Analysis of an Enhanced Signaling Network for Scalable Mobility Management in Next Generation Wireless Networks

Jeremy Abramson, Xiao-yan Fang and Dipak Ghosal
Department of Computer Science and Department of Electrical and Computer Engineering
University of California, Davis
Davis, CA 95616
abramson@cs.ucdavis.edu, jamesfxy@ece.ucdavis.edu, ghosal@cs.ucdavis.edu

Abstract—In order to support new Personal Communication Services (PCS) in 2.5G and 3G networks, the underlying signaling network must support scalable and fault-tolerant mobility management. A mobility management architecture based on multiple Home Location Registers (HLRs) and multiple Gateway GPRS Support Nodes (GGSNs) can meet both of these requirements. In this paper, we consider an enhanced network architecture in which the mobility management databases, in particular, the HLRs, are interconnected by a TCP/IP network. The primary function of this additional network interface will be to support synchronization of the replicated and/or distributed HLRs. We outline the architecture of the converged signaling network and discuss the integration with the GPRS network. We then present analytical models to quantify the benefit of distributed HLR and replicated GGSN architectures. Our results show that a distributed HLR architecture can reduce the update and query delay. Finally, by appropriately load balancing the multiple GGSNs, the mean packet delay can be significantly reduced.

I. INTRODUCTION

In standard GSM [3], mobility management is handled by a two-tier scheme. Every user (hereafter referred to as a mobile) is associated with a database called the Home Location Register (HLR). This database consists of profiles that contain information about each mobile’s subscribed services and payment information, as well as a pointer to the Visitor Location Register (VLR) that is currently serving the mobile. When a mobile moves from an area governed by one VLR to an area governed by a different VLR, the mobile’s profile must be updated. This procedure is called a location update.

General Packet Radio Service (GPRS) [3] is a standard that provides mobile Internet access to GSM systems. Two nodes are added to the standard GSM system: the Gateway GPRS Gateway Support Node (GGSN) and the Serving GPRS Support Node (SGSN). The GGSN acts as a gateway between the Internet (and other GGSNs) and a provider’s private GPRS network [3]. The GGSN’s main function is to tunnel packets from outside networks (other GPRS networks and the Internet) to the SGSN currently serving the mobile. It does this through an IP based GRPS backbone. The SGSN acts in much the same way as the VLR in GSM. It provides mobility management for the GPRS network, as well as other services (including ciphering and billing). To provide a mobile with packet services, the SGSN establishes two contexts: an IP based Packet Data Protocol (PDP) context, connecting a packet flow with the GGSN, and a Mobility Management context, logically connecting the mobile and the SGSN. The SGSN then has both a logical link-layer connection to the mobile and is one end of an IP tunnel to the GGSN. Note that the SGSN has both an SS7 address (to facilitate location updates using standard GSM MAP [3]) and an IP address.

In this paper we study the performance of multiple HLR in the GSM signaling network and multiple GGSN in the GPRS network. The motivation is that distributed mobility management can meet both the requirements of scalability and high availability demanded by the next generation PCS services. There are a number of aspects that limit the deployment of a multiple HLR in the current Signaling System 7 (SS7) network. Perhaps the most important is that the existing SS7 network does not provide advanced services such as multicasting that are required to support replicated HLR databases. In this paper, we consider an enhancement to the existing SS7 network, which can support replicated HLR databases. Specifically, we consider a TCP/IP overlay network to interconnect multiple HLR databases; the TCP/IP network is primarily used to maintain synchronization of the profiles in multiple HLRs. We present an analysis to quantify the tradeoffs when HLR are organized in a purely distributed or replicated mode. Based on our analysis, we determine an “optimal” number of HLRs for a given user population.

