Development of the Home Location Register/Authentication Center in the CDMA Mobile System

Sun Bae Lim, Kyeongsuk Shin, and Hyungon Kim

CONTENTS

ABSTRACT

In this paper, a home location register (HLR) for CDMA mobile communication system (CMS) is introduced. It stores the mobile station (MS) subscribers' locations and supplementary service information. Call processing procedures for HLR are developed to receive and store subscriber's location coming from mobile exchange (MX) during the location registration, and to transfer subscriber's location and supplementary service information to the MX during the mobile-terminated call setup. For fast call processing by increasing database access speed, a memory-resident database management system is devised. For easy and secure HLR operation, administration and maintenance functions and overload control mechanisms are implemented. Designed HLR hardware platform is expandable and flexible enough to reallocate software blocks to any subsystems within the platform. It is configurable according to the size of subscribers. An authentication center (AC) is developed on the same platform. It screens the qualified MS from the unqualified. The calls to and from the unqualified MS are rejected in CMS. To authenticate the MS, the AC generates a new authentication parameter called "AUTHR" using shared secret data (SSD) and compares it with the other AUTHR received from the MS. The AC also generates and stores seed keys called "A-keys" which are used to generate SSDs. The HLR requirements, the AC requirements, software architecture, hardware platform, and test results are discussed.

I. INTRODUCTION

In recent mobile communications systems, databases called home location register (HLR) are used for storing and providing subscribers' information [1]. These databases make mobile switching systems free from managing user dependent services and user related information resulting in easier service development and more flexible service provision than before. As a database of the CDMA mobile communication system (CMS), HLR is developed which stores and provides the mobile station (MS) subscribers' locations and supplementary service information. As the MS moves while its power is on, its location information is transferred to the HLR via the mobile exchange (MX). The location information is transferred to the MX at the request of the MX to setup a mobile-terminated call. The supplementary service information is stored into the HLR at subscription time and handed to the MX at location information update time so that the MX can reference it at call setup time. To handle many location updates and call setups in seamless and real time manner, we analyzed our requirements as follows:

- *Fault tolerance*
  To provide continuous and uninterrupted service, the HLR must be fault tolerant.
- *CCS No.7 connectivity*
  The standard signaling protocol CCS No.7 [2] is used to communicate with the MX.
- *Call processing*
  Location registration, incoming call processing and supplementary services are major functions.
- *Authentication processing*
  The authentication processing must be done prior to location registration, outgoing call processing or incoming call processing.
- *Memory-resident database*
  GSM recommends that the HLR must handle 1.8 location registrations and 0.4 call setups per hour per subscriber [3]. The HLR must complete a location registration and a call setup in two seconds and one second, respectively. A call setup needs one database transaction and a location registration needs two database transactions. If the HLR is to serve 500,000 subscribers, it must handle $(500{,}000\times(1.8\times2+0.4))$ / 3,600 = 555 transactions in a second. Authentication needs one database transaction for location registration, outgoing call processing and incoming call processing respectively. This means that the AC takes additionally a comparable amount of transactions for the MS authentication. To process this vast amount of transactions, a memory-resident database system must be introduced.
- *Operation, administration and maintenance*
  The HLR must have operation and maintenance (OA&M) functions.

