A HASHING SCHEME FOR PHONE NUMBER PORTABILITY IN PCS SYSTEMS WITH ATM BACKBONES

Ravi Jain, Subhashini Rajagopalan and Li-Fung Chang
Applied Research
Bell Communications Research

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. A special feature of next generation wireless access service will be to support PCS and wireless subscribers with portable personal numbers, or non-geographic phone numbers (NGPN), that do not indicate the service provider or HLR database serving the user. In addition the Global Title Translation (GTT) function that converts the subscriber's number to an HLR database address may not be available when the wired backbone is an ATM network. Thus a key function required will be to translate an NGPN to the ID 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 a scheme for fast, efficient, scalable and flexible NGPN translation which applies ideas of dynamic hashing, caching, and indirection. The scheme uses a hash function in the Visiting Location Registers (VLR) (or serving SCP) and a set of distributed Translation Servers which store the NGPN-to-HLR mapping.

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) subscriber, when the only relevant information available is the subscriber's non-geographic 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. This is accomplished by a process called Global Title Translation (GTT), executed at signaling switches called Signaling Transfer Points (STPs) which translate information obtained from a subscriber's phone number to the identity of the Home Location Register (HLR) database which serves that subscriber (for a tutorial on location management in wireless networks, see [1]). For next generation wireless access service, however, subscribers will be assigned non-geographic phone numbers or 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, STPs may not be used for signaling and hence 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. VLRs) 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.

The NGPN translation scheme we have used, although motivated by future PCS number portability with wired ATM backbones, can be applied to current applications which require number or address translation. 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.

In the rest of this section we present the background and motivation for this work and the requirements of the NGPN translation process. In section II we discuss some possible alternative architectures for NGPN translation, including a completely distributed approach (store the NGPN-to-HLR mapping in the VLRs), and a completely centralized approach. In section III we present a scheme for NGPN translation, which combines ideas of distributed dynamic hashing, caching, and indirection. In section IV we end with a summary.

A. Background and Motivation

A.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

Address correspondence to:
Ravi Jain, Bellcore, 331, Newman Springs Rd., Red Bank, NJ - 07701.
Email: rjain@thumper.bellcore.com
Phone: (908) 758 2844, FAX: (908) 758 4371

0-7803-3692-5/96 © 1996 IEEE

called handset or subscriber unit), as well as the subscriber's profile and authentication information, using a two-level hierarchy of databases called Home Location Registers (HLR) and Visiting Location Registers (VLR). 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].

The signaling and control traffic carried over the existing Signaling System No. 7 (SS7) network 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 signaling network may not be adequate to support the stringent delay requirements imposed by broadband and PCS services [2, 3, 4, 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, 7, 8, 9, 10].

We consider the situation where PCS signaling traffic is integrated with user traffic on the same physical ATM backbone network (although in separate logical channels). The network reference architecture we assume is shown in Fig. 1. The architecture contains a network entity called a Wireless Service Center (WSC) which replaces the current Mobile Switching Center (MSC). A WSC 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 shows the VLR connected to the WSC, but it may instead be connected to an ATM switch in the ATM backbone.

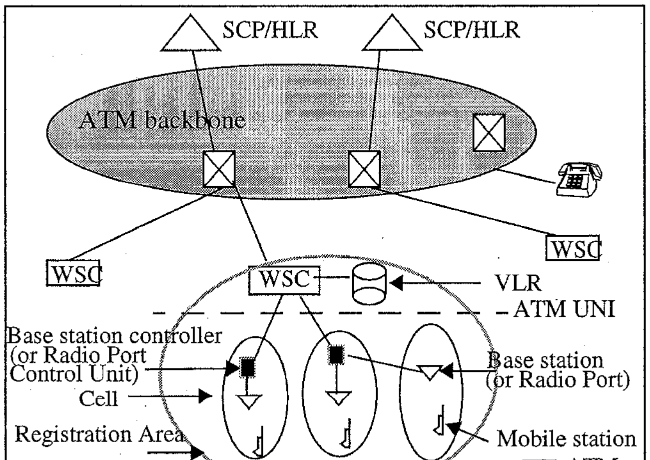

Figure 1. ATM-backbone network reference architecture for PCS

The ATM User-Network Interface (UNI) is between the WSC and the Base Station Controller (BSC). 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, 10].