The GGSN tunnels packets to and from the SGSN currently serving the mobile. For this purpose, it maintains a PDP context and a mobility management context for each mobile. In order to ensure low delay across the GGSN, we consider a network architecture with multiple GGSNs. Since data traffic is very bursty, in order to minimize the delay it is important to ensure that the multiple GGSNs are load balanced. While there are many different approaches to load balance the GGSNs, the most simple and efficient method would be one in which incoming and outgoing packets are distributed to the multiple GGSN in a round-robin manner. This, however,

∗ This work was funded in part by NSF grant NCR-9703275 and ANI 9714668.
requires that the PDP context of each mobile be replicated in all the GGSNs. In this paper, we study this approach and quantify the benefit using an analytical model. Our results show that using multiple GGSNs decreases the mean packet delay when overall capacity is kept the same.

The remainder of this paper is organized as follows. In the next section (Section II) we describe the enhanced signaling architecture. In particular, we discuss the TCP/IP overlay network that interconnects multiple HLRs and describe the mobility management functions in this enhanced network. In Section III, we present a model to quantify the benefits of distributed and replicated HLR schemes. In Section IV, we describe the multiple GGSN architecture and present an analysis that quantifies the performance improvement in terms of lower delay. In Section V, we discuss the related literature. Finally, in Section VI, we conclude with a discussion of how the proposed enhancement can evolve to a yet more scalable and fault-tolerant signaling network.

II. AN ENHANCED SIGNALING ARCHITECTURE

To support multiple HLR in the existing SS7 network, we consider a TCP/IP overlay network that interconnects the HLRs. In this architecture, shown in Fig. 1, each HLR has a TCP/IP interface. The primary use of this interface is to provide a scalable route for HLRs to exchange synchronization messages that are required for database concurrency control. Note that a GGSN already has both an SS7 and a TCP/IP interface. The GGSN interact with the HLR though the SS7 network. There are other advantages to this converged signaling network. First, since additional bandwidth can be added to the TCP/IP network as required, higher degree of replication can be supported. This is true both with regards to the required link bandwidths as well as the routing capacities that are required for routing the additional synchronization messages. The TCP/IP network can in fact be built over the circuit switched network. If large degree of replication can be supported, one can potentially replicate the HLR database in each MSC [1]. Secondly, this architecture can also support multiple GGSNs. The new TCP/IP interface at the HLRs can be used for querying PDP context information. Finally, this may enable new types of mobile applications.

The above converged network enables HLRs to be organized in replicated or distributed modes. In the case of the latter, the mobiles will be partitioned among the HLRs with each mobile’s profile being stored in one HLR. Each HLR also control a fixed subset of the VLRs. When a mobile moves between VLRs that are controlled by the same HLR, only the corresponding HLR needs to be updated. When the mobile moves to a VLR controlled by a different HLR, the new HLR obtains the profile form the old HLR and send and sends deregistration message to the old HLR. In the replicated mode, the mobile’s profile is replicated in each HLR. Whenever the mobile moves, all the HLRs need to be updated. Clearly, these schemes present different tradeoff with regard to update and query costs.

As mentioned before, the GGSN is the ingress point into the GPRS network. All packet-switched traffic to and from mobiles in the provider’s network is routed through the GGSN. State information for all users is maintained in the GGSN, as well as context information for all open connections. So while the GGSN cannot be considered an “active agent”, it clearly requires more processing power than a normal router, and therefore can become a bottleneck under high load. As the number of GPRS users is expected to increase dramatically, a replicated GGSN architecture can be implemented to transparently provide scalability and fault-tolerance to the GPRS network. There are many issues to be resolved in creating a replicated GGSN architecture. First, the GGSN maintains state information for mobiles in the network. This information will now be replicated, and all copies will need to be synchronized. Second, load balancing is critical to the stability and efficiency of the replicated system. Finally, the replicated architecture should be transparent to outside networks (the Internet).