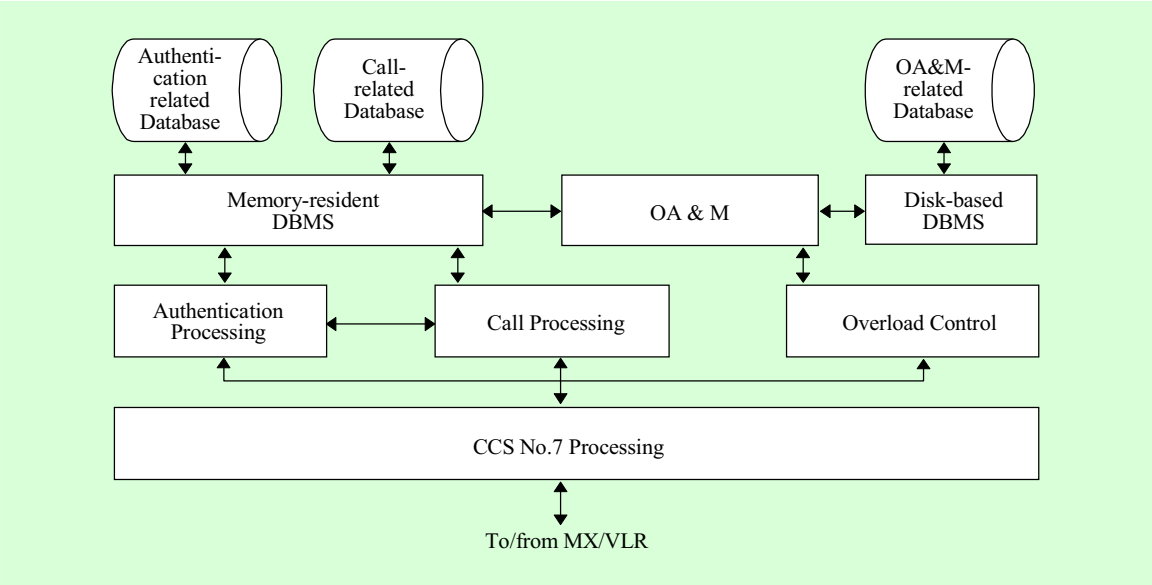

Fig. 1. Software architecture of the HLR/AC.

- *Overload control*
  The overload control mechanisms must prevent the HLR from malfunctioning under heavy traffic load.
- *Expandable hardware platform*
  The capacity of the HLR must be commensurate with the size of subscribers. For this, the HLR hardware platform must be designed to be expandable and flexible enough to reallocate software blocks to any subsystems within the platform if necessary.
- *The AC platform*
  The AC must be designed either to be integrated into the HLR platform or to be separated into the independent hardware platform.

We fulfilled all the above requirements for the HLR except a commercial fault tolerant computer and a CCS No.7 message transfer part processor (CMP) to meet the first two requirements .

In this paper, we introduce the development of the HLR and the AC. Section II discusses software architecture, section III deals with expandable and flexible hardware platform, section IV describes test and verification, and finally section V concludes this paper.

II. SOFTWARE ARCHITECTURE

The HLR/AC software architecture consists of seven major different functional blocks as shown in Fig. 1. The AC shares memory-resident DBMS, disk-based DBMS, OA&M, overload control and CCS No.7 processing with the HLR. This architecture can make the AC integrated into the

HLR or separated into the new independent platform if necessary.

The memory-resident DBMS deals with call-related data. The disk-based DBMS deals with OA&M-related data. The OA&M monitors the HLR's operation and reports its status to the operator. The OA&M also takes commands from the operator and executes them. The call processing receives the MX's requests and responds them to provide the necessary information during the call setup. The authentication processing screens the qualified MS user and allows the qualified MS to access the CMS network for the subscribed services. The overload control mechanism monitors the input/output message flows and system resources of the HLR. It initiates overload control activities when it detects the HLR overload. CCS No.7 processing includes message transfer part (MTP), signaling connection control part (SCCP), and transaction capabilities application part (TCAP).

In the following, we discuss call processing, authentication processing, database management, administration and maintenance, and overload control.

1. Call Processing

The HLR traces roaming mobile stations, stores their location information and provides the location information to the MX during the call setup [4]-[6]. As the MS moves while its power is on, its location information is transferred to the HLR via the MX. When the MX requests the HLR of a routing information to setup a mobile-terminated call, the HLR responds to send the location information to the MX. The HLR also stores supplementary service information which includes outgoing call barring, incoming call barring, call forwarding, supplementary service activation/deactivation, mobile station type, mobile inactive/active status, and access deny reasons. The supplementary service information is stored into the HLR at subscription time and handed to the MX at location information update time so that the MX can reference it at call setup time.

For implementation of the functions above, we defined our own mobile application part (MAP) protocols and designed database schemas for handling the location information and supplementary service information. The developed HLR's main call processing functions include the followings:

- *Location registration/cancellation*
  Location registration is an operation to store the newly changed location of a roamed mobile station in the HLR and to send the subscriber's profile to an appropriate visitor location register (VLR). Location cancellation is an operation to request the previous VLR to delete the roamed subscriber's profile.
- *Routing information interrogation*
  Routing information interrogation is a procedure for the MX to request the HLR location information of a mobile station during the mobile-terminated call setup.

- *Incoming call transfer*
  Incoming call transfer is a kind of supplementary services that an incoming call to a subscriber is forwarded to a designated number. The HLR stores and manages the designated number information for this service.
- *Mobile station activation/deactivation*
  The HLR stores MS activation/deactivation information by receiving CCSactive/CCSinactive messages from the corresponding VLR. The terminated call to the deactivated MS is immediately rejected resulting in saving unnecessary traffic flow.

2. Authentication Processing

Authentication is a procedure to screen the qualified MS users and to allow the qualified MS to access the CMS network for the subscribed services. Authentication processing must be done prior to location registration, mobile-terminated call setup or mobile-originated call setup.