For PCS systems we anticipate 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 HLRs.

A.2. Signaling message routing and number translation

In the current design for PCS service, a process called Global Title Translation (GTT) is carried out at the STPs to handle the routing of PCS signaling messages associated with registration and call delivery.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 WSC. If the VLR does not have the identity of the HLR serving this subscriber it forwards 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 more than one HLR that can be queried. In addition, we expect that the subscribers's MIN may be "non-geographic".. 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 HLRs contain identical information, then GTT at the STP's can be used. However, 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. Secondly, it is likely that with the integration of signaling and user data transport on an ATM-backbone network, STPs and the GTT function may no longer be in use.

A.3. Requirements for NGPN translation

From the discussion above, it follows that in order to support future PCS users with NGPN, it is necessary to develop a new method for translating the NGPN to the ID of the appropriate HLR database when a user registers, receives a call, and (possibly) originates a call. In this paper, we will simplify the discussion by assuming that for all three cases, 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, two possibilities arise. If the caller is also a PCS subscriber, the VLR in the caller's RA may be responsible for performing the translation[1]. If the caller is calling from a fixed telephone, the caller's request will be forwarded to other network entities, e.g., an AIN SSP, which will in turn forward the request to a serving SCP.

In this paper, therefore, when we discuss the operations required to perform (or obtain) a translation as being performed at a VLR, it is understood that for call delivery to PCS subscribers, the operations may in fact have to be performed at a serving SCP.

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 and 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 NGPNs to be chosen by an independent entity (e.g. a Numbering Plan Administrator) without considering how they are to be mapped to HLR ID.

It is expected that the NGPN-to-HLR mapping may have to be changed from time to time a subscriber may change his or her service provider, and hence needs to be mapped to a different HLR, or a subscriber may move (semi-)permanently from one geographical region to another, so that it is more efficient to serve the subscriber from another HLR. Mappings will also need to be changed when an HLR runs out of storage or processing capacity, so that some load has to be diverted to another HLR. Therefore it is important that changing the NGPN-to-HLR mapping be an operation which can be performed quickly and efficiently and without too much software complexity or overhead.

II. Some Alternative Schemes

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

A. Store mapping in VLRs

A simple method of performing NGPN translation is as follows. Every VLR is loaded with a table which specifies a mapping from an NGPN to the ID of the HLR serving that NGPN. An Operations Support System (OSS) can be set up to allow the VLRs to be loaded and maintained. Now when a switch is presented with an NGPN it simply forwards 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. Firstly, every VLR must store the mapping for all the NGPNs, 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 non-trivial when we consider all the VLRs in the country. More importantly, when the NGPN-to-HLR mapping for a subscriber needs to be modified, then all the VLRs 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 VLRs 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 TSs do not perform the other functions performed by a VLR, it is expected that there will be far fewer TSs than VLRs, and in addition, they will not be so widely dispersed. Thus if the NGPN-to-HLR mapping has to be changed, only the TSs have to be updated, rather than every VLR, and this will be a simpler operation.

One approach is that the NGPN-to-HLR mapping is only stored in a single, central location, 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 forwards 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 VLRs, 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. As the number of PCS subscribers grows, a single TS will not be able to handle the queries for call delivery.

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 VLRs. 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 which are fully replicated, i.e., each TS contains the mappings for all NGPNs. However, such an architecture entails an operations support system which must be capable of modifying all the databases concurrently and efficiently.

III. Proposed NGPN Translation Scheme

The logical architecture of our proposed NGPN translation scheme is such that the VLRs in the system are connected via a signaling network to the Translation Servers (TSs) as well as the HLRs. Each TS contains NGPN-to-HLR mappings, and an NGPN can be translated to any of the HLRs in the system.

