Phone Number Translation Delay in PCS Systems with ATM Backbones

Ravi Jain[*]
Applied Research
Bell Communications Research

February 13, 1997

Summary

Future wired networks with an ATM backbone will need to support PCS and wireless subscriber services. One of the key functions required to support PCS and wireless subscribers with non-geographic phone numbers (NGPN) will be the ability to efficiently identify which Home Location Register (HLR) database serves the subscriber. (Note that the same functionality is also needed to serve subscribers with portable phone numbers.) In a previous paper we have presented a scheme for NGPN translation based upon distributed, dynamic hashing. The scheme uses a hash function in the Visitor Location Registers (VLR) and a set of distributed Translation Servers (TS) which store the NGPN-to-HLR mapping.

In this paper we present a preliminary investigation of the additional call setup delay introduced by the NGPN translation scheme we have proposed. We develop a queuing network model for both the one-stage and two-stage versions of our NGPN translation scheme, and we investigate the impact of two key aspects of our scheme, namely the use of hashing and of caching. A poor choice of hash function can lead to imbalanced loading at the TS. We quantify the impact of this imbalance for an example scenario and show that caching at the VLR can substantially reduce the mean translation delay. We also consider the use of the two-stage scheme and a second-level cache when the translation load increases.

[*] Address correspondence to: Ravi Jain, Applied Research, Bellcore, 331 Newman Springs Rd, Red Bank, NJ 07701. Phone: (908)-758-2844. Fax: (908)-758-4371. Email: rjain@bellcore.com

1 Introduction

One of the key functions required to support Personal Communications Services (PCS) and wireless subscribers in PCS systems is mobility management, i.e., determining the location of a mobile user so that calls can be delivered to and from that user. In current and proposed standards for PCS and cellular systems a set of databases, called Home Location Registers (HLR) and Visitor Location Registers (VLR), is used to maintain the information about the current location of a user, and this information is consulted or updated whenever a user moves across geographical regions called Registration Areas (RA), or when a call is to be delivered to and from a user. For a tutorial on mobility management procedures, see [7, 10].

This paper considers an issue that arises when two important trends in the development of future PCS systems converge: (1) the use of high-speed ATM networks for the fixed network backbone which supports mobile users, and (2) the rise in the number of PCS users with non-geographic phone numbers (NGPN), i.e., phone numbers which do not indicate the home geographical region or the service provider of the mobile user. In this scenario, as part of the mobility management procedures, it is important that the system be able to efficiently translate the NGPN of a PCS subscriber to the identity of the HLR which maintains the current location of the subscriber. In previous work [5, 6] we have presented a scheme for NGPN translation which uses a set of distributed Translation Servers (TS). In this paper we develop a model for estimating the mean delay introduced by the NGPN translation scheme we have proposed.

We begin by describing, in this section, the NGPN translation problem and summarizing the solution we have proposed. In sec. 2 we describe our performance model for the one-stage and two-stage versions of our NGPN translation scheme, and in sec. 3 we demonstrate its application for an example scenario. Finally, we end with some conclusions.

1.1 Background on NGPN Translation

The problem of NGPN translation is to determine the identity of the signaling network database which serves a Personal Communications Services (PCS) subscriber, when the only relevant information available is a 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. For proposed PCS systems, subscribers will be assigned NGPNs (e.g., 1-500-XXX-XXXX), which do not contain this information; a process called Global Title Translation (GTT) has been designed for this purpose [2, 4]. GTT is executed at signaling switches called Signaling Transfer Points (STPs), and essentially translates a subscriber's NGPN to the identity of the Home Location Register (HLR) database which serves that subscriber.

For future PCS systems in which the wired backbone is an ATM network, however, signaling traffic will be likely to use the same physical transport as the user data traffic [3]. Thus STPs may not be used for signaling and GTT cannot be performed [11].

NGPN translation is required in three situations:

1. When a PCS subscriber with a NGPN is called (by a fixed or PCS subscriber), its HLR ID is required to set up the call.
2. In some implementations, NGPN translation may be required when a PCS subscriber crosses Registration Areas (RAs) served by different VLRs, since the ID of its serving HLR needs to be determined in order to update the subscriber's location information.
3. In some implementations, NGPN translation may also be required when a PCS subscriber originates a call, in order to obtain authentication and service profile information.

