from 0.1 to 1.0, 0.3, and then 5.0 for every 1000 calls (other sequences have been tested and similar results are observed). For every data point, the simulation is repeated 1000 times to ensure that the confidence interval for the 98% confidence level of the output measure C_C/C_B is within 3% of the mean value. Fig. 7 plots the location tracking costs for the two algorithms for these experiments. By changing the distributions of the incoming call process, we observe that the reset-K algorithm is better than the running average algorithm for all C_H/C_B values.

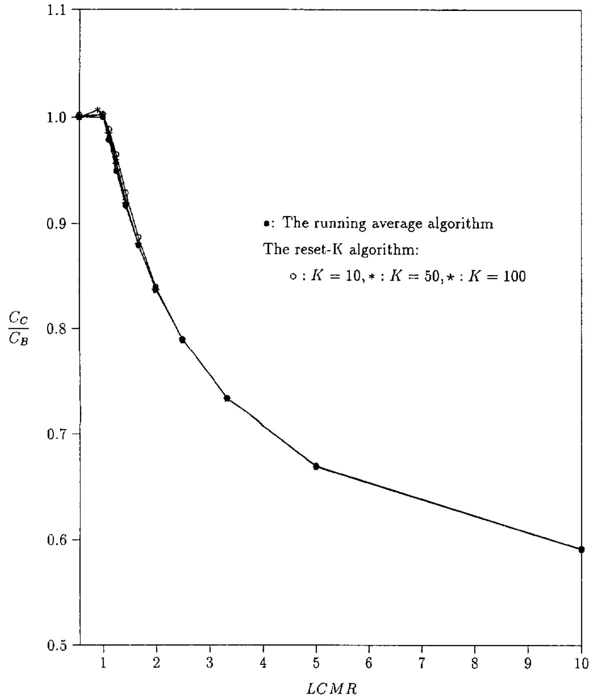


Fig. 6. The location tracking costs for the running average algorithm and the reset-K algorithm ($C_H = 0.5C_B$).

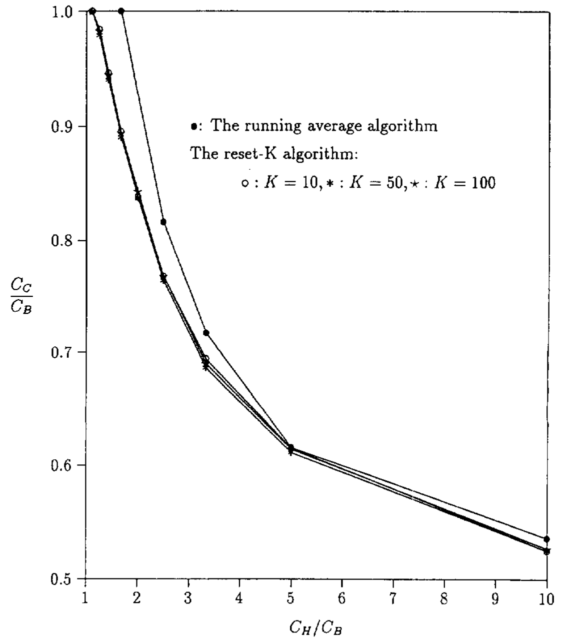


Fig. 7. Comparing the running average algorithm and the reset-K algorithm under unstable call traffic.

VI. Discussion

In this section we discuss aspects of the caching strategy presented here. Caching in PCS systems raises a number of issues not encountered in traditional computer systems, particularly with respect to architecture and locality in user call and mobility patterns. In addition, several variations in our reference assumptions are possible for investigating the implementation of the caching strategies. Here we sketch some of the issues involved.

A. Conditions When Caching Is Beneficial

We summarize the conditions for which the auxiliary strategies are worthwhile, under the assumptions of our analysis.

The caching strategy is very promising when the HLR update or query load, or the remote A-link, is the performance bottleneck, since a low LCMR (LCMR > 0) is required. For caching, the total database load and signaling network traffic is reduced whenever there is a cache hit. In addition, load and traffic is redistributed from the HLR and higher-level SS7 network elements (RSTP, D-links) to the VLR's and lower levels where excess network capacity may be more likely to exist. If the VLR is the performance bottleneck, the caching strategy is not promising unless the VLR capacity is upgraded.

