

“A DISTRIBUTED DATABASE ARCHITECTURE FOR GLOBAL ROAMING IN NEXT-GENERATION MOBILE NETWORKS”

Ms.Tejasshri H. Mohite¹, Mr.Patil A. D.², Ms.Sapana Kishor Soudagar³, Ms.Pooja Adgonda Patil⁴
¹Asst.prof., Electronics dept., Dr. J.J.M.C.O.E., Jaysingpur, Dist -Kolhapur, Maharashtra ,
⁴Associate.prof., Mechanical Dept, Textile & Engineering Institute, Ichalkaranji, Dist. Kolhapur.,
Maharashtra., ³Asst.prof., Electronics dept., Dr. J.J.M.C.O.E., Jaysingpur, Dist -
Kolhapur, Maharashtra, ⁴Asst.prof., Electronics dept., Dr. J.J.M.C.O.E., Jaysingpur, Dist -
Kolhapur, Maharashtra

ABSTRACT: Next-generation wireless systems are envisioned to have an IP-based infrastructure with the support of heterogeneous access technologies. One of the research challenges for next generation all-IP-based wireless systems is the design of intelligent mobility management techniques that take advantage of IP-based technologies to achieve global roaming among various access technologies. Next-generation wireless systems call for the integration and interoperation of mobility management techniques in heterogeneous networks. In this article the current state of the art for mobility management in next generation all-IP-based wireless systems is presented. The previously proposed solutions based on different layers are reviewed, and their qualitative comparisons are given. A new wireless network architecture for mobility management is introduced, and related open research issues are discussed in detail.

Keywords - next-generation (NG) cellular networks, mobility management, Intersystem roaming.

I. INTRODUCTION

Currently, various wireless technologies and networks exist that capture different needs and requirements of mobile users. For high-data-rate local-area access, wireless LANs (WLANs) are satisfactory solutions. For wide-area communications, traditional and next-generation (NG) cellular networks may provide voice and data services. For worldwide coverage, satellite networks have been used extensively in military and commercial applications. Since different wireless networks are complementary to each other, their integration will empower mobile users to be connected to the system using the best available access network that suits their needs. The integration of different networks generates several research challenges because of the following heterogeneities:

Access technologies: NG wireless systems will include many heterogeneous networks using different radio technologies. These networks may have overlapping coverage areas and different cell sizes, ranging from a few square meters to hundreds of square kilometers.

Network architectures and protocols: NG wireless systems will have different network architectures and protocols for transport, routing, mobility management, and so on.

Service demands: Mobile users demand different services ranging from low-data-rate non real-time applications to high-speed real-time multimedia applications offered by various access networks.

The above intrinsic technology heterogeneities ask for a common infrastructure to interconnect multiple access networks. IP is recognized to become the core part of NG integrated wireless systems to support ubiquitous communications [1]. For interoperation of different communication protocols, an adaptive protocol suite is required that will adapt itself to the characteristics of the underlying networks and provide optimal performance across a variety of wireless environments [2]. Furthermore, adaptive terminals in conjunction with “smart” base stations will support multiple air interfaces and allow users to seamlessly switch among different access technologies. One important component of the adaptive protocol suite is the integration of mobility management schemes. In this article we address the design of intelligent mobility management techniques that take advantage of IP-based technologies to achieve global roaming among heterogeneous networks [1]. To make this roaming seamless, the integration and interoperation of heterogeneous mobility management techniques with efficient support for both intra- and interdomain roaming are required. Some proposed mobility management techniques try to support mobility associated with intradomain roaming [3–5]. However, these

solutions have high signaling load and long handoff delay for interdomain roaming. Therefore, we advocate new mobility management architectures for a heterogeneous environment that reduce both intra- and interdomain signaling load and handoff delay. The rest of this article is organized as follows. In the next section we describe the challenges for mobility management in NG all-IP-based wireless systems. We review the network layer mobility management solutions, and then investigate the link layer solutions. We present mobility management solutions based on both the network.

