Phone Number Portability for PCS Systems with ATM Backbones Using Distributed Dynamic Hashing

Ravi Jain, *Member, IEEE,* Subhashini Rajagopalan,
and Li Fung Chang, *Senior Member, IEEE*

Abstract*— Current cellular subscribers have a geographic phone number (e.g., in AMPS and U.S. digital cellular systems) or a number which contains the network provider's identity (e.g., in GSM), and whenever subscribers register or receive (and possibly, originate) a call, a home location register (HLR) database has to be queried. The wired infrastructure supports a function called global title translation (GTT) that converts the subscriber's number to an HLR database address. A special feature of next generation wireless access service will be to support personal communication services (PCS's) and wireless subscribers with portable personal numbers, or nongeographic phone numbers (NGPN's), that do not indicate the service provider or HLR database serving the user. In addition, the GTT function may not be available when the wired backbone is an ATM network. Thus a key function required in future wireless access systems with wired ATM backbones will be the ability to translate an NGPN to the identity of the HLR which serves the subscriber, a process we call NGPN translation. (Note that the same functionality is also needed for subscribers with local, portable phone numbers.) We discuss the requirements of NGPN translation and some alternative schemes. We propose two schemes for fast, efficient, scalable and flexible NGPN translation which use ideas of dynamic hashing, caching, and indirection. The schemes use a hash function in the visiting location registers (VLR's) (or serving SCP) and a set of distributed translation servers which store the NGPN-to-HLR mapping. We discuss how the operations required to maintain the translation information can be performed. Finally, we perform a simplified analysis of the scalability of the alternative schemes as well as the hash-based schemes we propose.

***Index Terms*—Home location register, global title translation, visiting location registers.**

I. Introduction

THE PROBLEM we address is to devise an efficient, flexible, and extensible means of determining the identity of the signaling network database which contains the service profile of a personal communications services (PCS's) subscriber, when the only relevant information available is the subscriber's nongeographic phone number (NGPN).

Currently, fixed telephone subscribers are assigned a *geographic* phone number, which contains enough information to determine how the signaling messages required to set up a call to the subscriber are to be routed through the signaling network. Current cellular subscribers have a geographic phone number (e.g., in North American AMPS and U.S. digital cellular systems) or a number which contains the network provider's identity (e.g., in GSM) and a process called global title translation (GTT) is used to determine how the call is to be routed. GTT is executed at signaling switches called signaling transfer points (STP's), and essentially translates information obtained from a subscriber's phone number to the identity of the home location register (HLR) database which serves that subscriber, i.e., contains the subscriber's service profile information. For next-generation wireless-access service, however, subscribers will be assigned NGPN's (e.g., 1-500-XXX-XXXX), which do not contain any information specifying the geographic region *or* network provider serving that subscriber. Furthermore, for future PCS systems in which the wired backbone is an ATM network, STP's may not be used for signaling and GTT cannot be performed.

We call the process of translating a NGPN to the correct database identity *NGPN translation.* The problem requires that NGPN translation be fast and efficient. The problem is made further complicated by the following distinguishing requirements: 1) the translation has to be performed by network entities (e.g., visiting location registers, VLR's) which are widely *distributed,* 2) the translation scheme has to be *scalable* as the number of subscribers and signaling databases grows, and 3) the translation has to be flexible so that changes in the NGPN-to-database mapping can be accommodated *dynamically,* without halting the operation of the overall system. It is these three requirements which distinguish the problem, and our solution, from previous work in the area.

Although this paper is written using the specific application of NGPN translation for PCS systems with ATM backbones, we note that it is applicable to a variety of situations where number translation has to take place. Examples include local phone number portability (i.e., translating a subscriber's phone number to the ID of the service provider serving that subscriber), 800 number portability, Internet Protocol (IP) address and domain name translation, and cell relay address translation.

A. Organization of the Paper

In the rest of this section, we present the background and motivation for this work, including the reasons for multiple

Manuscript received February 1, 1996; revised June 1996.

The authors are with Applied Research, Bell Communications Research, Red Bank, NJ 07960 USA.

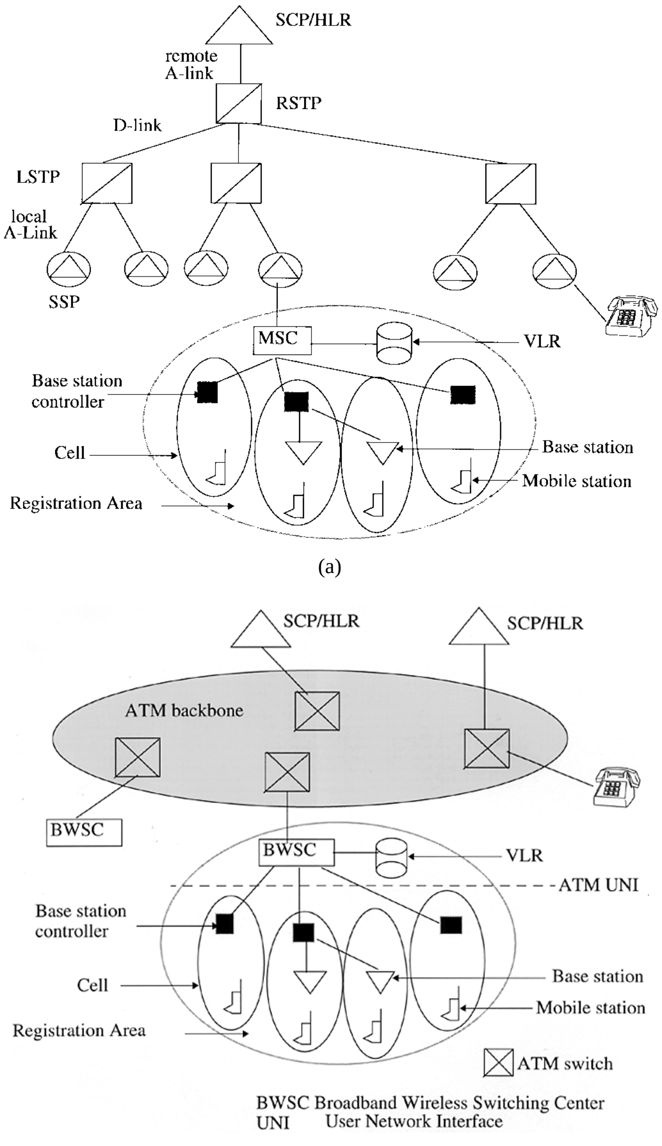
Publisher Item Identifier S 0733-8716(97)00057-7.