The benefits of the caching strategy depend upon user call and mobility patterns when the D-link, RSTP, and LSTP are the performance bottlenecks. We have used a Poisson call arrival model and exponential inter-move time to estimate this dependence. Under very conservative assumptions, for caching to be beneficial requires relatively high LMCR's (25–50); we expect that in practice this threshold could be lowered significantly (say, LCMR > 7). Further experimental study is required to estimate the amount of locality in user movements for different user populations to investigate this issue further. It is possible that for some classes of users, data obtained from active badge location system studies (e.g., [5]) could be useful. In general, it appears that caching could also potentially provide benefits to some classes of users even when the D-link, RSTP or the LSTP are the bottlenecks.

We observe that more accurate models of user calling and mobility patterns are required to help resolve the issues raised in this section. We are currently engaged in developing theoretical models for user mobility and estimating their effect on studies of various aspects of PCS performance [9].

B. Alternative Network Architectures

The reference architecture we have assumed (Fig. 1) is only one of several possible architectures. It is possible to consider variations in the placement of the HLR and VLR functionality, (e.g., placing the VLR at a Local SCP associated with the LSTP instead of at the SSP), the number of SSP's served by an LSTP, the number of HLR's deployed, etc. It is quite conceivable that different regional PCS service providers and telecommunications companies will deploy different signaling network architectures as well as placement of databases for supporting PCS within their serving regions [18]. It is also possible that the number and placement of databases in a network will change over time as the number of PCS users increases.

Rather than consider many possible variations of the architecture, we have selected a reference architecture to illustrate the new auxiliary strategy and our method of calculating its costs and benefits. Changes in the architecture may result in minor variations in our analysis but may not significantly affect our qualitative conclusions.

C. LCMR Estimation and Caching Policy

It is possible that for some user populations, estimating the LCMR may not be necessary, since they display a relatively high average LCMR. For some populations, as we have shown in Section V, obtaining accurate estimates of user LCMR in order to decide whether or not to use caching can be important in determining the net benefits of caching.

In general, schemes for estimating the LCMR range from static to dynamic, and distributed to centralized. We have presented two simple distributed algorithms for estimating LCMR, based on a long-range and short-range running calculation; the former is preferable if the call and mobility pattern of users is fairly static, while the latter is preferable if it is variable. Tuning the amount of history which is used to determine whether caching should be employed for a particular user is an obvious area for further study, but which is outside the scope of this paper.

An alternative approach is to utilize some user-supplied information, by requesting profiles of user movements, (e.g., see [20], [7]) and to integrate this with the caching strategy. A variation of this approach is to use some domain knowledge about user populations and their characteristics.

A related issue is that of cache size and management. In practice it is likely that the monetary cost of deploying a cache may limit its size. In that case cache entries may not be maintained for some users; selecting these users carefully is important to maximize the benefits of caching. Note that the cache hit ratio threshold cannot necessarily be used to determine which users have cache entries, since it may be useful to maintain cache entries for some users even though their hit ratios have temporarily fallen below the threshold. A simple policy which has been found to be effective in computer systems is the *least recently used* (LRU) policy [19], in which cache entries which have been least recently used are discarded, and may offer some guidance in this context.

VII. Conclusions

Previous studies [11], [14], [10], [12] of PCS-related network signaling and data management functionalities suggest a high level of utilization of the signaling network in supporting call and mobility management activities for PCS systems. We have presented an auxiliary strategy, called *per-user caching,* to augment the basic user location strategy proposed in standards such as GSM and the North American IS-41 cellular standard [3], [17], [16].

For a given PCS system architecture, we have quantified the criteria under which the caching strategy produces reductions in the network signaling and database loads in terms of users' LCMR's. We have shown that if the HLR or the remote A-link in an SS7 architecture is the performance bottleneck, caching is useful regardless of user call and mobility patterns. If the D-link, or STP's are the performance bottlenecks, caching is potentially beneficial for large classes of users, particularly if they display a degree of locality in their call reception patterns. Depending upon the numbers of PCS users who meet these criteria, the system-wide impacts of these strategies could be significant. For instance, for users with LCMR ~ 5 and stable call and move patterns, caching can result in cost reductions of 20–60% over the basic user location strategy, *BasicFIND( ),* under our analysis. Our results are conservative in that the *BasicFIND( )* procedure we have used for comparison purposes already reduces the network impacts compared to the user location strategy specified in PCS standards such as IS-41.