II. MOBILITY MANAGEMENT

Mobility management contains two components: location management and handoff management [6]. In NG wireless systems, there are two types of roaming for mobile terminals (MTs): intrasystem (intradomain) and intersystem (interdomain) roaming. Intrasystem roaming refers to moving between different cells of the same system. Intrasystem mobility management techniques are based on similar network interfaces and protocols. Intersystem roaming refers to moving between different backbones, protocols, technologies, or service providers. Based on intra- or intersystem roaming, the corresponding location management and handoff management can be further classified into intra- and intersystem location management and handoff management.

The proposed database system is a multitree structure (Fig. 1), consisting of a number of distributed database subsystems (DSs), each of which is a three-level tree structure. More than three levels may be adopted in a DS. However, adding more levels will introduce longer delays in location registration and call delivery. These DSs communicate with each other only through their root databases, DB0s, which are connected to the others by the public switched telephone network (PSTN), ATM networks, or other networks.

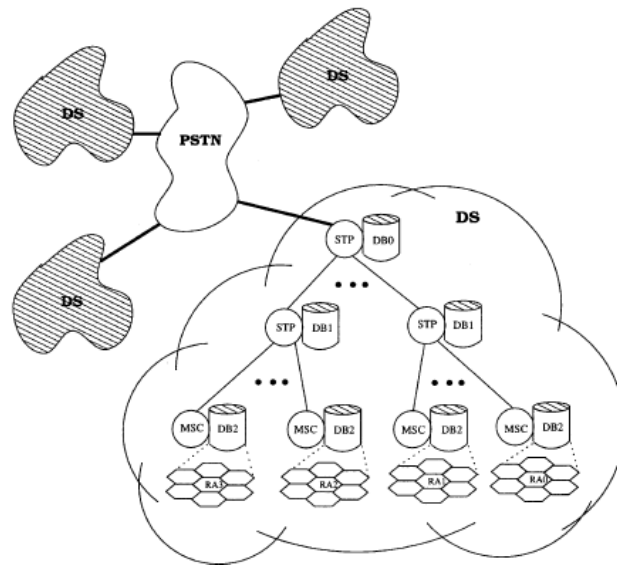


Fig. 1. Proposed multitree database architecture.

III. THE PROPOSED DATABASE

1) A location-independent PTN provides a basis for global roaming in the next-generation mobile networks where terminal mobility, personal mobility, and service provider portability will be implemented. A mobile subscriber can

retain its lifelong PTN regardless of its location and service provider.

2) The multitree database architecture is much more robust than the one-root hierarchical architecture. In the proposed architecture, an MT's profile is stored in one of the root databases according to its current location. Thus, each root database only maintains a small portion of the user profiles in the global mobile system. The crash of one root database will not disrupt the operation of other root databases, and the recovery of the failed root database is much easier than in the one-root database architecture where all user profiles need to be recovered once the root is crashed.

3) The multitree database architecture is scalable, which is crucial to support continuously increasing number of mobile subscribers in future mobile networks. When the capacity of a root database is saturated, a new DS is readily added. More importantly, the end-to-end delay in location registration and call delivery will not increase due to such an expansion in the mobile network. On the other hand, with the one-root structure, when the capacity of the root or a high-level database is saturated, more levels of databases need to be added in order to

reduce the burden on the root or high-level databases. This will increase the delays in location registration and call delivery.

4) The proposed multitree database system is easy to expand and maintain in the multioperator environment of a global mobile system. With the multitree architecture, each service provider can have its own DSs and it is straightforward for a service provider to expand its service coverage by adding new DSs. It is also easy to operate and manage a DS when the DS is wholly owned by a single service provider. The one-root architecture, however, may not have such advantages.

5) No GTT is required in the proposed database architecture, where a signaling message is only sent from a database

to another database in an adjacent level within the same subtree or from a DB0 to another DB0. Since a message sender always contains the address of the receiver in its database, no GTT is required. This greatly simplifies the implementation of the proposed architecture. In addition to the multitier location database architecture, this paper also proposes indexing schemes for each type of location databases and analyzes their efficiency and cost in terms of database access time and storage requirement.