HLR's in the network and the requirements of the NGPN translation process. In Section II, we consider some possible alternative architectures for NGPN translation, including a completely distributed approach (store the NGPN-to-HLR mapping in the VLR's) and a completely centralized approach. We discuss the drawbacks with these alternatives. In Section III, we present our first scheme for NGPN translation, which combines ideas of distributed dynamic hashing, caching, and indirection. In Section IV, we present our second scheme for NGPN translation, which is similar to the first but offers greater scalability at the expense of increased delay for some number translations. In Section V, we develop a simplified model of the peak NGPN translation load for a hypothetical metropolitan area and evaluate this model for certain parameter ranges in order to study the scalability of our hash-based scheme as well as some of the alternatives. In Section VI, we end with a brief discussion of how our hash-based scheme differs from those used in traditional file system applications, its applicability to other telecommunications problems such as Internet domain name and IP address translation, and end with a summary.

B. Background and Motivation

1) Network Architecture Model: The network architecture for offering PCS service, as embodied in current North American and European standards, involves maintaining the location of the PCS terminal (also called handset or subscriber unit), as well as the subscriber's profile and authentication information, using a two-level hierarchy of databases, i.e., HLR's and VLR's. We assume the reader has a basic familiarity with the current architecture of PCS systems and the process of registering the locations of mobile users and delivering calls to them; for a tutorial, see [1]. An example of the current network infrastructure design is shown in Fig. 1(a). Here the end-office switches, called service switching points (SSP) are connected via signaling links called A-links to local signaling transfer points (LSTP), which in turn are connected via D-links to a regional signaling transfer point (RSTP). The service control point (SCP) database is connected to the RSTP via a signaling link called a remote A-link and is assumed to contain the HLR. Each SSP is assumed to be connected to a mobile switching center (MSC) which serves a number of registration areas (RA), and the VLR is assumed to be connected to the MSC. For further details, please see [1].

The signaling and control traffic carried over the current signaling network [shown in Fig. 1(a)] is expected to grow rapidly due to increased service demands for existing services (e.g., 800 number service, alternate billing service), and the introduction of new services such as PCS and video dial tone. The current SS7 signaling network may not be adequate to support the stringent delay requirements imposed by broadband and PCS services [2]–[5]. For future network architectures in which ATM is used for the backbone, it has been considered that it may be cost-effective and efficient to integrate signaling traffic with the user data traffic on the same physical network. Some research on carrying PCS signaling traffic on an ATM transport platform has been reported recently [6]–[10].



(a)

(b)

Fig. 1. (a) Current (non-ATM) network reference architecture for PCS and (b) ATM-backbone network reference architecture for PCS.

We consider the situation where PCS signaling traffic is integrated with user traffic on an ATM backbone network. The network reference architecture we assume is shown in Fig. 1(b). The architecture contains a network entity called a broadband wireless service center (BWSC) which replaces the current MSC. A BWSC is a switching center capable of providing mobility support to mobile terminals and connects to the ATM backbone network through an ATM interface; it may be regarded as an ATM switch enhanced with mobility management and other functions required to handle PCS calls. Fig. 1(b) shows the VLR connected to the BWSC, but it may instead be connected to an ATM switch in the ATM backbone. Our reference architecture for PCS systems with ATM backbones assumes that both signaling and user data are transported on the same physical ATM backbone network (although in separate logical channels). The ATM user-network interface (UNI) is between the BWSC and the base station controller. In order to keep the present paper focused on the database and number translation issues, we will omit further discussion of the future network architecture(s) for supporting PCS and wireless subscribers when the backbone network is an ATM network; the reader is referred to [9] and [10].

For PCS systems, we anticipate that more than one HLR will be required, for several reasons. First, different service providers may own their own HLR's, storing the information of their own PCS subscribers only. Second, as the number of subscribers increases, and/or the amount of information stored for each subscriber increases, a single HLR will run out of storage capacity or processing capability. Finally, to take advantage of geographical locality multiple HLR's may be deployed, one for each geographical region, so as to reduce the communication costs of serving subscribers whose home locations are in that region.

2) Signaling Message Routing and Number Translation: We now sketch how signaling messages are generated and routed in current PCS system designs and discuss the issues raised when the wired network migrates from the current architecture to an ATM-backbone architecture.

In the current design for PCS service, a process called global title translation is carried out at the STP's to handle the routing of PCS signaling messages associated with registration and call delivery. GTT may be required in several situations: 1) when a PCS subscriber crosses registration areas (RA's) served by different VLR's, the ID of its serving HLR needs to be determined in order to update the subscriber's location information; 2) when a PCS subscriber is called (by a fixed or PCS subscriber), its HLR ID is required to set up the call; 3) in some implementations, the subscriber's HLR ID may also be required when a PCS subscriber originates a call, in order to obtain authentication and service profile information.

As an example, consider the procedure when a subscriber moves to a new RA and the subscriber's PCS terminal registers. The registration message contains the terminal's mobile identification number (MIN), which is essentially the subscriber's phone number, and is forwarded to the VLR by the BWSC. If the VLR does not have the identity of the HLR serving this subscriber, it forward the MIN (and other information) to an STP equipped to perform GTT, along with a translation type indicating PCS service. The STP uses this information to route the query to an HLR. (Further details of the GTT process can be found in [11].)

As discussed previously, in future we expect that there will be multiple HLR's which could be queried, both because there will be multiple service providers which can provide service to the user (even within a single region) and because each provider may own multiple HLR's. In addition, we expect that the subscribers's terminal MIN may be "nongeographic," i.e., it may not contain any information indicating which geographical region or which provider serves that user; we will say that the PCS subscriber has an NGPN. Thus, in our registration example above, when the VLR is presented with an NGPN, it must determine which HLR is to be queried to complete the process.

If all the HLR's contain identical information, then GTT can be used: the VLR forward the NGPN and the translation type to the STP, which ignores the NGPN and routes the query to any suitable HLR. However, there are two problems with this approach. First, for various reasons, it is likely that service providers may wish to keep the service profiles of their own subscribers private and thus may not be willing for other service providers to keep copies of this information. Second, it is likely that with the migration to an ATM-backbone network in which signaling and user data transport are integrated, STP's and the GTT function will be eliminated.