The authentication scheme is based on the so called "challenge and response" technique operating as follows. The MS produces its authentication challenge parameter called "authentication response (AUTHR)" using subscriber authentication algorithm for CMS (SAC) with random number (RAND), shared secret data (SSD), electrical serial number (ESN) and mobile identification number (MIN). (In case of call origination, dialed digits are used instead of MIN.) The AUTHR parameter is sent to the AC with the RAND (mobile-initiated authentication). The AC produces a new AUTHR using its own SAC with the stored SSD and the received RAND from the MS. If the new AUTHR is the same as the old one received from the MS, the MS is qualified to access the network for the subscribed services [7]-[10]. If the new AUTHR is different from the old one, the AC initiates the so called "unique challenge" procedure to attempt the failed authentication once more for the MS (AC-initiated authentication). Since it is an important parameter to produce AUTHR, the SSD must not be disclosed to any unqualified users. If the SSD is suspicious of being disclosed to unqualified users, the AC initiates the so called "SSD update" procedure to change SSDs both in the AC and the MS.

For the authentication processing mentioned above, the AC stores and manages the following information.

- SAC and SSD which are the same as those stored in the MS stores.
- Authentication-related information which includes ESN and MIN of each MS.
- Random number generators and SSD generation algorithm.
- Authentication policy which includes retry options, authentication restrictions and pre-set decisions to be applied if the authentication trial fails.
- Seed keys and algorithms to generate SSDs.

To implement the functions above, we defined our own authentication mobile application part (MAP) protocols, and designed authentication database schemas and authentication algorithms. The developed AC's main call processing functions include the followings:

- *SAC algorithm*
  It is used for generating SSDs and AUTHRs. It is stored both in the MS and the AC.
- *Registration authentication procedure*
  This authentication procedure is used prior to location update procedures. The MS produces AUTHR parameter and sends it to the AC with the RAND. The AC produces a new AUTHR using its own SAC with the stored SSD and the received RAND from the MS. If the new AUTHR is the same as the old one received from the MS, the MS is qualified to access the network for the subscribed services.
- *Call origination authentication procedure*
  This authentication procedure is used prior to call origination procedures and works in the same manner as registration authentication procedure.
- *Call termination authentication procedure*
  This authentication procedure is used prior to call termination procedures and works in the same manner as registration authentication procedure.
- *SSD update procedure*
  This procedure is invoked when the SSD is suspicious of being disclosed to unqualified users to change SSDs both in the AC and the MS.
- *Unique challenge procedure*
  This procedure is invoked when the AC produced AUTHR is different from the MS produced AUTHR to attempt authentication trial once more.

3. Database Management

The subscribers' information stored in the HLR database can be divided into two distinctive data according to their characteristics [11]. One is call-related data which need real-time processing [12], [13] and the other is operation and management-related data which do not need real-time processing. Therefore, we designed two database systems to accommodate such characteristics. The call-related data are stored in the main memory-resident database for fast access, and the operation and management-related data are stored in the disk-based database for safety.

The designed HLR database system consists of database access client (DBAC), database access server (DBAS), disk-based DBMS, query interface (QI), and memory-resident DBMS as shown in Fig. 2. The operator accesses disk-based DBMS via DBAC and DBAS. The call-related application service element (ASE) interacts with memory-resident DBMS via QI.

Function of each block is as follows:

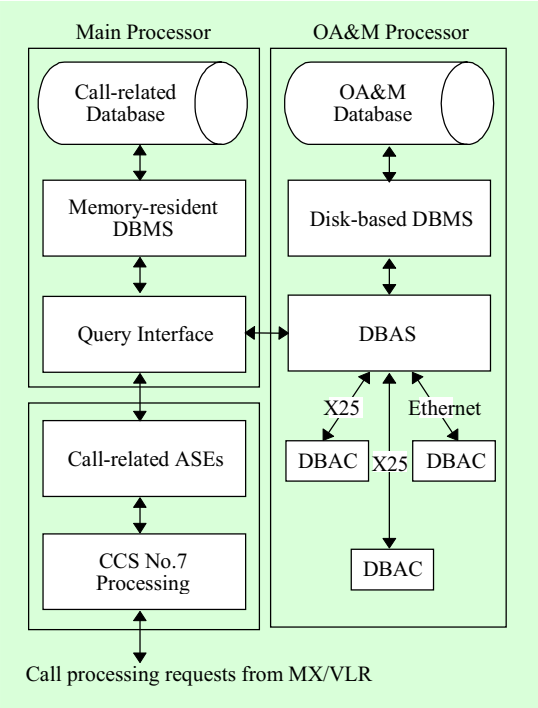

Fig. 2. Structure of the HLR database.