In a replicated GGSN architecture, incoming packets destine for a GPRS mobile will first reach the load balancer¹. The load balancer will have a hash table of open sessions. It will hash the destination address and port number, and determine the route the packet should take (i.e. which GGSN is currently serving the user). If there is no hash entry, one will be created, and a new GGSN will be assigned to the user, using a load-balancing scheme. This scheme can be simple as simple maintaining a counter of how many open connections each GGSN has and choosing the one with the least number of open connections. More complex methods of load balancing, such as having each GGSN report their load or throughput can also be implemented. The problem of user-state coherency is solved in much the same manner as the replicated HLR scheme. Upon a location update, the SGSN will simply multicast the location update message to all the GGSNs. An SGSN can then query any of the GGSNs to determine user state information.

Note that, since the load balancer takes the place of the “original” GGSN, outside networks such as the Internet will

¹ Note that the load balancer does little computation and should not be a bottleneck even at high loads.
have no knowledge of the internal configuration of the GGSN cluster. GGSNs can be added or removed as the need arises. Only list of GGSNs at the load balancer is affected when a server is added or removed.

III. QUANTIFYING THE BENEFIT OF MULTIPLE HLR

In order to quantify the benefit of multiple HLRs, we develop an analytical model. In this model, we consider a single HLR and aggregate the effects of the other HLRs into the arrival rate at this, our “tagged” HLR. The arrival rate of messages at this queue consists of the following components:

- **Incoming calls.** Messages querying the location of mobile currently under the domain of our tagged HLR. Denoted \( \lambda_i \). A mobile will receive a call from a mobile in the same HLR with probability \( \beta \), and a call from a mobile in a different HLR with probability \( 1-\beta \).

- **Outgoing calls.** Queries from mobiles under the domain of the tagged HLR to other users, denoted \( \lambda_o \). A mobile will place a call to a mobile within the same HLR with probability \( \alpha \), and will move to a different HLR with probability \( 1-\alpha \).

- **Location Updates.** When a mobile moves from one VLR to another VLR, a location update message is generated at the HLR currently governing the VLR that the mobile has just moved in to. A mobile will move to a VLR governed by the same HLR with probability \( \alpha \), and will move to a VLR governed by a different HLR with probability \( 1-\alpha \).

The overall message rate for location updates is given by \( \lambda_o \).

We will model two different schemes, a **fully replicated** (FR) scheme and a **fully distributed** (FD) scheme. To simplify analysis, we partition the mobiles equally among the HLRs. We further assume that whenever a mobile moves out of the domain of an HLR, a different mobile moves in to replace her.

In the FR scheme, any location update (be it inter-HLR or not) generates a broadcast location update message to all the HLRs. Also, since the scheme is fully replicated, all calls (be they to remote or local users) can be serviced locally without the need for remote queries. Therefore, if \( R \) is the number of HLRs, the total message rate at our tagged HLR is then given by:

\[
\lambda_m = \frac{\lambda_i}{R} + \frac{\lambda_o}{R} + \frac{\lambda_o}{R}
\]  

(1)

Note that when \( R=1 \), we get the expected result of having our overall message rate equal to \( \lambda_i + \lambda_o + \lambda_o \).

In the fully distributed scheme whenever an Inter-HLR location update occurs, the new HLR will send a broadcast invalidation message to the other HLRs. So, the messages rate at our tagged HLR due to mobility is given by:

\[
\frac{\lambda_m}{R} + \left[ \frac{\lambda_o}{R} \right] \left( 1 - \frac{\alpha}{\sqrt{R}} \right) \left( R - 1 \right)
\]  

(2)

The first term is due to mobility that can be resolved locally. The second term is the aggregated affects of users moving “away” from our tagged HLR, and the invalidation messages the other R-1 HLRs will broadcast. Note that as the number of HLRs increases, the likelihood of a movement being “local” to one HLR decreases, which explains the \( \frac{\alpha}{\sqrt{R}} \) term. The equations for incoming and outgoing calls are the same, the only change being the differing parameters (replacing, for example, \( \lambda_m \) and \( \alpha \) with \( \lambda_i \) and \( \beta \) [5]).