3) NGPN Translation Requirements and Scalability: In this paper, we will simplify the discussion by assuming that whenever NGPN translation is required as discussed above, the NGPN is presented to a VLR, which has the burden of performing a translation or obtaining a translation from other entities in the network. This is accurate for registration and call originations by a PCS subscriber. When a call is to be delivered to a PCS subscriber, however, the NGPN may be presented to some other entity (e.g., an SSP or serving SCP) instead. In this paper, it is understood that for these situations, the operations required to perform NGPN may in fact have to be performed at a serving SCP instead of a VLR.

In general, there will be two types of operations which have to be supported by any NGPN scheme: translation operations and service operations. The translation operation is the actual translation of an NGPN to an HLR ID. Since NGPN translation is required at call setup time, when it is only one of several steps required, it is important that the translation process be fast. It is also desirable that the process be efficient, both in terms of signaling network traffic and database loads. Another desirable feature for any proposed NGPN translation scheme is that it should permit NGPN's to be chosen by an independent entity (e.g., a numbering plan administrator) without considering how they are to be mapped to HLR ID.

Service operations can be further divided into three types, i.e., those required to: a) initiate or terminate NGPN service to a user, or modify the user's NGPN-to-HLR mapping, b) increase the number of HLR's, and c) increase the number of databases used to store the NGPN-to-HLR mapping.

Among service operations of type A, initiating and terminating NGPN service to a user are self-explanatory. It is expected that the NGPN-to-HLR mapping may have to be changed from time to time for the following reasons: 1) a subscriber changes his or her service provider, and hence needs to be mapped to a different HLR, 2) an HLR becomes saturated (runs out of storage or processing capacity) so that some load has to be diverted, and 3) a subscriber moves semipermanently from one geographical region to another, so that it is more efficient to serve the subscriber from another HLR. It is possible that with increased competition and increased use of PCS as well as NGPN's, modifying the NGPN-to-HLR mapping will become increasingly frequent. Thus, changing the NGPN-to-HLR mapping should be a fast and efficient operation.

Service operations of type B occur because the number of PCS subscribers (and not necessarily NGPN subscribers) increases, or because new service providers (with their own

HLR's) enter the market. Service operations of type C occur because the number of NGPN users increases.

It is highly desirable that not only the translation operation, but the service operations be fast and efficient. In this paper, in general, we consider a scheme to be *scalable* if both translation operations and service operations remain relatively fast and efficient as the workload due to the operations increases for any of the reasons discussed above.

II. Some Alternative Schemes

We discuss three alternative schemes for NGPN translation and discuss their pros and cons. In Section II-A, we consider a completely distributed approach to storing the NGPN-to-HLR mapping, and in Section II-B, we consider a completely centralized approach. In the Section III, we will describe the scheme we propose, which is based upon hashing.

A. Store Mapping in VLR's

A simple method of performing NGPN translation is as follows. Every VLR is loaded with a table which specifies a mapping from a NGPN to the ID of the HLR serving that NGPN. An operations support system (OSS) can be set up to allow the VLR's to be loaded and maintained. Now when a switch is presented with an NGPN it simply forward it to the VLR, which contains all the information required to map the NGPN to an HLR ID.

This method is straightforward, and the NGPN translation is fast. However, it has a couple of drawbacks. First, every VLR must store the mapping for all the NGPN's, since it cannot be predicted from which RA a call to a given subscriber may originate, or to which RA a subscriber may roam. This duplication of information entails a storage cost which, while not prohibitive, may be nontrivial when added up for all the VLR's in the country. More importantly, when the NGPN-to-HLR mapping for a subscriber needs to be modified, then all the VLR's have to be updated; this represents a significant overhead, as well as a significant concurrent database update problem for the OSS.

B. Translation Servers

The drawbacks with the scheme of storing the mapping in all the VLR's can be alleviated by the use of indirection. Instead of storing the NGPN-to-HLR mapping itself at the VLR, a pointer to a translation server (TS) is stored. (Translation servers are entities not present in current and proposed PCS architectures; they are introduced for the purpose of NGPN translation.) Logically, a TS is a database which stores the actual NGPN-to-HLR mapping; physically, it may or may not be co-located with an HLR. Since TS's do not perform the other functions performed by a VLR, it is expected that there will be far fewer TS's than VLR's, and in addition, they will not be so widely dispersed. Thus, if the NGPN-to-HLR mapping has to be changed, only the TS's have to be updated, rather than every VLR, and this will be a simpler operation.

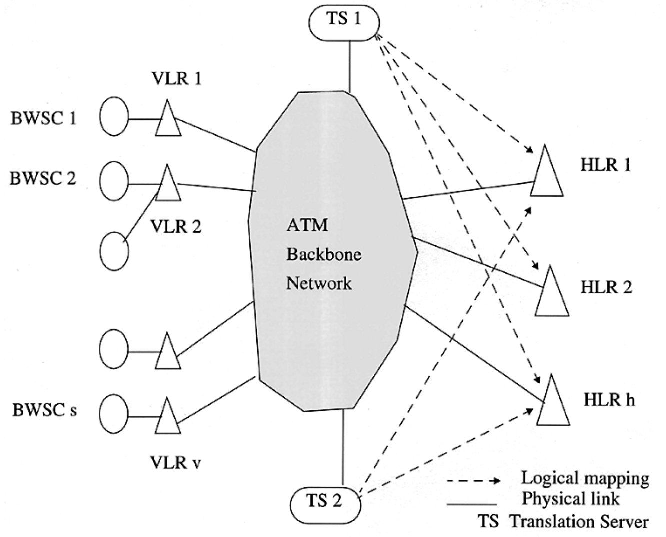


Fig. 2. High-level architecture for NGPN translation schemes.

One approach is that the NGPN-to-HLR mapping is only stored in a single, central location (in Kansas, say), in a TS database system. The TS database management system is used to load and modify the mapping. Now when a switch is presented with an NGPN it forward it to the VLR as before; the VLR queries the TS, which returns the HLR ID.

This second method is not as fast as storing the mapping in all the VLR's, but it is probably fast enough. By storing the mapping in only one place, the concurrent update problem is eliminated, and the OSS is simplified. However, this method also has a drawback, which is the lack of scalability for translation operations. As the number of PCS subscribers grows, a single TS will not be able to handle the queries for call delivery fast enough.