The location registration and call delivery procedures based on the proposed database structure are also given. Analysis models are developed to study the service response time of each type of databases in the proposed multitier architecture as well as the end-to-end delays incurred by the proposed location registration and call delivery procedures. The proposed architecture is compared with the one-root architecture as well as the HLR-VLR architecture in terms of the signaling loads due to location registration and call delivery. Numerical results have demonstrated that the proposed database architecture outperforms the one-root architecture and the HLR-VLR architecture, and can effectively cope with the anticipated high access rates to various location databases in future mobile networks while meeting the end-to-end delay requirements for location registration and call delivery. The remainder of this paper is organized as follows. Section II describes the proposed distributed database architecture for location tracking as well as the indices of the location databases. Section III presents the database searching strategies associated with the location update and call delivery procedures. Section IV describes the analytic model for performance evaluation of the proposed database architecture. Numerical results are presented in Section V, and conclusions are given in Section VI.

IV. MULTITREE DATABASE ARCHITECTURE FOR LOCATION TRACKING

A. MULTITIER LOCATION DATABASE ARCHITECTURE

The proposed database architecture for location tracking is a multitier structure, where each subsystem is a three-level architecture (Fig. 1), referred to as a *database subsystem* (DS) in this paper. Various DSs may represent networks operated possibly by different service providers. All these DSs are interconnected together via a fixed network, such as PSTN or ATM network, and communicate with each other only through their root databases. This architecture can support a multioperator environment which is expected in future mobile networks. In each DS, databases DB0 and DB2 may correspond to the HLR and the VLR in the two-level database system, respectively. Each DB2 may control an RA where a user can roam freely without triggering registrations. Each DB2 is collocated with an MSC, which performs call processing on origination or termination calls.

B. TWO EFFICIENT DATABASE INDICES

A database usually consists of two parts: an *index file* and a *data file*. The index file contains an access structure called *index*, which provides search paths for locating the records in the data file. The index determines the database access time, thereby being the critical component for improving database throughput. Efficient indices should be based on application characteristics such as the types of storage devices available, the affordable storage capacity, the types of queries required, the available keys, etc.

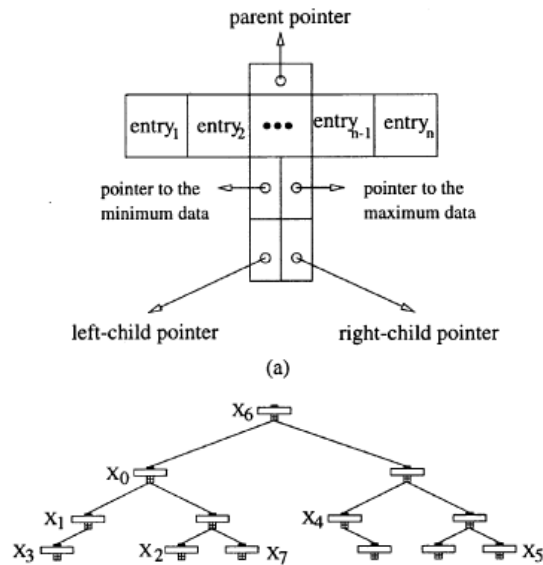


Fig. 2. (a) T-node. (b) T-tree.