We will simplify the discussion by assuming that for all three cases 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, although it is understood that for call delivery from wireline phones to PCS subscribers, the operations may in fact have to be performed at a serving Service Control Point (SCP) [1].

Clearly, the process of NGPN should be *fast* (in order to reduce overall call setup time), *efficient* (in terms of signaling network and database loads), and *scalable* (as the translation load increases.) Another key requirement is that the process allow the NGPN-to-HLR mapping for any user to be changed easily and efficiently, since the mapping may need to be changed any time the user changes service providers, or if the service provider changes the HLR serving that user (for performance or administrative reasons, e.g. if the user moves permanently from one region of the country to another.) This requirement for *efficient mapping modification*, along with the requirement for scalability, are the main reasons that naive schemes for performing the NGPN translation do not suffice, necessitating the scheme that we have developed.

1.2 The proposed NGPN translation scheme

In order to keep this document self-contained, the scheme presented in [5, 6] is briefly described here.

1.2.1 One-stage NGPN Translation Scheme

We first describe the one-stage version of the scheme [5]; see Fig. 1.

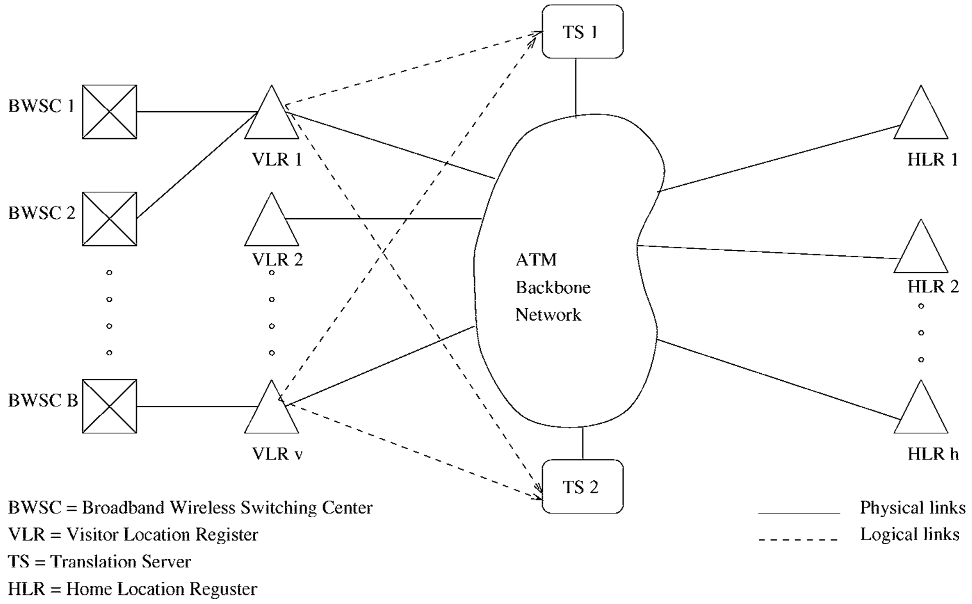

Figure 1: Architecture for the (One-Stage) NGPN Translation Scheme

1. When any of the situations requiring NGPN translation occur, the non-geographical PN is presented to a switch; depending upon the architecture deployed, this may be a Mobile Switching Center (MSC) or Broadband Wireless Switching Center (BWSC), or a Service Switching Point (SSP).

2. 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 an entity called a Translation Server (TS). Translation servers are entities not present in current and proposed PCS architectures; they were introduced for the purpose of NGPN translation. Note that TSs are logical entities; physically they may be implemented as databases and be co-located with HLRs.

3. The VLR launches a query to the TS specified by $f(NGPN)$, passing it the value NGPN. The TS contains a table mapping the NGPN to the ID of the HLR serving that NGPN.

4. The TS responds to the VLR by returning the HLR ID.

5. The VLR uses the HLR ID to continue with the registration, call delivery, or call origination signaling operations as usual.

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. 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.

A simple example of the hash function, f, 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 TSs. In addition, if the way in which NGPNs are chosen is non-uniform (e.g., a disproportionate number of subscribers request phone numbers ending in 0), the load on the two TSs will not be balanced; similarly, if the service provider assigns NGPNs to new subscribers using some administrative process which somehow introduces some non-uniformity, the load on the TSs will not be balanced. The effect of load imbalance at the TSs will be a factor we will consider in our model.