We have also investigated in detail two simple on-line algorithms for estimating users' LCMR's and examined the call and mobility patterns for which each would be useful. The algorithms allow a system designer to tune the amount of history used to estimate a users' LCMR, and hence attempt to optimize the benefits due to caching.

The particular values of cache hit ratios and LCMR thresholds will change with variations in the way the PCS architecture and the caching strategy is implemented, but our general approach can still be applied. There are several issues deserving further study with respect to deployment of the caching strategy, such as the effect of alternative PCS architectures, integration with other auxiliary strategies such as the use of user profiles, and effective cache management policies.

Recently we have augmented the work reported in this paper by a simulation study in which we have compared the caching and basic user location strategies [6]. The effect of using a time-based criterion for enabling use of the cache has also been considered [8]. We are currently investigating the use of an auxiliary strategy involving a system of forwarding pointers to reduce the signaling traffic and database loads for users with low CMR's [7], [9]. We are also investigating the effect of alternative user mobility models on performance analysis of PCS systems [9].

Acknowledgment

We thank T. Ikuenobe and A. Knapp for providing valuable feedback, particularly during the early stages of this work. S. Levenson and J. Rizzo also provided helpful comments. Discussions with A. Atai, B.-R. Chen, E. Cohen, N. Crystal, D. Lukacs, S. Lin, J. Tanzini, S. Wainberg, and R. White were very useful. We thank D. Ghosal, E. Lipper, T. Noerpel, and A. Ranade for their comments on an earlier draft of this paper. Finally, we thank R. Wolff for his insightful comments and encouragement.

References

[1] B. Awerbuch and D. Peleg, "Concurrent online tracking of mobile users," in *Proc. SIGCOM Symp. Comm. Arch. Prot.,* Oct. 1991.

[2] Bellcore, "Switching system requirements for interexchange carrier interconnection using the integrated services digital network user part (ISDNUP)," Tech. Reference TR-NWT-000394, Bellcore, Dec. 1992.

[3] EIA/TIA, "Cellular radiotelecommunications intersystem operations," Tech. Rep. IS-41 (Revision B), EIA/TIA, July 1991.

[4] W. Feller, *An Introduction to Probability Theory and Its Applications.* New York: Wiley, 1966.

[5] N. Fishman and M. Mazer, "Experience in deploying an active badge system," in *Proc. Globecom Workshop Networking for Pers. Commun. Appl.,* Dec. 1992.

[6] H. Harjono, R. Jain, and S. Mohan, "Analysis and simulation of a cache-based auxiliary location strategy for PCS," *IEEE Conf. Networks Pers. Commun.*, 1994.

[7] R. Jain, "A classification scheme for user location strategies in personal communications services systems," Aug. 1993, submitted for publication.

[8] Y.-B. Lin, "Determining the user locations for personal communications networks," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 466–473, 1994.

[9] R. Jain and Y.-B. Lin, "An auxillary user location strategy employing forwarding pointers to reduce network impact of PCS," submitted for publication, 1994.

[10] C. Lo, S. Mohan, and R. Wolff, "Performance modeling and simulation of data management for personal communications applications," Special Rep. SR-TSV-002424, Bellcore, Nov. 1992.

[11] C. N. Lo, R. S. Wolff, and R. C. Bernhardt, "An estimate of network database transaction volume to support personal communications services," in *Proc. Int. Conf. Univ. Pers. Comm.*, 1992.

[12] C. Lo and R. Wolff, "Estimated network database transaction volume to support wireless personal data communications applications," in *Proc. Int. Conf. Commun.*, May 1993.

[13] E. Lycksell, "GSM system overview," Tech. rep., Swedish Telecom. Admin., Jan. 1991.

[14] K. Meier-Hellstern and E. Alonso, "The use of SS7 and GSM to support high density personal communications," in *Proc. Int. Conf. Commun.*, 1992.

[15] A. R. Modaressi and R. A. Skoog, "Signalling system no. 7: A tutorial," *IEEE Commun. Mag.*, pp. 19–35, July 1990.

[16] S. Mohan and R. Jain, "Two user location strategies for PCS," *IEEE Pers. Commun. Mag.*, premiere issue, First Quarter, 1994.

[17] M. Mouly and M. B. Pautet, *The GSM System for Mobile Communications*, M. Mouly, 49 rue Louise Bruneau, Palaiseau, France, 1992.