- *DBAC* receives the operator's request and passes it to DBAS. When DBAS responds to the request, DBAC fetches and displays it to the operator's terminal. If DBAC is a remote user, it communicate with DBAS using X.25 protocols. If DBAC is a local user, it communicates with DBAS using TCP/IP protocols. DBAC supports two-level accesses: one is administrator's access and the other is operator's access. Administrator's access has more rights than operator's access.
- *DBAS* proceeds the operators' requests via DBAC. It handles subscribers' registration and deregistration, subscribers' data retrieval and update. It also converts X.25 protocols into TCP/IP protocols and vice versa.
- *QI* maps location registration and call setup requests from call processing functional block into the database access transactions. QI also interacts with the operator via DBAS for system operation and administration.
- *Memory-resident DBMS* is a database engine which handles data on the main memory. It provides data retrieval, data insertion, data deletion and data modification functions on the main memory. The main memory database supports two indexing mechanisms. One is modified linear hashing (MLH) [14] and the other is T-tree indexing [15]. The MLH is used for fast searching for a specific data. The T-tree is used for associated but relatively slow in searching for an associated data. Main memory-resident database is much faster but less stable than disk-based database. A synchronization mechanism which performs periodic data backup from the main memory to disk is supported to increase safety and data integrity between memory-resident database and disk-based database.
- *Disk-based DBMS* is a database engine which handles data on hard disks. It also supports data retrieval, data insertion, data deletion and data modification functions on hard disks.

4. Operation, Administration and Maintenance

Operation, administration and maintenance function helps the operator to manage and operate all the other functional blocks in the HLR. It contains administrator and operator management, system status management and system configuration management.

- *Administrator and operator management* supports subscriber registration and deregistration, administrator and operator grade adjustment. There are three operation grades in HLR management. The first grade is administrator grade. Administrator can manage all operations of the HLR. The second grade is auditor grade. The auditor can audit important data and status of the HLR. The third grade is operator grade. The operator can see general information and perform non- critical operations of the HLR. The administrator has the right to do what auditor and operator can do, and auditor has the right to do what operator can do.
- *System status management* performs hardware and software status manageent. Hardware status manager initializes the HLR hardware devices and loads application software. It scans periodically hardware devices which includes processor, memory, bus, power, and fans to check their status. The software status manager checks each application process periodically and reinitializes the process if the process is abnormally terminated. The software status manager collects transaction statistics for operation, maintenance and performance evaluation. The system statistics are displayed at a graphical browser.
- *System configuration management* manages addition and deletion of hardware and software components of the HLR. It displays the current configuration of the HLR. Configurable items of the HLR includes signaling points, software blocks and hardware devices.

The structure of the system maintenance processes is shown in Fig. 3. In the figure, internal/external process communication (IXPC) is a communication agent between communication processes on hardware subsystems. To protect the HLR from hardware and software faults, a mechanism that provides fault alarms and fault treatments is supported.

- *Maintenance control block* is the superblock which initializes following blocks.
- *Execution control block* audits all processes of the HLR periodically by sending audit request messages and receiving audit response messages. If a process is stopped abnormally, execution control block reinitializes the process and resumes to send audit request messages to the new process.
- *Configuration management block* manages all hardware and software configuration of the HLR and stores them

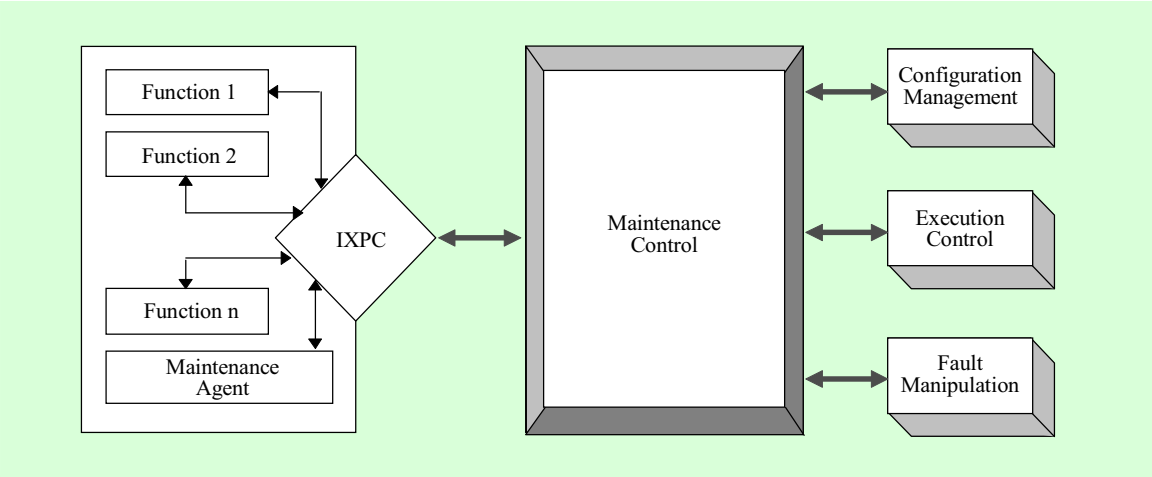

Fig. 3. Structure of system maintenance processes.

on the disk-based database. During system initialization, the configuration management block downloads the configuration data.