As we have discussed in [6], we believe the one-stage NGPN translation scheme offers speed and efficiency, and the use of indirection (storing the NGPN-to-HLR mapping in the TS) offers efficient mapping modification. In order to improve scalability, we have proposed the use of a *dynamic hashing* method which can be used if necessary to increase the number of TS as the translation load grows.

1.2.2 Two-Stage NGPN Translation Scheme

We have also presented a two-stage version of our NGPN translation scheme in order to improve scalability (see Fig. 2). The two-stage translation scheme is identical to the one-stage scheme described above as far as step 2, i.e., the VLR applies the hash function f and determines the identity of a TS. However, step 3 is modified by introducing a second stage of TS, as follows:

3(a) The VLR launches a query to the TS specified by $f(NGPN)$, passing it the value NGPN.

3(b) This *first stage* TS applies another hash function, $g(NGPN)$, and obtains the identity of a *second stage* TS.

3(c) The first-stage TS launches a query to the second-stage TS, passing it the NGPN.

3(d) The second-stage TS contains a table mapping the NGPN to the ID of the HLR serving that NGPN.

Step 4 for the two-stage scheme is identical to that for the one-stage scheme, except that it is the the second-stage TS which responds to the VLR by returning the HLR ID. Similar to the VLR cache, the first stage TS can maintain a *TS cache* for speeding up the translation process.

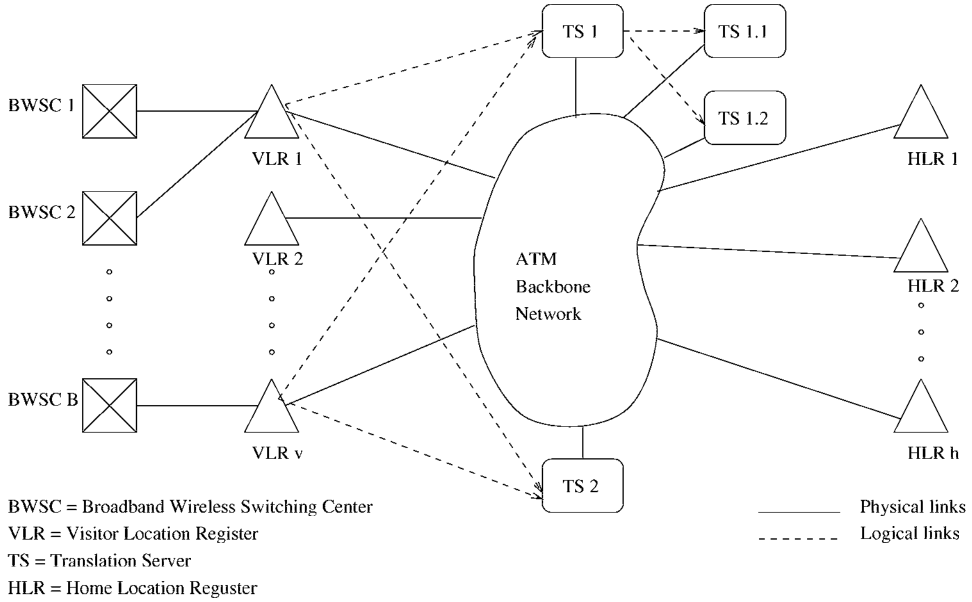

Figure 2: Architecture for the Two-stage NGPN Translation Scheme

2 Performance model

We have developed a queuing network model for estimating the delay entailed by our proposed NGPN translation scheme. In this section we briefly describe the model and its assumptions.

The model is developed for the two-stage NGPN translation scheme; by an appropriate choice of parameters it models the one-stage scheme also. A schematic diagram of the model is shown in Fig. 3, and a list of the parameters is shown in the first two columns of Table 1.

The model focuses on the path taken by a single translation request as it arrives at a given VLR, which we call VLR_1, and travels to a selected TS, TS_1, and its second- stage TS, $TS_{1.1}$. Referring to Fig. 3, translations arrive at VLR_1 at a rate λ; a fraction p_1 enjoy cache hits, i.e., are served immediately by the cache and exit the system, while the remaining $(1-p_1)\lambda$ require further processing. Recall that a hash function is applied at the VLR, so that this remaining traffic is (ideally) divided among all the T TSs in the system, i.e., with no load imbalance, the traffic from VLR_1 to each TS is

$$\lambda_{ms} = \frac{(1-p_1)\lambda}{T} \qquad (1)$$

In order to model the effect of load imbalance, the traffic from VLR_1 to a selected TS, TS_1, is given in terms of a load imbalance factor f_1, where $1 \leq f_1 \leq T$. Thus, the traffic from VLR_1 to