[18] P. Russo, K. Bechard, E. Brooks, R. L. Corn, W. L. Honig, R. Gove, and J. Young, "IN rollout in the United States," *IEEE Commun. Mag.*, pp. 56–63, Mar. 1993.

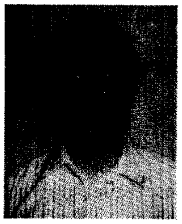
[19] A. Silberschatz and J. Peterson, *Operating Systems Concepts.* Reading, MA: Addison-Wesley, 1988.

[20] S. Tabbane, "Comparison between the alternative location strategy (AS) and the classical location strategy (CS)," Tech. rep., Rutgers Univ. WINLAB, July 1992.

Ravi Jain (S'84–M'92) received the Ph.D. degree in computer science from the University of Texas at Austin in 1992.

Prior to his Ph.D. degree he worked at Syntrex Inc., SES Inc., and the Schlumberger Laboratory for Computer Science on developing communications and systems software, performance modeling, and parallel programming. In 1992 he joined Bellcore, where his research interests include design and analysis of algorithms and techniques for efficient resource management, focusing on reducing network impacts of supporting PCS, as well as Intelligent Vehicle-Highway Systems (IVHS) applications such as the SCOUT traveler information system. His activities also include research in mobile and parallel computing.

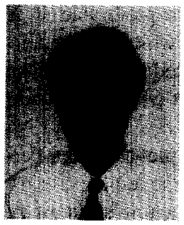
Dr. Jain is co-chair of the annual Workshop on Input/Output in Parallel Computer Systems held in conjunction with the International Parallel Processing Symposium. He has several publications in the area of communications and parallel computing and is a member of the Upsilon Pi Epsilon and Phi Kappa Phi honorary societies, as well as ACM, ORSA, and CPSR.



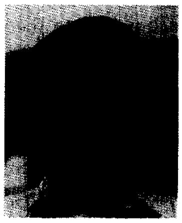
Yi-Bing Lin received the B.S.E.E. degree from National Cheng Kung University in 1983, and the Ph.D. degree in computer science from the University of Washington in 1990.

Since then, he has been with the Applied Research Area at Bell Communications Research (Bellcore), Morristown, NJ. His current research interests include design and analysis of PCS networks, distributed simulation, and performance modeling.

Dr. Lin is the Guest Editor of the Special Issue on Simulation of Communication Systems for the *International Journal of Computer Simulation,* the Guest Editor of the Special Issue on Parallel and Distributed Simulation for the *Journal of Parallel and Distributed Computing,* the editor of a book "Advanced Topics in Distributed Simulation," an associate editor of the *International Journal of Computer Simulation,* and Program Chair for the 8th Workshop on Distributed and Parallel Simulation. Dr. Lin is an Adjunct Research Fellow at the Center for Telecommunications Research, National Chiao-Tung University, Taiwan, R.O.C.



Charles Lo received the B.S. and M.S. degrees in electrical engineering from Cornell University. He is currently Manager of Internetworking at AirTouch Communications' (formerly PacTel Corp.) Wireless Data Division, Walnut Creek, CA, where he is leading the development and implementation of advanced cellular data communications infrastructure and solutions. Prior to joining AirTouch, Charles spent nine years as Member of Technical Staff in Applied Research at Bellcore, Morristown, NJ, where his activities have included PCS, Intelligent Networks, BISDN, and lightwave transmission technology and systems. Before Bellcore, he worked for three years on terrestrial lightwave transmission systems development at AT&T Bell Laboratories, Holmdel, NJ.



Seshadri Mohan (S'76–M'79) received the Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, Ont., Canada, in 1980.

Since receiving the Ph.D. degree, he has worked for Bell Laboratories, Holmdel, NJ, and has taught at Wayne State and Clarkson Universities, where he conducted research in the area of coding algorithms, and computer communications and multiple access protocols. He joined Bellcore in 1990, where he is currently a Member of the Network Architecture and Analysis Research Laboratory. His current research interests include designing efficient data management and location strategies for the support of personal communications applications and investigating the applicability of distributed computing and communications architectures for nomadic personal communications and information networking. He has coauthored the text titled *Source and Channel Coding: An Algorithmic Approach* (Kluwer Academic). He is a member of the editorial board of IEEE Personal Communications: The Magazine of Communications and Computing. He is a member of the ACM.