One way to increase the effective capacity of the central TS is by installing a parallel database machine as the TS and using caching at the VLR's. However, there will still be additional drawbacks, such as the vulnerability to a catastrophic failure at the centralized location (earthquake, fire, etc.), and the communication cost and time delay for querying the TS from all over the country. Yet another way to increase effective capacity is to use multiple, distributed TS's which are fully replicated, i.e., each TS contains the mappings for all NGPN's. However, such an architecture entails an operations support system which must be capable of modifying all the databases concurrently and efficiently for service operations.

III. First Proposed NGPN Translation Scheme

The logical architecture of the scheme is shown schematically in Fig. 2. At the center is the ATM backbone signaling network; the signaling network is composed of ATM virtual circuits, and is not physically distinct from the transport network used to actually carry user calls, data, etc. There are a number of VLR's which are connected via the signaling network to TS's as well as HLR's. Each TS contains NGPN-to-HLR mappings, and an NGPN can be translated to any of the HLR's in the system.

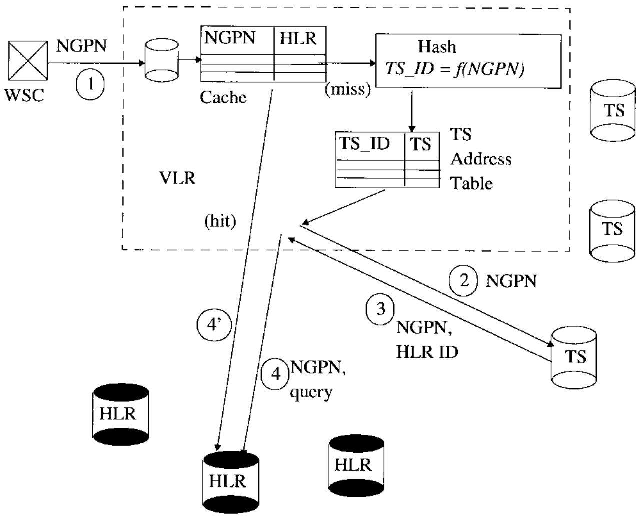


Fig. 3. First proposed NGPN translation scheme.

A. The Basic Scheme

The scheme can be briefly described as follows; see Fig. 3. We first describe the scheme without the use of the cache in the VLR.[1]

1) When any of the situations requiring NGPN translation occur, the NGPN is presented to a switch; depending upon the architecture deployed, this may be a BWSC or a SSP.
2) The switch forward the NGPN to the VLR serving that switch.
3) The VLR performs a *hash function* upon the binary representation of the NGPN, to obtain a value f(NGPN), where f is the hash function. This specifies the ID of a TS where the mapping is stored.
4) In general, the TS ID returned by the hash function will not be a network address; to obtain the network address of the TS, the VLR consults its TS address table.
5) The VLR launches a query to the TS address corresponding to TS ID, passing it the value NGPN. The TS contains the NGPN to HLR mapping.
6) The TS responds to the VLR by returning the HLR ID.
7) The VLR uses the HLR ID to continue with the registration, call delivery, or call origination signaling operations as usual.

The OSS function for the PCS system will be required to load VLR's with hash functions, manage the NGPN-to-HLR mapping in the TS's, etc., in order to support this scheme.

B. The Caching Scheme

The VLR can maintain a cache of NGPN translations to avoid querying the TS. Thus when presented with a NGPN for the first time, the VLR performs a hash and queries the indicated TS to obtain the ID of the serving HLR, as described above. It then stores the NGPN-to-HLR mapping for that NGPN in its cache. If presented with the same NGPN a second time, the VLR can search its cache first. If the mapping is found (a cache hit), a hash and a query to the TS is avoided; otherwise, (a cache miss) the VLR performs a hash and queries the TS as usual.

Similarly, the TS can also maintain a cache of NGPN translations in high-speed memory to speed up its own operation. It is expected that a simple least recently used (LRU) cache management scheme would be adequate for maintaining the cache. Thus if the cache is full, and a new NGPN is presented, the record containing the least recently used mapping is deleted and replaced by the mapping for the new NGPN. It is possible to experiment with more sophisticated schemes which attempt to use some additional information about the call reception and mobility patterns of individual subscribers to manage the cache more efficiently.

These caches can be of different sizes for different VLR (or TS) depending upon how frequently the VLR (or TS) is queried and how much memory is available, thus allowing finer-grained control of performance.

C. The Hash Function

The two key requirements of the hash function are that it should be easy to compute, and that it should result in mapping roughly the same number of NGPN's to each TS.

An example of a simple hash function is the function *even()*, which returns 0 if the argument is even and 1 otherwise. Obviously, this function can only be used if there are only two TS's. In addition, if the way in which NGPN's are chosen is nonuniform (e.g., a disproportionate number of subscribers request phone numbers ending in 0), the load on the two TS's will not be balanced; similarly, if the service provider assigns NGPN's to new subscribers using some administrative process (e.g., by reusing the NGPN's of subscribers who have canceled service) which somehow introduces some nonuniformity, the load on the TS's will not be balanced.

Numerous hash functions have been designed and evaluated for applications such as searching for a given record in a file. Some which have been found to be useful in practice include the mod function (i.e, $f(\text{NGPN}) = \text{NGPN} \bmod t$, where t is the number of TS's). The reader is referred to [12] for other functions and more details. The choice of the hash function has to be guided by the number of TS's and the way in which NGPN's are selected and assigned to subscribers. This will require information and experimentation for each specific deployment scenario.

As mentioned previously, in general the value returned by the hash function will not be a suitable signaling network address (e.g., an ATM address.) One way of mapping the output of the hash function into a signaling network address is to load a small table containing this mapping into each VLR. Note that this table will be the same for all VLR's, its size will be very small (proportional to the number of TS's, as opposed to the number of NGPN's) and it will only need to be updated if a new TS (rather than a new subscriber or HLR) is added. Another possibility may be to reserve a block of consecutive addresses for the TS's (ranging from *TS_low* to *TS_high,* say),

[1] As mentioned previously, for simplifying the presentation, we assume that NGPN's are presented to VLR's for translation; in practice, they may be presented to serving SCP's.

and then generate the actual TS address as the value *TS_low* $+f$(NGPN).

D. Service Operations

The main operations that have to be performed in order to maintain the translation information are described in decreasing order of their frequency below.