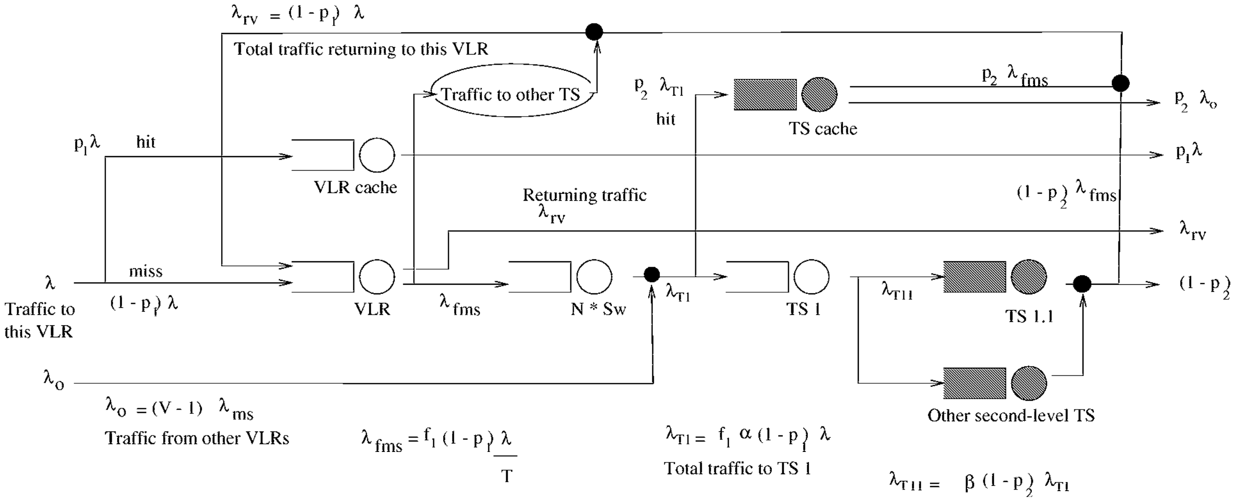

Figure 3: Model of NGPN translation delay. The shaded queues are omitted for the one-stage version of the NGPN scheme.

TS_1 is $\lambda_{fms} = f_1 \lambda_{ms}$, and the traffic from VLR_1 to each of the other TSs is

$$\lambda_{oms} = \frac{(T - f_1)\lambda_{ms}}{T - 1} \tag{2}$$

To keep Fig. 3 intelligible we do not show the model for traffic to the other TSs; it is simply indicated in the oval marked "Traffic to other TS". We assume that an ATM Switched Virtual Circuit (SVC) connection setup is required to forward the translation request, and on average, there are N switches between the VLR and the (first-stage) TS, each of which has a mean service time of $1/\mu_{sw}$. The traffic into the switches connecting VLR_1 and $TS1$ is λ_{fms}, and the traffic into the switches connecting VLR_1 to the other TSs is λ_{oms}.

When the traffic from VLR_1 reaches the first-stage TSs, it is combined with the traffic from all the other $(V - 1)$ VLRs in the system, which we denote λ_O. We assume that all the other VLRs also have an incoming load of λ each, and that the cache hit ratio is also p_1 at each. Then, assuming that the load at the other $(V - 1)$ VLRs is perfectly balanced,

$$\lambda_O = (V - 1)\lambda_{ms} \tag{3}$$

Thus the total traffic reaching the first-stage TS TS_1 from all the VLRs (including VLR_1) is $\lambda_{t1} = \lambda_{fms} + \lambda_O$, while that reaching the other the first-stage TSs is $\lambda_{ot1} = \lambda_{oms} + \lambda_O$.

The total traffic entering the first-stage TS can be satisfied directly by the cache with hit probability p_2. The remaining traffic is assumed to be equally divided amongst the second-level TSs, where $1/\beta \geq 1$ is the ratio of second-stage to first-stage TSs.