- *Fault manipulation block* collects faults detected by execution control block and maintenance agents (MA). The MA resides in each hardware subsystem and checks the health of the subsystems periodically to report it to the fault manipulation block. The collected faults are analyzed, classified and displayed at the system console and alarm panel. The fault status is classified into three grades as follows:
  - CRITICAL is a status that the HLR cannot serve any more.
  - MAJOR is a status that one out of two duplicated modules does not work. Major is not a stopped but serious status that the HLR cannot serve 50 % of its capacity.
  - MINOR is a status that a relatively insignificant fault occurs but does not affect the HLR to continue to serve.

All commands given by the operator are performed via command interpreter for administration and maintenance (CIAM) as shown in Fig. 4. CIAM adopts the basic features recommended in ITU-T man-machine language specification and additional user-friendly features. Addition and deletion of a command can be performed without recompilation of source codes by using a dynamic command manipulation technique.

5. Overload Control

In the CMS, several MXs are usually connected to an HLR resulting in an increased data traffic flow toward the HLR. The HLR, therefore, needs an overload control mechanism [16], [17] to prevent a catastrophic increase of the traffic.

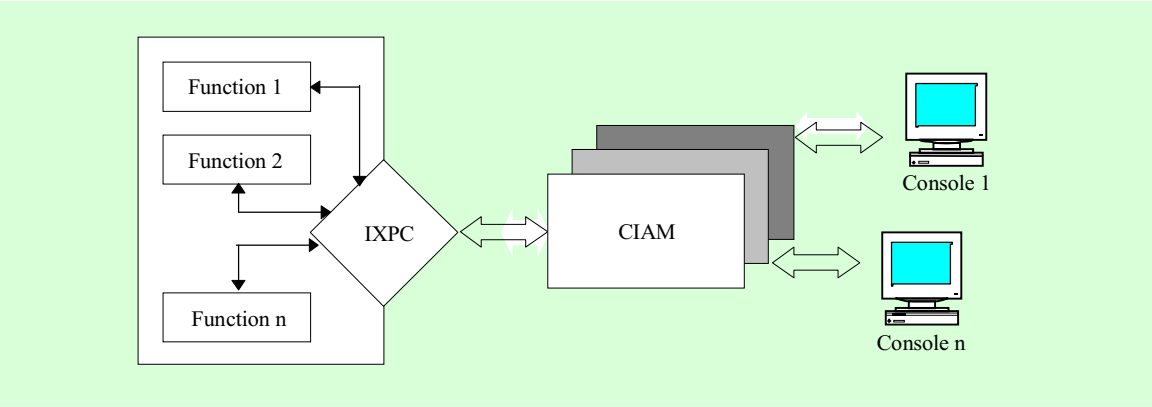

Fig. 4. OA&M command processing.

The overload control mechanism of the HLR employed works as follows [18]: When the traffic from the MXs to the HLR is increased to a dangerous load level, the HLR reduces its load level by discarding newly arrived messages using local overload control mechanism and informs the MXs of its overload status so that the MXs control messages being sent to the HLR using global overload control mechanism. The local overload control and the global overload control mechanisms are discussed below:

- *Local overload control mechanism*
  The overload control mechanism determines the number of messages that the HLR can handle in one second—permitted message per second (PMPS)—by considering the system I/O, message queue usage ratio and CPU usage ratio and updates PMPS every second. If the incoming number of messages per second (IMPS) exceeds the PMPS, the exceeded portion is discarded during the second.
- *Global overload control mechanism*
  The global overload control mechanism works between the HLR and the MXs. When it is overloaded, the HLR sends the MXs OverloadStatusReport messages to trigger the execution of the global overload control mechanism in every MX. The OverloadStatusReport message contains overload status information and load limits for unit time interval. The MXs begin to control the number of request messages being sent to the HLR. When the overload is over, the HLR sends OverloadStatusRelease messages to the MXs to stop the execution of the global overload control mechanism.

III. EXPANDABLE AND FLEXIBLE HARDWARE PLATFORM

The expandable architecture of the HLR

is based on the duplicated optical fiber rings of 110 Mbps called "highway" to which hardware subsystems are connected (Fig. 5). The major hardware subsystems connected to the rings are CCS No.7 MTP processor (CMP), front-end processor (FEP) and back-end processor (BEP).