1) *Add a Subscriber:* At service provisioning time, the service profile, etc., for the subscriber is created and loaded into an HLR. The hash function is applied to the subscriber's NGPN to obtain a TS ID, and that TS [i.e., f(NGPN) is updated with a record containing the NGPN-to-HLR mapping for this NGPN].
2) *Delete a Subscriber:* At service termination time, the TS indicated by f(NGPN) is searched to find the HLR serving this NGPN; the subscriber's record in that HLR, as well as the TS, is deleted.
3) *Move a Subscriber's HLR:* At service modification time, the TS indicated by f(NGPN) is queried to find the current HLR ID. The service profile of the subscriber is transferred to the new HLR and the TS is updated with the new HLR ID.

   As the number of subscribers grows, additional operations may be required. Obviously, the following operations will be performed much less frequently than the previous three operations.
4) *Add a New HLR:* This operation will be performed relatively infrequently. When an HLR is reaching its capacity and a new HLR is installed, TS entries for new subscribers contain the mapping from the NGPN to the new HLR ID.
5) *Split an HLR into Two:* This is a major operation and will be performed infrequently. One way to accomplish this is as follows. Suppose an HLR already supports h subscribers and $n < h$ new subscribers are to be added, which would exceed its capacity. A new HLR is to be installed, such that the two HLR's will each support roughly $(h+n)/2$ subscribers. The n new subscribers are installed in the new HLR, using the procedure described above. Then $(h-n)/2$ old subscribers are moved from the old to the new HLR, as described above. Finally, the $(h-n)/2$ records of old subscribers in the old HLR are deleted. (A more efficient method can obviously be designed which only accesses each TS only once.)
6) *Split a TS into Two:* This operation is more time consuming than the previous one and will be performed even more rarely. It is the only operation which requires modification to all the VLR's.

Suppose we wish to increase the total number of TS's, from t to $t+1$, and, in order to do so efficiently, modify the contents of at most one TS. One method to do this is to apply a kind of dynamic hashing called *extendible hashing* [13]. In extendible hashing, the hash function in the VLR is assumed to return a large number of valid bits, but only k of them are used (say, the highest k bits). The k bits are used to look up the TS address table in the VLR, which maps them to the network address of the appropriate TS. When the number of TS's is to be increased to $t+1$, the number of hash bits used is increased to $k+1$.

The procedure for splitting the TS will thus be as follows. First, for each NGPN in the target TS, the hash function is applied and the $(k+1)$th bit is examined; if it is zero, the record remains in the current TS, otherwise, it is copied to the new TS. Once all the NGPN's in the TS have been processed, all the VLR's in the system have their value of k incremented and their address table modified as described above. Finally, the records in the target TS which were copied to the new TS are deleted.

E. Discussion

The NGPN translation operation in the hashing scheme, while not as fast as that for a centralized server, only involves one extra database lookup. This also holds for the other common service operations (adding and deleting subscribers, etc.). In addition, the difficult or time-consuming operations in this scheme are those which will be carried out very infrequently (e.g., splitting an HLR or a TS).

The NGPN translation scheme we have proposed addresses the following requirements of NGPN translation.

1) *Speed:* The use of the hash function, and caching if necessary, makes the NGPN translation fast. In addition, only a single query and response (VLR to TS and back) is required.
2) *Ease of Modification:* It permits the NGPN-to-HLR mapping for a given PN to be stored in only one place–a single TS. This information need not be stored in all the VLR's or all the TS's. This eliminates the concurrent update problem, and allows the NGPN-to-HLR mapping to be modified easily.
3) *Reduced Storage Space:* The NGPN-to-HLR mapping for each NGPN is only stored in one place, avoiding any duplicate storage. The hash function must be stored in all the VLR's, but this is the same for all NGPN's, and is in any case a small piece of software and TS address table, rather than a large NGPN-to-HLR table.
4) *Location Flexibility:* The location of the TS can be chosen independently of the location of the HLR's. In addition, the TS can be geographically distributed to improve fault-tolerance.
5) *Scalability:* Each TS only receives a fraction of the total mapping queries for all NGPN subscribers, and the hash function can be chosen so as to balance the load across the TS's. As the number of subscribers grows, the number of HLR's can be increased and the NGPN-to-HLR mapping for new subscribers can be added to the TS. At some point, the TS's may also run out of storage capacity. In that case, the number of TS's can also be increased as discussed.

One issue we have not discussed is how the NGPN mapping process is carried out internally in the TS, i.e., given an NGPN, how a TS finds the correct record which maps it to the HLR ID. This is a standard file or database searching problem, and known techniques of indexing or hashing can be applied [14].

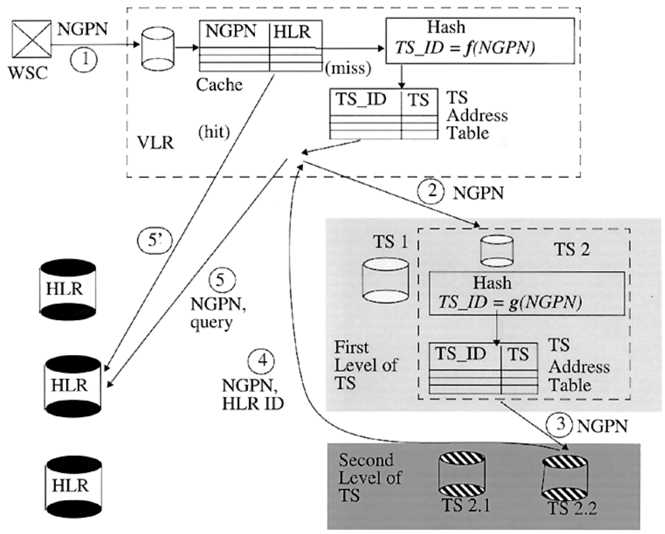


Fig. 4. Second (two-level) proposed NGPN translation scheme.

IV. Second Proposed Translation Scheme

The second translation scheme is very similar to the first but addresses the issue of scalability of TS. In the first scheme, if the number of TS's is to be increased, and a dynamic hashing method such as extendible hashing is used, two changes must take place in every VLR:

1) the number of bits, k, used from the output of the hash function must be increased;
2) the TS address table must be updated to include the addresses of the new TS's.