Parameter	Symbol	Value
Direct translation load to this VLR	λ	1 - 30 trans/sec.
Service time for VLR	$1/\mu_v$	5 ms
Background load to this VLR	λ_{bv}	$.4\mu_v$
Service time for VLR cache	$1/\mu_{vc}$	1.25 ms
Cache hit ratio at VLR	p_1	0, 0.1, 0.25, 0.5
Number of TSs	T	3
Number of VLRs	V	50
Load imbalance factor	f_1	1.0 - 3.0
Avg. number of ATM switches from VLR to TS	N	3
Service time for SVC setup, per switch	$1/\mu_{sw}$	20 ms
Background load to each switch	λ_{bs}	$.4\mu_{sw}$
Cache hit ratio at first-stage TS	p_2	0, 0.1, 0.25, 0.5
Service time at first-stage TS (one-stage) Service time at first-stage TS (two-stage)	$1/\mu_{t1}$ $1/\mu_{t1}$	5 ms 1.25 ms
Ratio of first-stage to second-stage TS	$1/\beta$	2
Service time at second-stage TS	$1/\mu_{t11}$	5 ms

Table 1: Parameters of the NGPN translation model

It is worth pointing out that the second stage is modeled as $1/\beta$ separate queues, rather than a single queue with $1/\beta$ servers. This is because the second-stage hash function g completely determines which second-stage TS an incoming translation request is sent to, i.e., incoming requests are not indistinguishable jobs which can be arbitrary sent to any of the $1/\beta$ queues.

After this second stage the traffic from the other VLRs, which is now $(1-p_2)\lambda_O$, returns to those VLRs. The traffic from VLR_1, which is $(1-p_2)\lambda_{fms}$ at TS_1 and $(1-p_2)\lambda_{oms}$ at the other TSs, returns to VLR_1; it is joined by the traffic which enjoyed a cache hit at the TS cache, so that the total returning traffic at VLR_1 is $\lambda_{rv} = (1-p_1)\lambda$. This returning traffic undergoes final processing at the VLR before exiting the translation process (and moving on to other processing, i.e. HLR querying.) Finally, we assume that the VLR and the switches both have background traffic of λ_{bv} and λ_{bs} respectively. For simplicity these background flows are not shown in Fig. 3.

For the purposes of this paper we consider that translation requests are Poisson arrivals and all the queues in the model are M/M/1 queues [8]. (Note once again that the second-stage TS is modeled as $1/\beta$ separate M/M/1 queues, rather than a single M/M/$(1/\beta)$ queue.) We can then calculate the total response time R, i.e, delay, of the translation process using well-known techniques [9, 8]. In the following, R_i is the residence time for the traffic from VLR_1 to TS_1 at entity i, where $i = v, vc, sw, t1, t1c, t11$ respectively represents the VLR, VLR cache, ATM switch, TS1, TS1 cache, and TS1.1. R_{other} denotes the delay experienced by the traffic from VLR_1 which is serviced by the other TSs in the system. The binary variable $Stage$ is set to 0 for a one-stage system and 1 for a two-stage system.

$$
\begin{aligned}
R &= R_v + R_{vc} + R_{sw} + R_{t1} + Stage * R_{t1c} + Stage * R_{t11} + R_{other} && (4)\\
where & \\
R_v &= \frac{2(1-p_1)}{\mu_v - (\lambda_{bv} + 2(1-p_1)\lambda)} && (5)\\
R_{vc} &= \frac{p_1}{\mu_{vc} - p_1\lambda} && (6)\\
R_{sw} &= \frac{(1-p_1)Nf_1}{T\mu_{sw} - (T\lambda_{bs} + (1-p_1)f_1\lambda)} && (7)\\
R_{t1} &= \frac{\frac{(1-p_2)\lambda_{fms}}{\lambda}}{\mu_{t1} - (1-p_2)(\lambda_{fms} + \lambda_O)} && (8)\\
R_{t1c} &= \frac{\frac{p_2\lambda_{fms}}{\lambda}}{\mu_{t1c} - p_2(\lambda_{fms} + \lambda_O)} && (9)\\
R_{t11} &= \frac{\frac{(1-p_2)\lambda_{fms}}{\lambda}}{\mu_{t11} - \beta(1-p_2)(\lambda_{fms} + \lambda_O)} && (10)
\end{aligned}
$$

The delay experienced by the traffic from VLR_1 which is serviced by the TSs other than TS_1 is given by

$$
\begin{aligned}
R_{other} &= \sum_{i=2}^{T}(R_{osw}(i) + R_{ot1}(i) + Stage * R_{ot1c}(i) + Stage * R_{ot11}(i)) \\
&= (T-1)(R_{osw} + R_{ot1} + Stage * R_{ot1c} + Stage * R_{ot11}) && (11)
\end{aligned}
$$

In Eq. 11 R_j is the delay at entity j, and $j = osw, ot1, ot1c, ot11$ represents the other switches, other first-stage TSs, other second-level cache, and other second-level TSs, and by "other" we mean the entities not associated with TS_1. We can derive expressions for the terms in R_{other} in a manner similar to those in Eq. 4.