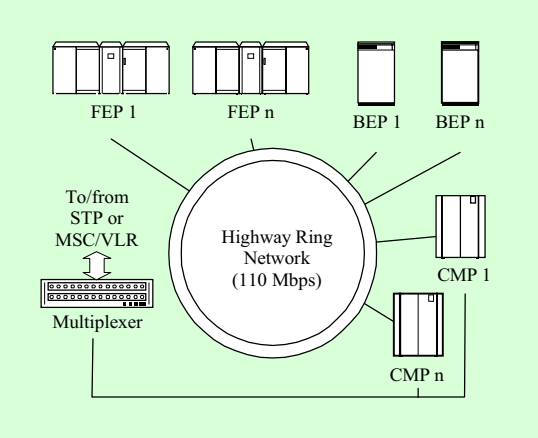

Fig. 5. Expandable hardware platform of the HLR.

The HLR hardware platform is expandable to be commensurate with the required capacity. To extend such flexibility to software, a communication mechanism between subsystems over the rings is devised. It is called internal/external process communication (IXPC). The IXPC enables any process on a hardware subsystem to communicate with any other process on any subsystem by sending messages. It uses highway interface modules at lower communication layer and routing tables for managing message queues and addresses of all processes in the HLR to route messages. By doing so, software processes can be allocated to any subsystems without any defects. In case more hardware processing power is needed, any required hardware subsystems are easily hooked onto highway ring network and software processes can be newly allocated to them without any defects. This implies that hardware subsystems are expandable while increasing software flexibility.

IV. TEST AND VERIFICATION

The test and verification of the HLR are performed in three stages as follows.

- *Local test*
  It is concerned with the basic function of every software block. The test is done by observing I/O messages of each software block. The input messages for the test are developed to simulate the output of the adjacent software block.
- *Function test*
  It tests the integrated function of the local software blocks. An MX simulator is developed to check I/O messages of the HLR from and to the MX. The I/O messages of the HLR from and to the MX simulator are based on the mobile application part (MAP) message flows [10].
- *Load test*
  It tests the maximal processing capability of the HLR. The MX simulator is developed to put maximal messages into the HLR and to count the number of processed messages from the

HLR in every second. The message load can be adjusted variously in a link and multiplied by link by link, which makes it easy to measure CPU load and find the maximal capacity limit of the FEP/BEP.

The test environment of the HLR is shown in Fig. 6.

The local and function test results indicate that those software blocks discussed in Section II perform well. The HLR hardware platform can be implemented with one CMP and one processor (FEP and BEP are integrated into one processors), and one CMP and two processors (FEP and BEP are separated into two processors) while reallocating software blocks. This has led us to believe that more CMPs and more processors can be added to the HLR hardware platform while reallocating software blocks. In other words, the hardware is expandable and the software is flexible.

The load test results of the HLR with the minimum hardware configuration—one CMP and one Tandem CM1495 processor into which FEP and BEP are integrated together are shown in Table 1 and Fig. 7. It shows that the HLR can handle 550 transactions under 89 % CPU load while meeting the timing criteria of one second for location registration and two seconds for call setup. This implies that the minimum configuration can support up to 500,000 HLR subscribers—the number of subscribers is assumed to be comparably decreased to one second if HLR and AC functions are performed together—and the number of HLR subscribers is increased approximately by 500,000 with every addition of Tandem CM1495 processor.

Table 1. Load test result of the HLR.

Transactions	System mode	User mode	CPU load (%)
0	2	0	2
100	4	0	4
200	5	2	7
300	8	3	11
400	16	6	22
450	35	11	46
500	46	12	58
550	68	21	89

Note) system mode : CPU load used by operating system kernel including device drivers for FEP and BEP
user mode : CPU load used by application software including CCS No. 7, call processing and database access

V. CONCLUSION

The HLR and AC software is implemented in seven major functional blocks. The memory-resident DBMS deals with call-related data. The disk-based DBMS deals with operation, administration and maintenance-related data. The OA&M monitors the HLR's operation and reports its status to the operator. The OA&M also