![Figure 2. Delay as a function of the number of HLRs](image)

To analyze this system, we assume 100,000 users, each of which make 3 calls an hour (with an equal likelihood of originating or terminating a call). We also assume four location updates an hour per user. We set \( \alpha, \beta, \) and \( \gamma \) to be .95. With 1 HLR (i.e. \( R = 1 \)), we find that \( \mu = 270 \) queries/second gives us approximately 75% utilization. Fig. 3 shows how delay across the queue at the tagged HLR varies as a function of the number of HLRs.

For FD scheme, \( R=3 \) provides the minimal delay across the queue at our tagged HLR. The delay across our tagged HLR will continue to decrease in the FR scheme, due to the fact that queries can be served locally. The overall network traffic is increased in the FR scheme, however. Note that in the above experiments the capacity of the tagged HLR is kept constant. This implies that the total system capacity increases linearly with \( R \), the number of HLRs.

IV. ANALYSIS OF REPLICATED GGSN ARCHITECTURE

All packet-switched traffic to and from mobiles in the provider’s network is routed through the GGSN. State information for all users is maintained in the GGSN, as well as context information for all open connections. So while the GGSN cannot be considered an “active agent”, it clearly requires more processing power than a normal router, and therefore can become a bottleneck under high load. As the number of GPRS user is expected to increase dramatically, a replicated GGSN architecture can be implemented to transparently provide scalability and fault-tolerance to the GPRS network.
Many problems arise when implementing a replicated GGSN architecture. First, the GGSN maintains state information for mobiles in the network. This information must be replicated and all copies must be synchronized. Second, load balancing is critical to the stability and efficiency of the replicated system. Finally, the replicated architecture should be transparent to outside networks (the Internet).

Incoming packets destined for a GPRS mobile will first reach the load balancer. The load balancer will have a hash table of open sessions. It will hash the destination address and port number, and determine the route the packet should take (i.e. which GGSN is currently serving the user). If there is no hash entry, one will be created, and a new GGSN will be assigned to the user, using a load-balancing scheme. This scheme can be simple as simple maintain a counter of how many open connections each GGSN has a choosing the one with the least number of open connections. More complex methods of load balancing, such as having each GGSN report their load or throughput can also be implemented.

The problem of user-state coherency is easily solved in much the same manner as the replicated HLR scheme given in Section 3. Upon a location update, the SGSN will simply multicast the location update message to all the GGSNs. An SGSN can then query any of the GGSNs to determine user state information.

Lastly, since the load balancer takes the place of the “original” GGSN, outside networks such as the Internet will have no knowledge of the internal configuration of the GGSN cluster. GGSNs can be added or removed as the need arises. Only list of GGSNs at the load balancer is affected when a server is added or removed.

In this paper, we study the performance of data flow from the servers in the Internet through the GGSN to the mobile users. Typically, there will be more data flow in this direction. The problem of user-state coherency is easily solved in much the same manner as the replicated HLR scheme given in Section 3. Upon a location update, the SGSN will simply multicast the location update message to all the GGSNs. An SGSN can then query any of the GGSNs to determine user state information.

In this paper, we study the performance of data flow from the servers in the Internet through the GGSN to the mobile users. Typically, there will be more data flow in this direction. The reverse direction will primarily consist of small request packets and acknowledgements. While we consider a queuing model consisting of two sources, and four GGSN nodes, the approach developed here can be generalized to arbitrary number of GGSN nodes. Each source is modeled by a two states MMPP (Markov Modulated Poisson Process) consisting of an ON state and an OFF state. In the ON state, packets are generated according to a Poisson process with rate \( \lambda_1 \) packets/sec, while in the OFF state no packets are generated, i.e., \( \lambda_2=0 \) packets/second. The source turns on with rate \( \alpha \) and turns off with rate \( \beta \). Each GGSN has the capacity of \( \mu \) packets/sec with an infinite queue.