3 Effect of caching and load imbalance

In this section we illustrate the use of the model described above for an example scenario. We focus on investigating the effect of caching and translation load imbalance. We use the values of the parameters given in Table 1 and plot the mean delay given by Eq. 4.

The translation workload is calculated assuming a low-tier PCS system, with small cells. We assume n active users per cell (base station), b base stations per base station controller, B base station controllers per Broadband Wireless Switching Center (BWSC), v BWSC per VLR, and c

call originations and terminations per user during the busy hour. The translation traffic due to registrations is ignored since it is much less than that due to call originations and terminations [5]. Then the total translation workload per VLR is $\lambda = nbBvc$. For this paper we will first consider an example PCS system with $n = 15$, $b = 96$, $B = 8$, $v = 1$ and $c = 3$ calls/hour, so that $\lambda = 9.6$ trans/sec during the busy hour. We will plot the total translation delay for the range $1 \leq \lambda \leq 15$ trans/sec. We will later also consider a scenario where PCS penetration has increased and $n = 30$, so that $\lambda = 19.2$ trans/sec during the busy hour; for this case we will plot the delay in the range $1 \leq \lambda \leq 30$ trans/sec. As a rule of thumb, it is desirable that the mean delay due to translation alone be about 0.5 second or less.

3.1 One-stage version

To model the one-stage version using Eq. 4, we set $Stage = 0$, $p_2 = 0$, $\beta = 1$ and arbitrary positive values for $1/\mu_{t11}$ and $1/\mu_{t1c}$.

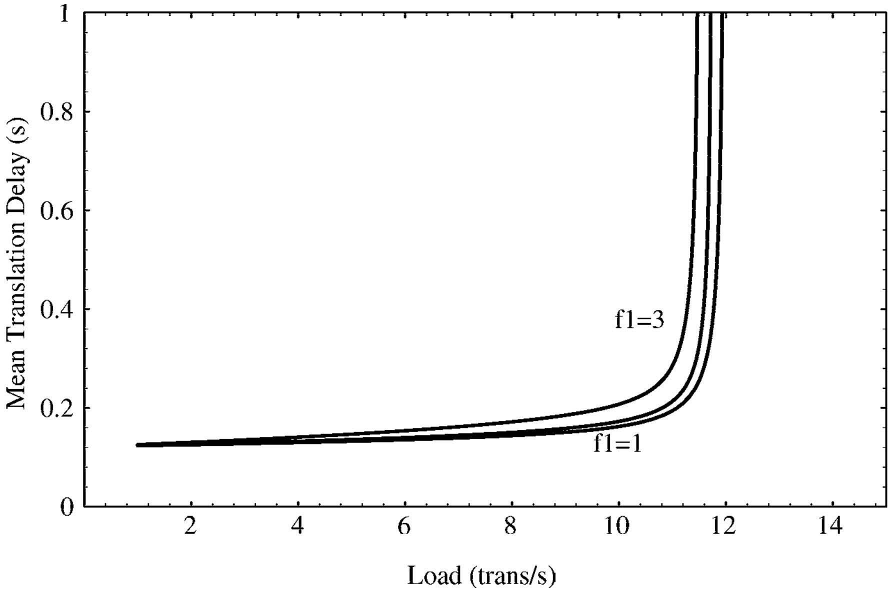

Figure 4: One-stage version: Mean translation delay for $p_1 = 0$ (no cache at VLR) and $1/\mu_{t1} = 5ms$. The load imbalance factor f_1 is varied; the three curves are for $f_1 = 3.0, 2.0, 1.0$ respectively.