We anticipate that increasing the number of TS's would be a very infrequent event, and so the cost and inconvenience of modifying the VLR's may not be important. If this is viewed as a drawback, however, we propose the second scheme, in which neither of these modifications to the VLR need to take place. However, in general, a penalty must be paid: sometimes it may be necessary to perform an additional database lookup in order to complete the translation.

The second scheme differs from the first scheme as follows (see Fig. 4). When the number of TS's is to be increased, instead of modifying the VLR, a second level of TS's is created. Suppose the capacity of some TS, say TS2, has almost been exhausted and a new TS is to be added. The new TS will not be directly addressable by the VLR's, but only by TS2. At TS2, a second hash function g is applied to all the records in TS2. For each record, if g(NGPN) = 0, the record remains in TS2, otherwise it is moved to the new TS. Assuming g is uniform, roughly half the records of TS2 are moved to the new TS.

Suppose an NGPN is presented to a VLR. The VLR applies the first hash function f and obtains the TS ID; suppose this is TS2. The VLR queries TS2 for the NGPN-to-HLR mapping. TS2 performs the second hash function g on the NGPN. If g(NGPN) $= 0$, TS2 searches its own records to find the NGPN-to-HLR mapping for this user; otherwise it forward the query to the new TS. The appropriate TS returns the mapping to the VLR. (Notice that TS2 applies the hash function to all queries but only searches its records for those with g(NGPN) $= 0$; doing the reverse is not desirable since hashing is much faster than searching.)

Thus as outlined above, TS2 can be regarded as a first-level TS, and the new TS as a second-level TS. This leads to the question: What happens when the new TS becomes full? Obviously, this process could be continued recursively, with new TS's being added as required. However, for some queries a long chain of TS's might need to be queried in order to complete the NGPN translation, increasing call setup time. Instead, we propose that the extendible hashing scheme discussed in the previous section be used. Thus instead of applying the extendible hashing scheme at the VLR's, it is applied at the first level of TS's. Thus the same extendible hashing scheme as described in Section III-D is used, except that instead of using k bits of the output of hash function f, k bits of function g are used. By doing so, we obtain the following benefits.

1) At most, two levels of TS are ever created. Any translation will involve at most two TS's, and at most one TS will actually be searched for the mapping.
2) Extendible hashing is only used at those TS's which actually run out of capacity; the other TS's can remain unchanged, and translations involving them will not involve any further lookups.
3) The VLR's need not be modified as the number of TS's is increased.
4) The extendible hashing scheme becomes much simpler to apply. The value of k and the TS address table only needs to be modified at the first-level TS which is running out of capacity.
5) Different first-level TS's can use different hash functions g, if desired. (However, note that all the VLR's must still use the same first-level hash function, f.)
6) Caching can be applied not only at the VLR's as described previously, but also at the first-level TS's in order to avoid querying the second-level TS's.

V. Evaluation of Scalability

In this section, we present an analysis of the scalability of each of the four schemes presented in Sections II and III. We first estimate the NGPN translation workload as the number of NGPN subscribers increases. This analysis is quite simplified, but does indicate the potential scalability problems which can arise.

A. Total Translation Load

We develop a simple analytical model of the total rate of NGPN translations in a geographical region assuming that translations are required for call origination as well as delivery and for registrations. We then evaluate the model for an example scenario.

1) NGPN Translation Model: We first estimate the rate of translations due to call origination and delivery. Let:

n total number of users;
p fraction of users which are NGPN;
c calls/s rate of calls originated by users.

We also assume that caching at the VLR's will reduce the workload at the server, and let the cache hit ratio, i.e.,

h fraction of NGPN translation requests for which correct information is cached.

Then the rate of NGPN translations due to call origination and delivery is

$$C = 2pnc(1-h). \tag{1}$$

In order to estimate the translations due to registrations, we need a model of user mobility. As a simple first cut, we assume the uniform fluid flow model, for which the rate of registrations per registration area is given by [15]

$$a = \frac{\rho \nu L}{\pi}$$

where ρ is the user density (persons/m^2), ν is the speed at which users are moving (in a direction uniformly distributed over [0, 2π]), and L is the length of the perimeter of a registration area (in meters). For our case, we let:

s size of each registration area; each is a square;

r number of registration areas;

m fraction of NGPN users who are mobile at any given time.

Then the rate of NGPN translations due to registrations is given by

$$\begin{aligned} R &= ar \\ &= \frac{4mpn\nu}{\pi s}. \end{aligned} \tag{2}$$

From (1) and (2) for C and R, the total rate of NGPN translations is

$$\begin{aligned} T &= R + C \\ &= 2pn\left[c(1-h) + \frac{2m\nu}{\pi s}\right]. \end{aligned} \tag{3}$$

2) Scenario: We evaluate the rate of NGPN translations using the equations above for an example scenario. Consider a (real) geographical region with the following statistics. The total population of this area is 19.55 million, of which 77%, i.e., 15 million people, are over 18 years of age.

Thus we set the following.

1) $n = 15$ million.
2) We assume that users are moving at pedestrian speeds, i.e., we set $\nu = 5$ km/hr.
3) During the busy hour, the rate of call originations and deliveries is 1.4 calls/h, i.e., $c = 1.4/3600$ calls/s.
4) We assume that the percentage of NGPN users which are mobile is 10%, i.e., $m = 0.1$.
5) We assume that each registration area is a square of side $s = 5$ km.

In Fig. 5, we plot the total rate of NGPN translations, T, as a function of the fraction of users who have NGPN's, i.e., NGPN penetration in this region. The plot shows curves for various values of cache hit ratio h, i.e., the fraction of operations for which caching is effective.

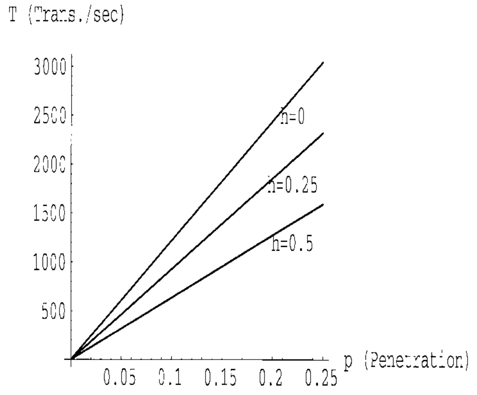


Fig. 5. Total rate of NGPN translations for the example scenario, for different cache hit ratios, h, and NGPN market penetration, p.