In this paper, we focus on the indices suitable for a variety of databases in mobile systems. There are two classes of indices: the disk-oriented index, such as the B-tree, and the memory-resident index, such as the AVL-tree and the T-tree. While the disk-oriented indices are designed primarily to minimize the number of disk block accesses and to minimize disk space, the memory-resident indices aim to reduce computation time while using as little memory as possible. For real-time applications, the memory-resident indices are preferred due to their much faster access times than the disk-resident indices. The indices can also be classified into the following two categories: the order-preserving indices and the randomizing indices. The primary order-preserving indices include arrays, B-trees, AVL-trees, T-trees, and direct files. The randomizing indices include various hashing indices. Essentially, the direct file is a special form of hashing indices. We can call the direct file *perfect hashing* due to its collision-free property and use it in the DBOs due to its fast response time and easy implementation. The hashing indices have been applied in various computer and communications systems. For example, in [10], a hash function was used to balance the query load across multiple GTT servers by distributing users' PTN-to-HLR address mappings evenly among the GTT servers. In the peer-to-peer systems [19], [21], hash-based techniques were used to map file names to their locations in the peer-to-peer systems while balancing the query load amongst all nodes. The hardest task of applying hashing techniques is to design efficient hash functions that can minimize collisions while keeping memory usage low.

On the other hand, the order-preserving indices are much easier to implement and provide guaranteed upper bounds on the search time while keeping memory usage efficient. It has been shown in [12] that among the order-preserving indices—array, B-tree, AVL-tree, and T-tree, the T-tree provides the best overall performance for a mix of searches, inserts, and deletes at a relatively low storage cost. Inserts and deletes incurred by location update as well as searches required by call delivery in the DB1 and the DB2 make the T-tree suitable for these databases. On the contrary, the biggest drawback with the array is that data movement is for each update, thus the array seems only suitable for a read-only environment [12]. The AVL-tree has poor storage utilization since each node stores only one data item while requiring two pointers and some other control information. As mentioned earlier, we also suggest that the *memory-resident direct file* be used as the index for large databases such as DBO, etc., due to its much faster speed than the other order-preserving indices.

1) T-Tree: The T-tree, which evolved from the AVL-tree and the B-tree, is a binary tree in which each node called *T-node* contains a number of data items, a parent pointer, a left-child pointer, a right-child pointer, and some other control information (Fig. 2). The T-tree is fast since it retains the intrinsic binary search nature of the AVL-tree. On the other hand, unlike the AVL-tree that holds only one data item in each node, the T-tree contains a number of data items in each node similar to the B-tree, thus having good storage utilization. In a T-node, the data items are arranged in increasing order of their keys. To find a value in the T-tree, a search algorithm for the T-tree is needed. According to [12], one efficient *search algorithm* for the T-tree can be described as follows: 1) each search begins with the root node; if the search value is less than the minimum value of the node, then the left-child node is searched. Otherwise, the current node is marked for future consideration and the search goes down the subtree pointed to by the right-child pointer. When the search reaches a leaf, the last marked node is searched using a binary search. The search fails when the search value is

not found in the marked node that bounds the search value (this node is called the *boundingnode*) or when the bounding node does not exist in the T-tree. Refer to [12] for details about the T-tree.

2) **Direct File:** In the direct file, there is a direct relationship between the record key and its storage location. The fastest searching method to access a direct file is *direct addressing* [2]. The key value is used as a relative record number that can be translated into a hardware address by the system. When the direct file is memory resident, the hardware address is the memory address. One potential disadvantage of direct addressing is that space must be reserved for every possible key value, resulting in wasting large amounts of storage. However, when the number of possible key values is relatively close to the number of actual key values, direct addressing is very cost effective. Whenever access time is the vital criterion, even lower packing densities are acceptable. To use direct addressing, the key values must be numeric, in ascending order, and the records must have fixed length. The location-independent PTN numbering plan makes direct addressing quite suitable for large centralized databases in mobile networks.

V. ANALYTIC MODEL

In this section, the access rates to the location databases as well as the end-to-end delays in location registration and call delivery of the proposed database architecture are given. In addition, the access costs of the T-tree and the B-tree are evaluated while formulas for the response times of the direct file, T-tree, and B-tree as well as their storage capacity requirements are presented. Finally, the signaling load of location registration and call delivery is evaluated for the multitree architecture, the one-root architecture, and the HLR-VLR architecture. *A. Access Rates to Location Databases* Each database can be modeled as a server with a buffering queue. The database provides service to queries using the first-come-first-served (FCFS) principle. For simplicity, the wait queue is usually assumed to be infinite. Due to the large number of users in future mobile networks, it has been shown in [18] that the arrival traffic to a database can be approximated by a Poisson process. We also assume that the service time of a database follows a general distribution. Thus, each database can be modeled as a queue. The whole database system forms a queueing network. The *service response time* of database, denoted by t_s , which consists of the waiting time spent in the queue and the service time, is given by [11]