In Fig. 4 we plot the mean delay as a function of the translation load, assuming no caching at the VLR ($p_1 = 0$), the service time at TS1 is $1/\mu_{t1} = 5ms$, the load imbalance factor f_1 is varied, and the remaining parameters are as in Table 1. There is a sharp increase in the total delay R around $\lambda = 11$ trans./sec.. This occurs because the first-stage TS, TS_1, becomes saturated. The load imbalance factor is not significant for $\lambda < 8$ trans/sec.; this is because the load imbalance at TS_1 results in higher load at TS_1 but lower load at the other TSs, so that the increased delay at TS_1

due to load imbalance is mitigated by the reduced delay at the other TSs. For $\lambda > 8$ trans./sec, the load imbalance factor begins to have a noticeable effect on the overall delay.

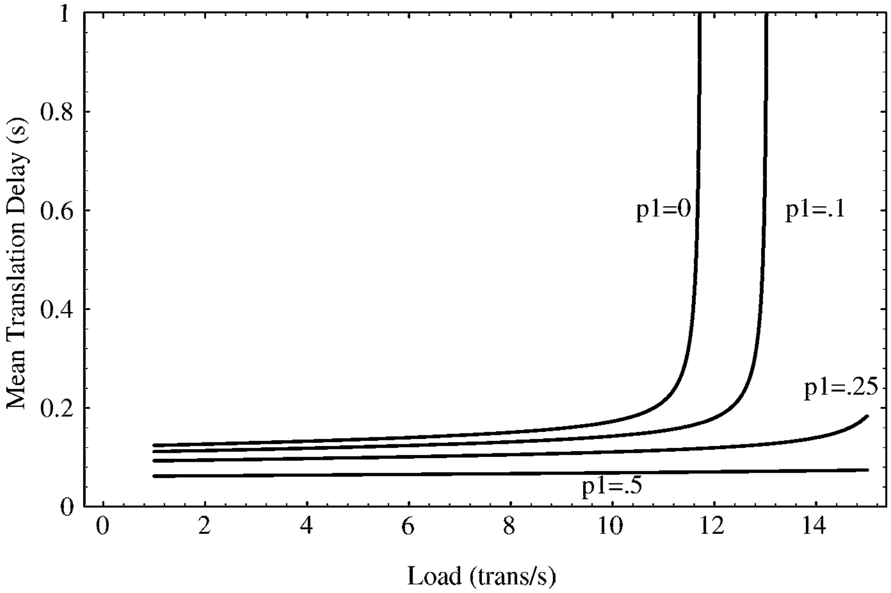

Figure 5: One-stage version: Mean translation delay for $f_1 = 2$ and $1/\mu_{t1} = 5ms$; p_1 varied.

In Fig. 5 we plot the mean delay assuming some load imbalance due to hashing at VLR_1, and that a cache is installed at all the VLRs. (We continue to assume that the hashing at all VLRs other than VLR_1 results in perfect load balance.) It is clear that caching can result in very substantial reductions in total translation delay.

3.2 Two-stage version

To model the two-stage version, we set $Stage = 1$. In addition, we assume that processing at the first-stage TS now simply consists of performing a second-level hash function and other minor tasks, so that $1/\mu_{t1} = 1/\mu_{t1c} = 1.25ms$, while the actual database lookup is done at the second level TS, so that $1/\mu_{t11} = 5ms$.

We assume that the two-stage version of the scheme will only be used when translation rates have become very high due to high PCS and NGPN penetration. Thus we consider the scenario where the number of active users per PCS cell, $n = 30$, and $\lambda = 19.2$ in the busy hour.

Fig. 6 shows the variation of the mean delay with translation load when there is some load imbalance at VLR_1 ($f_1 = 2$), and there are two second-level TSs, i.e., $1/\beta = 2$. (We continue to assume that the hashing at all VLRs other than VLR_1 results in perfect load balance.) We consider the situation where no caching is used at the VLRs ($p_1 = 0$). We can see that even without the use