We study one of the tagged GGSN whose level state diagram are shown in Figures 4 and 5. The state is defined as \( (m, b, a) \), where \( m \) is the packet in the tagged GGSN node, \( b \) is the number of sources in the ON state transmitting packets to the tagged GGSN node, and \( a \) is the number of source in the ON state transmitting packets to the rest of the GGSNs.

\[
\begin{align*}
\gamma_1 &= \frac{N-k}{N} \lambda, \\
\gamma_2 &= \frac{k}{N} \lambda
\end{align*}
\]  

We represent three different load distribution algorithms by the value of \( k \). If \( k=4 \), it is the non-distribution policy in which all packets are routed to the tagged GGSN node. If \( k=1 \), it is the complete-distribution policy in which all packets are evenly distributed among all the GGSN nodes. While 4>\( k \)\( >1 \) correspond to the partial-distribution policies in which packets are unevenly distributed among the GGSN nodes. We compare the mean packet delay for these different distribution policies.

Using standard matrix-geometric approach the mean packet delay can be derived. The results are shown in Figure 5. As we can see the complete-distribution has the best result in terms of mean packet delay while the non-distribution scheme has the worst performance. It verifies that replicated GGSN
architecture with appropriate load balancing can improve the quality of service by decreasing mean packet delay.

Our work differs from previous work in that it addresses the underlying infrastructure - the signaling network - that allows any mobility management scheme be it hierarchical, cached or pointer-based to be implemented. In order to implement a scalable, fault-tolerant and efficient mobility management scheme, the signaling network must be enhanced. Studies such as [11] and [12] consider all IP based mobility management based on standards such as Mobile IP, Cellular IP and HAWAII for wireless data applications.

VI. CONCLUSIONS

In this paper we have shown how key network elements such as the HLR in GSM and the GGSN in GPRS can be replicated in order to decrease delay as well as increasing fault tolerance. In the case of GSM, we proposed augmenting the existing SS7 network with a TCP/IP overlay network to support replicated and distributed mobility management functions. Using analytical models, we explored the benefits of multiple HLRs and GGSNs in the network. The proposed converged network is a step in the evolution to an IP-centric cellular network infrastructure. Future work will focus on the continued development of the analytical models, and will consider network enhancements such as the caching of profiles and pointer forwarding, as well as simulation results to further support the benefit of multiple HRL/GGSN architectures. A move toward an end-to-end IP signaling infrastructure will also be explored.

V. RELATED LITERATURE

Many studies in the literature have considered a hierarchical approach to location management, with tree-like database structure. In a hierarchical approach, leaf databases correspond to registration zones, and store user profiles for users currently residing within a specific zone. Higher level databases store a user ID + database ID pointers that points to a database at the next lower level in the tree. Clearly, the root must store pointers for all users in the network. Optimizing placement of user profiles is the goal of [2], [6] and [7]. A threshold-based, offline approach is offered in [6], where a per-user local call mobility ratio (LMCR) is determined, and profiles are disseminated to leaf databases if their benefit meets a certain threshold based on the users LMCR. In [2], the LMCR calculation is done on-line; (i.e. while the system is active) and profiles can migrate from one leaf database to another if the new database is deemed more advantageous. An advanced replication algorithm is given in [7], which adds complexity to the network model and considers network link cost and user mobility patterns. Another hierarchical approach is presented in [4]. The databases in the tree maintain location information for all mobiles being served by nodes in the tree rooted below them, as well as the nodes in the sub-tree rooted at a sibling node. This hierarchy attempts to minimize the cost of multiple database queries, while being less expensive than a flat fully replicated architecture. Other techniques, such as pointer forwarding [8], local anchoring [9] and caching [10] have proposed to minimize call setup times and overall network traffic.

Our work differs from previous work in that it addresses the underlying infrastructure - the signaling network - that allows any mobility management scheme be it hierarchical, cached or pointer-based to be implemented. In order to implement a scalable, fault-tolerant and efficient mobility management scheme, the signaling network must be enhanced. Studies such as [11] and [12] consider all IP based mobility management based on standards such as Mobile IP, Cellular IP and HAWAII for wireless data applications.

VII. REFERENCES