. Alternatively, the caller's request may be presented to some other network entity (e.g. an AIN SSP or a serving SCP) for translation

A. The basic scheme

The scheme can be briefly described as follows; see Fig. 2. We first describe the scheme without the use of the cache in the VLR. (Recall that, as mentioned previously, we will simplify the presentation in this paper by assuming that for all cases where NGPN translation is required, the NGPN is presented to a VLR; it is understood that for call delivery to PCS subscribers, the operations may in fact have to be performed at a serving SCP.)

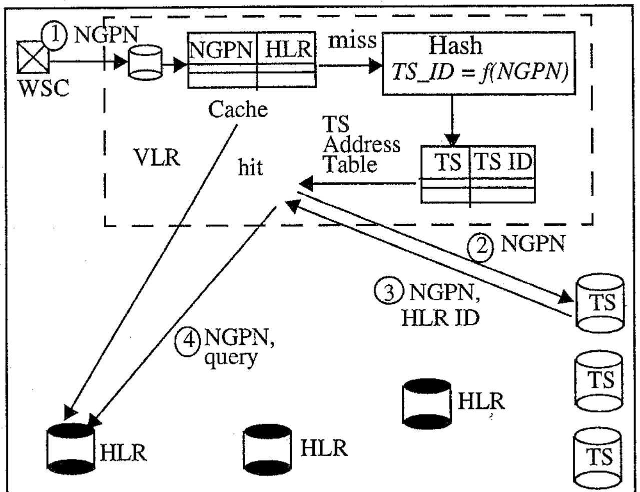

Figure 2. Proposed NGPN translation scheme

When any of the situations requiring NGPN translation occur, the NGPN is presented to a switch. The switch forwards the NGPN to the VLR serving that switch. 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. 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. 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. The TS responds to the VLR by returning the HLR ID. 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 VLRs with hash functions, manage the NGPN-to-HLR mapping in the TSs, 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. Similarly, the TS can also maintain a cache of NGPN translations in high-speed memory to speed up its own operation.

C. The hash function

The two key requirements of the hash function are that i should be easy to compute, and that it should result in mappin roughly the same number of NGPNs to each TS.

An example of a simple hash function is the functior *even()*, which returns 0 if the argument is even and 1 otherwise Obviously, this function can only be used if there are only tw TSs. In addition, if the way in which NGPNs are chosen is non uniform (e.g., a disproportionate number of subscribers reques phone numbers ending in 0), the load on the two TSs will not b balanced. Similarly, if the service provider assigns NGPNs t new subscribers using some administrative process (e.g. b reusing the NGPNs of subscribers who have canceled service which somehow introduces some non-uniformity, the load o the TSs will not be balanced.

Numerous hash functions have been designed an evaluated for applications such as searching for a given recor in a file. Some which have been found to be useful in practic include the following [12]:

- The *mod.* The function *f* takes the NGPN, and the numbe of TSs, *t,* as arguments, and returns the value *NGPN mo t.* When a *mod* function is used, the divisor (i.e., *t*) i typically chosen to be a prime number; this would mea that the number of TSs can only be increased from on prime number to the next. However, depending upon th assignment of NGPNs to subscribers, choosing *t* to b prime may not be required for our application.
- The "middle square". The function *f* takes the NGPN, positive integer *r*, and the number of TSs, *t*, as argument where *t* lies between 1 and 2^r. Let *p* be the numbe represented by the middle *r* bits of the binar representation of the square of the NGPN; the function returns *p mod t*. (Note that in this case *t* need not be prime.
- The "folded sum". The function *f* takes the NGPN, and th number of TSs, *t,* as arguments. It partitions the bi representation of the NGPN into several parts, and adds a the parts to obtain a value *p*, and returns *p mod t*. (Onc again, *t* need not be prime.)