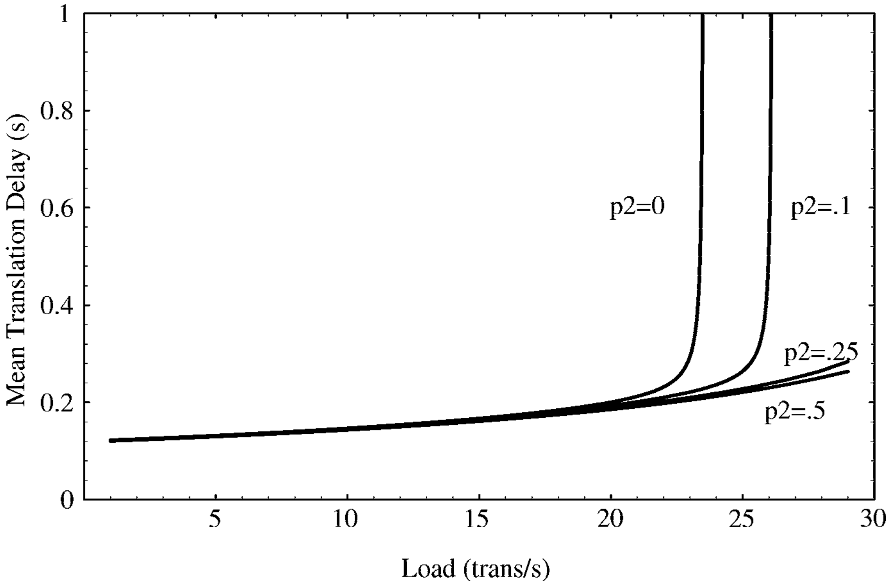

Figure 6: Two-stage version: Mean translation delay for $f_1 = 2$, $1/\mu_{t1} = 1.25ms$, and $1/\mu_{t11} = 5ms$; p_2 varied.

of a second-level cache (i.e, $p_2 = 0$), the effect of the second-level TSs is to allow a much greater translation load to be processed; saturation occurs at an offered load of around $\lambda = 22$ instead of the $\lambda = 11$ trans/sec in the one-stage case.

The use of a second-level cache is very effective for $0.1 \leq p_2 \leq 0.25$ in this example; for $p > 0.25$ the second-level cache provides diminishing returns. Since there are far fewer TSs than VLRs, second-level caches may be more cost-effective than caching at the VLRs. Note, however, that at low loads ($\lambda \leq 20$ trans/sec.) the VLR cache reduces the mean translation delay while the second-level cache does not; this is because at low loads the delay due to VLR processing and SVC setup is still a significant fraction of the total delay, and is not affected by second-level caching.

4 Conclusions

We have developed a simple model for the mean translation delay due to the one-stage and two-stage versions of our proposed NGPN scheme. Example calculations with the model show that the mean translation delay can be kept low (under 0.5 second), with appropriate use of caching and second-level translation servers if necessary. The model is fully parameterized and can be used to investigate other example scenarios if desired. At present the model does not provide variances or percentiles of delay; these can be incorporated if the translation delay becomes a more significant proportion of the total call setup time.

Acknowledgments. Thanks are due to Li-Fung Chang, Matthew Cheng, Greg Pollini and Subhashini Rajagopalan for useful and enlightening discussions.

References

[1] Bellcore. Network and Operations Plan for access services to Personal Communications Services (PCS) providers. Special Report SR-TSV-002402, Bellcore, Aug. 1992.

[2] Bellcore. CCS network interface specification (CSNIS) supporting SCCP and TCAP. Generic Requirements GR-1432-CORE, Bellcore, 1993.

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

[4] Bellcore. PCS Network Access Services. Special Report SR-TSV-002459, Bellcore, Dec. 1994.

[5] R. Jain, S. Rajagopalan, and L.-F. Chang. A hashing scheme for phone number portability in PCS systems with ATM backbones. In *Proc. IEEE Conf. Pers. Indoor Mobile and Radio Comm. (PIMRC)*, Oct. 1996.

[6] R. Jain, S. Rajagopalan, and L.-F. Chang. Phone number portability for PCS systems with atm backbones using distributed dynamic hashing. *IEEE J. Sel. Areas Comm.*, 15(1):96–105, 1997. Special Issue on Wireless ATM.

[7] Ravi Jain, Yi-Bing Lin, and Seshadri Mohan. Location strategies for personal communications services. In J. Gibson, editor, *Mobile Communications Handbook*. CRC Press, 1996.

[8] Leonard Kleinrock. *Queuing Systems Volume II: Applications.* Wiley, 1976.

[9] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik. *Quantitative System Performance.* Prentice-Hall, 1984.

[10] S. Mohan and R. Jain. Two user location strategies for PCS. *IEEE Pers. Comm. Mag.*, 1(1), First Quarter 1994. (Premiere issue).

[11] T.-H. Wu and L. F. Chang. Architectures for PCS mobility management on ATM transport architectures. In *Proc. Intl. Conf. Univ. Pers. Comm.*, pages 763–768, 1995.