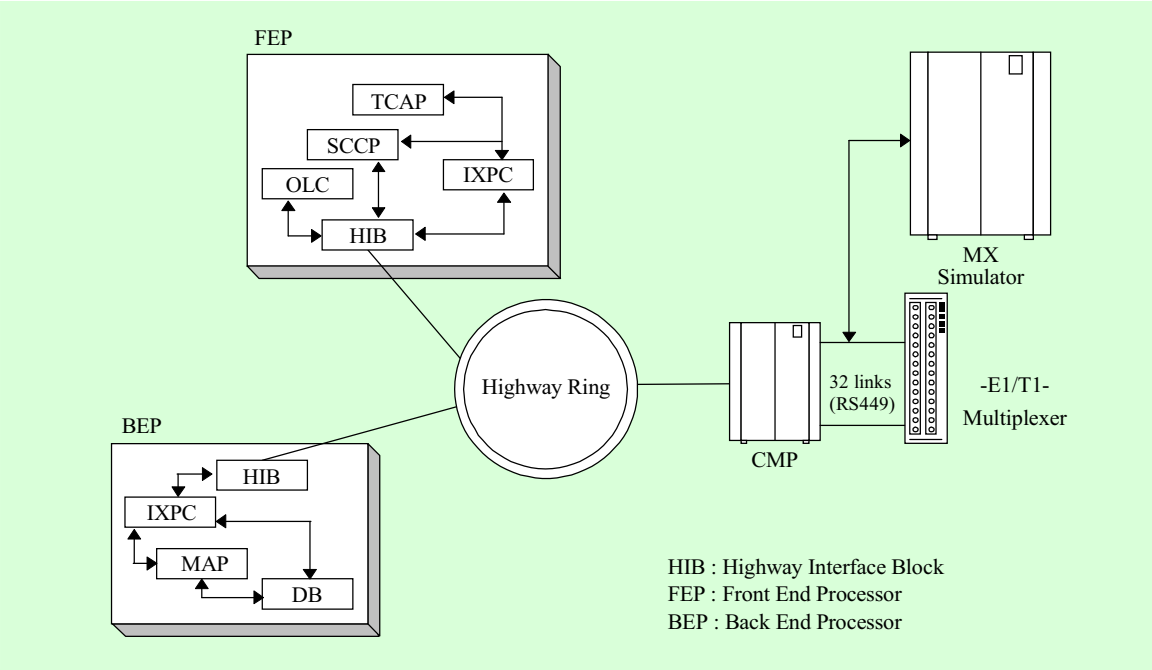

Fig. 6. Test environment of the HLR.

takes commands from the operator and executes them. The call processing receives the MX's requests and responds them to provide the necessary information during the call setup. The authentication processing screens the qualified MS user and allows the qualified MS to access the CMS network. The overload control mechanism monitors the input/output message flows and system resources. It initiates overload control activities when it detects the HLR overload. The CCS No.7 processing includes MTP, SCCP, and TCAP.

The designed HLR platform is expandable and flexible enough to reallocate software blocks to any hardware subsystems within the HLR if necessary. Flexible configurations are possible according to the size of the subscribers increasing 500,000 per Tandem CM1495 processor. The AC is either integrated into the HLR platform or separated into the independent hardware platform. The HLR/AC platform is applicable to intelligent network, personal communication services and other mobile communication networks.

ACKNOWLEDGMENTS

The HLR/AC is developed by Mobility Management Section of ETRI. Every member of the section made his or her best endeavors during the development. The authors appreciate Seungque Lee, Jong-Hyeon Lee, Sangsik Lim, Chulyi Moon, Jaesheung