VI. NUMERICAL PERFORMANCE RESULTS

A. MOBILITY MODEL

In order to estimate the end-to-end delays due to location update and call delivery, the average location update rate generated by an RA, λ_u , and the call rate originating from an RA, λ_c , should be determined first. To accomplish this, the simple uniform fluid flow model is used [17]. If v_1 of the users has speed v_1 , of them has speed v_2 , the rest of them does not move, and the direction of movement is uniformly distributed over 360° , the average arrival rate of location updates generated in an RA is [18]

$$\lambda_u = \frac{\rho L(v_1 r_1 + v_2 r_2)}{3600\pi}$$

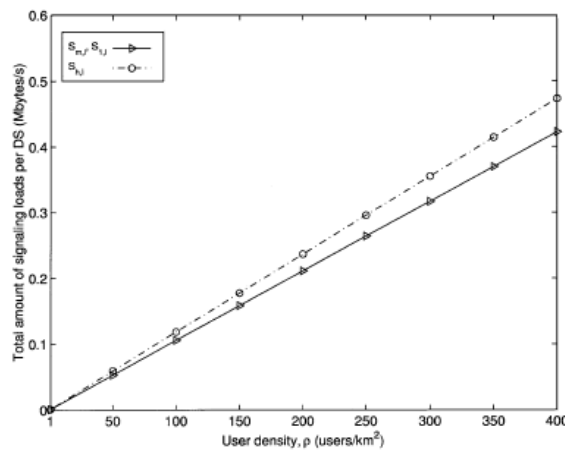


Fig. 2. Comparison of signaling loads on local links within a DS.

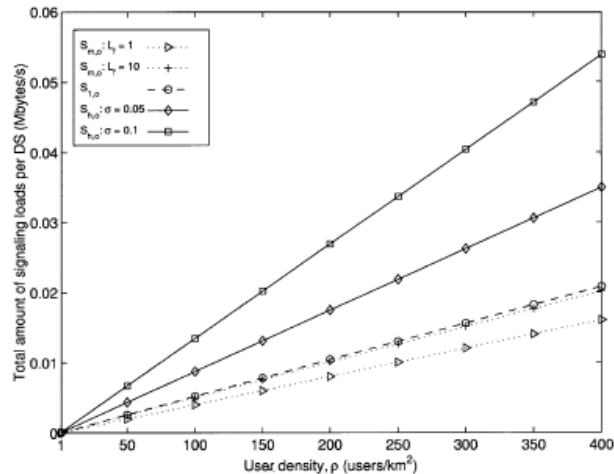


Fig. 3. Comparison of signaling loads on international links between DSs.

VII. CONCLUSION

Distributed multitier database architecture has been proposed for location management in a global mobile system, where the location-independent PTNs are employed to support seamless global roaming. To support the anticipated large number of mobile users in the future mobile system, two efficient database access structures—the memory-resident direct file and the T-tree—were proposed to achieve high database throughput, so that the end-to-end delays in location registration and call delivery can meet the delay requirements in mobile networks. The proposed database architecture is scalable, robust, and efficient. Compared to the existing two-level location database architecture, the proposed database architecture can support a much higher user density while reducing signaling load significantly. Compared to the one-root tree architecture, the proposed architecture provides better scalability and reliability while supporting a larger user population at a lower signaling cost. For performance evaluation, analysis model was developed. Numerical results have revealed that the proposed database architecture can effectively handle the anticipated high update and query rates to the location databases in future mobile networks. The proposed database access structures are also suitable for other large centralized databases in mobile networks, such as the authentication center and the equipment identity register.