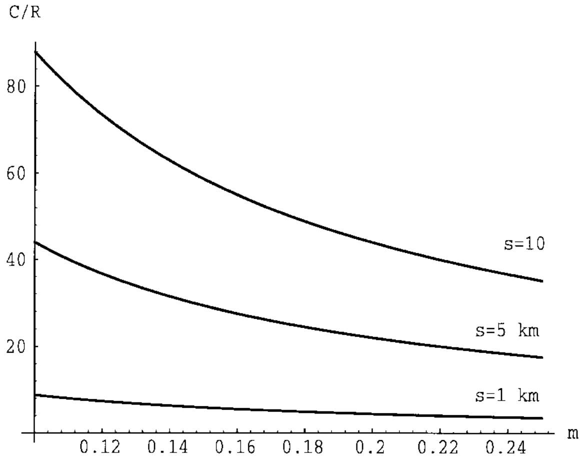


Fig. 6. Ratio of NGPN translations for calls to translations for registrations, for various fractions of mobile users, m, and registration area sizes, s.

We see that for an NGPN penetration of 25%, about 3000 translations/s. are required if no caching is used ($h = 0$). It is difficult to predict the effectiveness of caching [16], since it is highly dependent not only upon user calling and mobility patterns, but the size of the cache used at each VLR [17]. We see from Fig. 5 that under the assumption that caching is very effective (50% of translations can be performed at caches in the VLR), about 1500 translations/s are required at 25% NGPN penetration.

The rate of NGPN translations does depend upon the user mobility and, hence, is influenced by the mobility model chosen, the fraction of users assumed mobile, as well as the size of registration areas assumed. Admittedly, the uniform fluid flow mobility model we have chosen is very simplistic. However, for our assumptions, it is a conservative model, since registrations are not a significant fraction of the total translation load. To see this, in Fig. 6 we have plotted the ratio C/R, i.e., translations due to calls compared to registrations, as a function of the fraction of users which are mobile, m, for the case where caching is not used ($h = 0$). Curves for small ($s = 1$ km) as well as large ($s = 10$ km) registration area sizes are shown. It is clear that the relative contribution of registrations to the NGPN translation workload is small. Thus, for our assumptions, the contribution to translation load due to registrations is very conservative.

B. Comparison of Alternative Schemes

We now consider the various alternative schemes, applying the results from this model where appropriate.

1) *Store Mapping in VLR's:* Although the NGPN translation operation itself is very fast in this scheme, the service operations are not. In particular, adding or deleting a subscriber involves updating all the VLR's, as does moving a subscriber from one service provider to another, so that these operations will incur a relatively significant performance penalty as they have to be performed more often. Thus this scheme is scalable as the translation workload increases but not scalable as the service operation workload increases. We do not consider this option further.
2) *A Centralized Translation Server:* If a single centralized TS can be used, NGPN translation, as well as all the service operations (adding, deleting, and moving subscribers, and adding HLR's) are quite fast. However, it is unlikely that this will remain so as the number of NGPN subscribers grows. As we see from Fig. 5, even when caching is effective, at 25% penetration about 1500 translations/s are required. This is at least twice the capacity of high-availability database systems currently available. Thus even for the restricted geographical area considered in this scenario, a single TS database will be insufficient; clearly, a single TS for the entire country is infeasible from the point of view of scalability of translation operations.
3) *NGPN Translation Using Hashing:* The NGPN translation operation in this scheme, while not as fast as that for a centralized server, only involves one extra database lookup. This also holds for the other common service operations (adding and deleting subscribers, etc.). In addition, the difficult or time-consuming operations in this scheme are those which will be carried out very infrequently (e.g., splitting an HLR or a TS).

In terms of database translation loads, the number of TS's can be chosen so that it will be able to handle the rate of queries expected at some future level of penetration (say, $p = 25\%$) for a given region. This can be estimated simply by dividing the total query rate by the rate which a single database can handle. Note that this assumes that the hash function chosen will do a uniform mapping from NGPN to TS ID; if that is not the case, more TS's may be needed.

VI. Discussion and Conclusion

Our scheme relies upon combining and applying well-known concepts of hashing, extendible hashing, indirection, and caching. However, in some cases our application of these techniques differs from traditional applications.

In particular, our scheme differs from traditional hashing applications in several respects. We summarize these briefly in the following sentences. (The user is referred to [14] for more details about hashing techniques and terminology.) Individual collisions (i.e., two NGPN's mapping to the same TS) are not a problem; instead, it is desirable to design a fast hash function which splits the load evenly amongst the TS's. The size of a hash bucket (i.e., a TS) needs not be fixed *a priori*. A static hashing scheme cannot be used as the number of subscribers and service providers is expected to grow. In addition, the hash function is applied not at a single entity (e.g., the operating system of a computer) but at widely distributed entities, so that schemes which modify the hash function frequently are not desirable. To minimize call setup time, the NGPN translation should be fast, and also should place no constraints on how NGPN's are assigned to subscribers. Our scheme takes all these constraints into account.

The NGPN translation scheme we have used, although motivated by future PCS number portability with wired ATM backbones, can potentially be applied to current applications which require number or address translation also. One possible application of our hashing scheme is in the translation required for traffic between an enterprise ATM network (using some proprietary addressing scheme) and a public cell relay network (using an addressing scheme based upon E.164, say). Another potential application is in local phone number portability [18], [19], i.e., allowing a user to keep the same phone number even if he or she changes local telephone service providers or geographical regions. A third potential application is in 800 number portability, where a user can keep the same 800 number even if he or she changes service providers. Finally, we are currently developing schemes for portable Internet domain names, electronic mail addresses and IP addresses.

In summary, we have designed an architecture and method for translation of a subscriber's NGPN to the identity of a HLR database serving that subscriber. The architecture introduces a set of TS databases which store the mapping from the NGPN to the appropriate HLR. We present two schemes for performing the translation. In both schemes, when an NGPN is presented to a VLR (or serving SCP), a hash function is applied to the NGPN to route the query to the TS containing the mapping; a cache may be maintained at the VLR (or serving SCP) to avoid this query when the same NGPN is presented repeatedly, hence speeding up the translation process. A technique called extendible hashing can be used to increase the number of TS's as they run out of capacity. The extendible hashing technique entails modifying all the VLR's (and serving SCP's) when the number of TS's is increased. The second scheme allows a TS which has reached its capacity to apply another hash function to the NGPN, and route the query to one of a second layer of TS databases; the extendible hashing technique is applied at the first layer of TS's (rather than at the VLR's or serving SCP's) to ensure that at most two TS's must be queried in order to translate any NGPN.