The choice of the hash function has to be guided by th number of TSs and the way in which NGPNs are selected an assigned to subscribers. This will require information an experimentation for each specific deployment scenario.

D. Discussion

The NGPN translation operation in the hashing scheme while not as fast as that for a centralized server, only involve one extra database lookup. In terms of database loads, th number of TSs can be chosen to be those that will be able t handle the rate of queries expected at some future level o penetration (say, 25%) for a given region. This can be estimate simply by dividing the total query rate by the rate which a singl database can handle. Note that this assumes that the has function chosen will do a uniform mapping from NGPN to TS ID; if that is not the case, more TSs may be needed.

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.
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, eliminating the concurrent update problem.
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 VLRs, but this is the same for all NGPNs, 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 HLRs. In particular, the TS can be geographically distributed to improve fault-tolerance and to reduce the communication costs of querying the TSs.
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 TSs. As the number of subscribers grows, the number of HLRs can be increased and the PN-to-HLR mapping for new subscribers can be added to the TS. At some point, the TSs may also run out of storage capacity. In that case, the number of TSs can also be increased using extendible hashing as discussed in [13]. We have presented an evaluation of scalability of the schemes for address translation proposed in section II and III, in [13].

Notice that in the proposed scheme the hash function must be stored in all the VLRs, and must be the same for all the VLRs. If the hash function has to be changed or the number of TS has to be increased, all the VLRs must be updated. We anticipate that this would be a very infrequent event, and this effect would be outweighed by the advantages above. However, if this is viewed as a serious problem for some application, we have proposed a second translation scheme [13] which avoids it.

IV. Conclusions

We have designed an architecture and method for translation of a subscriber's NGPN to the identity of a HLR database serving that subscriber. We introduce a set of databases, called Translation Servers (TS), which store the mapping from the NGPN to the appropriate HLR. We present a scheme for performing the translation. The scheme we present is 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, HLRs) as well as to choose their own personal numbers.

Acknowledgments

We thank several colleagues at Bellcore for their useful comments and information, including Munir Cochinwala, Bob Largman and Linda Pierce. We also thank Russell Hsing for his support and encouragement of this work.

References

[1] Ravi Jain, Yi-Bing Lin and S. Mohan, "Location strategies for personal communications services", in *Mobile Communications Handbook*, ed. J. Gibson, CRC Press, 1995.

[2] Charles Lo and Richard Wolff, "Estimated network database transaction volume to support wireless personal data communications applications", Intl. Conf. Comm., May 1993.

[3] Bellcore, "Alternatives for Signaling Link Evolution", Special Report SR-NWT-002897, Issue 1, February 1994.

[4] J. Luetchford, N. Yoshikai and T. -H. Wu, "Network Common Channel Signaling Evolution", Intl. Switching Symp. (ISS '95), April 1995.

[5] E. Lipper and M. Rumsiewicz, "Teletraffic considerations for widespread deployment of PCS", IEEE Network, pp. 40-49, Sep. 1994.

[6] A. S. Acampora and M. Naghshineh, "An architecture and methodology for mobile-executed handoff in cellular ATM networks", IEEE J. Selected Areas Comm., pp. 1356-1375, Oct. 1994.

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

[8] D. Raychaudhari, "ATM based transport architecture for multiservices wireless personal communications networks", IEEE Globecom, Dec. 1994.

[9] T. -H. Wu and L. F. Chang, "Architectures for mobility management on ATM transport networks", Intl. Conf. Univ. Pers. Comm., 1995.

[10] B. A. Akyol and D. Cox, "Handling mobility in a wireless ATM network", 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, *The Art of Computer Programming. Vol 3. Sorting and Searching*, Addison-Wesley, 1973.

[13] R. Jain, S. Rajagopalan and L. F. Chang, "Phone Number Portability for PCS Systems with ATM Backbones Using Distributed Dynamic Hashing", to appear in IEEE JSAC issue on Wireless ATM, late 1996.