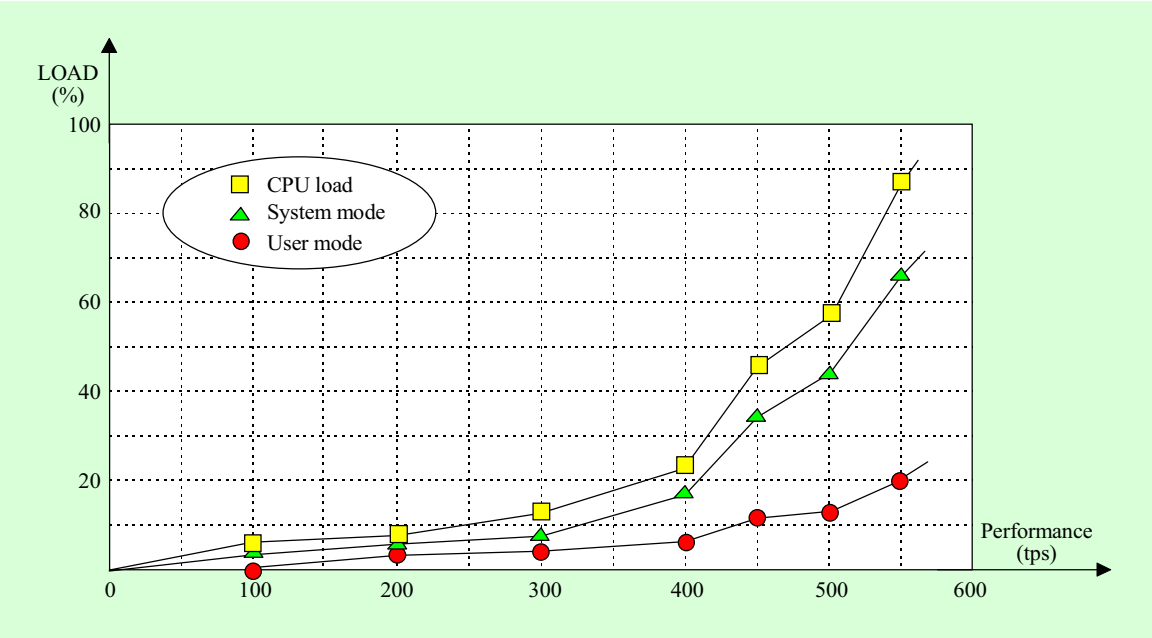

Fig. 7. Load test result graph of the HLR.

Shin, Taeg-hee Lee, Heumkeun Kang and Young-Hee Koo for their ideas, endeavors and warm-hearted cooperation. Finally, the authors express their gratitude to Dr. Keyoung Park for his advice and corrections of this paper.

REFERENCES

[1] Bijan Jabbari, "Intelligent network concepts in mobile communications," *IEEE Communication Magazine*, Feb. 1992.

[2] ITU-T, *Specification of Signaling System No. 7*, Recommendation Q.700 - Q.716, 1988

[3] GSM System Recommendation 3.05 ver.3.2, *Technical Performance Objectives*, Oct. 1991.

[4] GSM System Recommendation 9.02 ver.3.8, *Mobile Application Part Specification*, Jan. 1991.

[5] ITU-T Recommendations Q.1000 - Q.1032, *Public Land Mobile Network Interworking with ISDN and PSTN*, Nov. 1988

[6] ITU-T Recommendations Q.1051 - Q.1063, *Public Land Mobile Network and Mobile Application Part and Interfaces*, Nov. 1988

[7] TIA/EIA IS-41 Rev., *B Cellular Intersystem Operations*, Jul. 1991.

[8] TIA/EIA IS-41 Rev., *C Cellular Intersystem Operations Baseline Text*, June 1994

[9] TIA/EIA TSB51, *Cellular Radiotelecommunications Intersystem Operations: Authentication, Signaling Message Encryption and Voice Privacy*, May 1993

[10] ETRI, *MX/VLR - HLR/AC MAP Signaling Specification*, Jan. 1995.

[11] Sang Sik Lim and Sun Bae Lim, "HLR database architecture for CMS," *ITC- CSCC* '96, vol.2, July 1996.

[12] Hector Garcia-Molina, "Main memory database systems: An Overview," *IEEE Trans, on*

Knowledge and Data Engineering, vol. 4, no. 6, Dec. 1992.

[13] Tobin Jon Lehman, "Design and performance evaluation of a main memory relational database system," Ph.D. Thesis, Computer Science Department of University of Wisconsin, Madison, 1986.

[14] Per-Ake Larson, "Dynamic hash tables," *CACM*, vol. 31 no. 4, Apr. 1988.

[15] Tobin J. Lehman and Michael J. Carey, "A study of index structure for main memory database management systems," *Proc. of VLDB Conf.*, pp. 294-303, 1986.

[16] Ulf Korner, Christian Nyberg, "Overload control in communications networks," *GLOBECOM*, 1991.

[17] P. M. D. Turner and P. B. Key, "A new call gapping algorithm for network traffic management," *ITC-13*, Copenhagen, 1991.

[18] Chul Yi Moon, Hyun Gon Kim, and Jong Hyeon Lee, "Overload control of HLR using dynamic load limit," *ITC-CSCC* '96, July 1996.

Sun Bae Lim received a Ph. D. degree in electronics engineering at Korea University in 1993, M.S. degree in computer science at Korea Advanced Institute of Science and Technology in 1989, and B.S. degree in electronics engineering at Korea University in 1978. He joined ETRI in 1984. He worked for computer system development projects for six years as a senior engineer. Since 1990, he has been working for mobile communication system development projects. Currently, he is in charge of Mobility Management Section. His research interests include mobile network architecture, mobile network protocols, and mobile communication security mechanisms.

Kyeongsuk Shin received the B.S. degree in computer science at Hallym University in 1988, M.S. degree in computer engineering at Kyunghee University in 1991. She joined ETRI in 1991. She has been working for mobile communication system development projects. Now, she takes part in the IMT-2000 project. She is interested in mobile network protocols and mobile user registration.

Hyungon Kim received the B.S. and M.S. degrees in electronics engineering at Kumoh National University of Technology in 1992 and 1994, respectively. He joined ETRI in 1994. He has been working for mobile communication system development projects. Now, he takes part in the IMT-2000 system development. His interests include wavelet transform coding, mobile network architecture, and mobile network protocols.