Both schemes we present are fast in terms of database accesses required, efficient in terms of storage and signaling traffic, and flexible in allowing subscribers to change service providers (and hence, HLR's) as well as to choose their own personal numbers.

We have presented a simple analysis of the scalability of alternative architectures for NGPN translation and compared them with our hash-based schemes. In another paper, we have developed a simple model to estimate the installed first costs which would be incurred for a phased deployment of our NGPN translation schemes in terms of the present worth of expenditures. We are currently developing a simulation model to quantify the signaling traffic loads, database loads, delays and other performance parameters of our schemes as well as some of the alternatives.

Acknowledgment

The authors thank several colleagues at Bellcore for their useful comments and information, including M. Cochinwala, B. Largman, and L. Pierce. They also thank R. Hsing for his support and encouragement of this work.

References

[1] R. Jain, Y.-B. Lin, and S. Mohan, “Location strategies for personal communications services,” in *Mobile Communications Handbook,* J. Gibson, Ed. CRC Press, 1995.

[2] C. Lo and R. Wolff, “Estimated network database transaction volume to support wireless personal data communications applications,” in *Intl. Conf. Commun.,* May 1993.

[3] Bellcore, “Alternatives for signaling link evolution,” Special Report SR-NWT-002897, Feb. 1994, Issue 1.

[4] J. Luetchford, N. Yoshikai, and T.-H. Wu, “Network common channel signalong evolution,” in *Int. Switching Symp. (ISS ’95),* Apr. 1995.

[5] E. Lipper and M. Rumsiewicz, “Teletraffic considerations for widespread deployment of PCS,” *IEEE Network,* pp. 40–49, Sept. 1994.

[6] A. S. Acampora and M. Naghshineh, “An architecture and methodology for mobile-executed handoff in cellular ATM networks,” *IEEE J. Select. Areas Commun.,* pp. 1356–1375, Oct. 1994.

[7] L. V. Hauwermelren, L. Vercauteren, A. Saidi, and T. Van Landegem, “Requirements for mobility support in ATM,” in *IEEE GLOBECOM,* Dec. 1994.

[8] D. Raychaudhari, “ATM based transport architecture for multiservices wireless personal communications networks,” in *IEEE GLOBECOM,* Dec. 1994.

[9] T.-H. Wu and L. F. Chang, “Architectures for mobility management on ATM transport networks,” in *Int. Conf. Univ. Pers. Commun.,* 1995.

[10] B. A. Akyol and D. Cox, “Handling mobility in a wireless ATM network,” in *INFOCOM‘96,* San Francisco, CA, Mar. 1996.

[11] Bellcore, “CCS network interface specification (CSNIS) supporting SCCP and TCAP,” GR-1432-CORE, 1993.

[12] D. Knuth, “Sorting and searching,” in *The Art of Computer Programming.* Reading, MA: Addison-Wesley, 1973.

[13] R. Fagin, J. Nievergelt, N. Pippenger, and H. R. Strong, “Extendible hashing—A fast access method for dynamic files,” *ACM Trans. Database Sys.,* vol. 4, no. 3, pp. 315–344, Sept. 1979.

[14] J. Ullman, *Principles of Database Systems.* Rockville, MD: Computer Science Press, 1982.

[15] R. Thomas, H. Gilbert, and G. Mazziotto, “Influence of the mobile station on the performance of a radio mobile cellular network,” in *Proc. 3rd Nordic Sem.,* Copenhagen, Denmark, Sept. 1988.

[16] R. Jain, Y.-B. Lin, C. Lo, and S. Mohan, “A caching strategy to reduce network impacts of PCS,” *IEEE J. Select. Areas Commun.,* Oct. 1994.

[17] H. Harjono, R. Jain, and S. Mohan, “Analysis and simulation of a cache-based auxiliary location strategy for PCS,” in *IEEE Conf. Networks for Pers. Commun. (NPC ’94),* Long Branch, NJ, 1994.

[18] M. Rockwell, “Phone users: Price is most important,” *Commun. Week,* p. 28, Aug. 28, 1995.

[19] A. Lindstrom, “Making the switch to local number portability,” *Telephony,* pp. 48–49, July 10, 1995.

Ravi Jain (S’71–M’92) received the Ph.D. degree in computer science from the University of Texas at Austin in 1992.

Prior to receiving the Ph.D. degree, he worked at Syntrex Inc., SES Inc., NJ, and the Schlumberger Laboratory for Computer Science, TX, on developing communications and systems software, performance modeling, and parallel programming. In 1992, he joined Bellcore as a Research Scientist in the Personal Communications Services area. His research interests include design and analysis of algorithms, architectures and protocols for mobile computing and communications. He has designed and analyzed protocols for locating mobile users, mobile database access, and mobile sales applications. He was also technical team leader for the SCOUT system, an intelligent transportation system application for delivering personalized road traffic information to mobile users; this project is being incorporated into Bellcore’s AirBoss product line. Currently, he is studying issues arising from supporting mobile users with fixed ATM backbone networks and mobile IP. He is Guest Co-Editor of the Special Issue on “Networking and Performance Issues of Personal Mobile Communications” of the IEEE Journal on Selected Areas of Communications, on “Databases and Mobile Computing” of the journal *Distributed and Parallel Databases,* and on “Simulation of PCS Systems” of the *International Journal of Computer Simulation*. He is an Area Editor of the ACM/Baltzer journal *NOMAD*, has one patent issued and seven pending in the area of mobile computing and communications, and several publications.

Dr. Jain is a member of ACM, the Upsilon Pi Epsilon, and Phi Kappa Phi honorary societies.

Subhashini Rajagopalan received the M.S. degree in electrical and computer engineering from Rutgers University, New Brunswick, NJ, in 1995.

She has been with Bellcore, Red Bank, NJ, since March 1995. She is currently in the Wireless Network Broadband Research Group. Her research interests include mobile computing, internetworking and interoperability, and broadband wireless networking.

Ms. Rajagopalan is a member of ACM.

Li Fung Chang (S’84–M’85–SM’90), for a photograph and biography, see p. 3, this issue.