REFERENCES

- [1] I. F. Akyildiz, J. Mcnair, J. S. M. Ho, H. Uzunalioglu, and W. Wang, “Mobility management in next-generation wireless systems,” *Proc. IEEE*, vol. 87, pp. 1347–1384, Aug. 1999.
- [2] B. G. Claybrook, *File Management Techniques*. New York: Wiley, 1983.
- [3] I.-R. Chen, T.-M. Chen, and C. Lee, “Agent-based forwarding strategies for reducing location management cost in mobile networks,” *ACM/Baltzer J. Mobile Netw. Applicat.*, vol. 6, no. 2, pp. 105–115, 2001.
- [4] S. Dolev, D. K. Pradhan, and J. L. Welch, “Modified tree structure for location management in mobile environments,” *Comput. Commun.*, vol. 19, no. 4, pp. 335–345, 1996.
- [5] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 2nd ed. Menlo Park, CA: Addison-Wesley, 1994.
- [6] J. S. M. Ho and I. F. Akyildiz, “Dynamic hierarchical database architecture for location management in PCS networks,” *IEEE/ACM Trans. Networking*, vol. 5, pp. 646–660, Oct. 1997.
- [7] , “Local anchor scheme for reducing signaling costs in personal communications networks,” *IEEE/ACM Trans. Networking*, vol. 4, pp. 709–725, Oct. 1996.
- [8] R. Jain and Y.-B. Lin, “An auxiliary user location strategy employing forwarding pointers to reduce network impacts of PCS,” *ACM-Baltzer J. Wireless Netw.*, vol. 1, no. 2, pp. 197–210, July 1995.
- [9] R. Jain, Y.-B. Lin, C. Lo, and S. Mohan, “A caching strategy to reduce network impacts of PCS,” *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1434–1444, Oct. 1994.
- [10] R. Jain, S. Rajagopalan, and L. F. Chang, “Phone number portability for PCS systems with ATM backbones using distributed dynamic hashing,” *IEEE J. Select. Areas Commun.*, vol. 15, pp. 96–105, Jan. 1997.
- [11] L. Kleinrock, *Queueing Systems: Vol. I—Theory*. New York: Wiley, 1976.
- [12] T. J. Lehman and M. J. Carey, “A study of index structures for main memory database management systems,” in *Proc. 12th Int. Conf. Very Large Data Bases*, Aug. 1986, pp. 294–303.
- [13] Y.-B. Lin and I. Chlamtac, *Wireless and Mobile Network Architecture*. New York: Wiley, 2001.

- [14] C. N. Lo and R. S. Wolff, “Estimated network database transaction volume to support wireless personal data communications application,” in *Proc. IEEE Int. Conf. Communications*, May 1993, pp. 1257–1263.
- [15] A. D. Malyan, L. J. Ng, C. M. Leung, and R. W. Donaldson, “Network architecture and signaling for wireless personal communications,” *IEEE J. Select. Areas Commun.*, vol. 11, pp. 830–840, Aug. 1993.
- [16] Z. Mao, “Location management strategies for personal communications services networks,” Ph.D. dissertation, Dept. Electr. Comput. Eng., Univ. Miami, Miami, FL, 2000.
- [17] S. Mohan and R. Jain, “Two user location strategies for personal communications services,” *IEEE Pers. Commun.*, pp. 42–50, First Quarter 1994.
- [18] X. Qiu and V. O. K. Li, “Performance analysis of PCS mobility management database system,” in *Proc. IEEE IC3N*, Sept. 1995, pp. 434–441.
- [19] S. Ratnasamy, P. Francis, M. Handley, and R. Karp, “A scalable content-addressable network,” in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 161–172.
- [20] N. Shivakumar, J. Jannink, and J. Widom, “Per-user profile replication in mobile environments: Algorithms, analysis, and simulation results,” *ACM/Baltzer J. Mobile Netw. Applicat.*, vol. 2, no. 2, pp. 129–140, Oct. 1997.
- [21] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications,” in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 149–160.