Architecture of SQL Databases for WLAN Access Control and Accounting

Josip Lorincz, Goran Udovičić* and Dinko Begušić
FESB-Split, University of Split, Croatia
KRON d.o.o. Split, Croatia*
josip.lerinc@fesb.hr, goran.udovicic@kron.hr*, dinko.begisic@fesb.hr

Abstract: **In recent years, there has been a growing demand for the development of authentication, authorization and accounting (AAA) systems for WLAN networks. In this paper we present a network architecture of SQL database servers especially tailored for AAA functions in large WLAN systems. Proposed database network architecture is vendor independent in sense that can be deployed with different AAA server implementations. WLAN management information database for AAA purposes is center of a WLAN network management system providing control of parameters such as: authentication and authorization of users, deletion or creation of users information, storage of accounting information which can be used for billing. Detailed database structure of each SQL database server in proposed centralized-distributed AAA database network architecture is described. Working principles of network database system for WLAN AAA purposes are based on MySQL master/slave chain replication. Proposed network architecture of SQL databases offers high level of reliability, availability and scalability.**

Keywords- AAA server, RADIUS server, MySQL, IEEE 802.1x, WLAN, SQL database, authentication, accounting

1. INTRODUCTION

Today, IEEE 802.11 Wireless LAN(WLAN) network is widely deployed and used as an emerging technology to connect high-speed Internet in the public wireless environment. Wireless networking presents an issue based on the fact that there is no physical method to restrict access to the system within radio range of a wireless network. Thus, WLAN networks primarily need an Authentication, but also Authorization and Accounting (AAA) mechanism, especially for usage in the public environment. Information's dedicated to AAA mechanism are mostly placed in centralized and distributed AAA databases for WLAN access control, accounting and monitoring [3].

In networking, a database is a structured collection of records or data that is stored in a computer (server) so that a computer program can consult it to answer queries. The records retrieved while answering to queries are information that can be used to make decisions. Databases play important role in contemporary network systems. Thus, it is crucial that structure and dimensions of databases be carefully planned and designed according to database purpose. With introduction to growing number of users and services that are offered in today's wireless networks, databases for WLAN AAA that offer different services become more and more complex [14]. It is also relevant to take into consideration increasing demand for integration of PLMN and WLAN networks and different mobile operator WLAN networks imposing even more challenges to WLAN AAA databases [12,13,5]. Technical background and advantages of deploying a centralized user database as a general repository for network and management applications is shown in [11]. A new accounting configuration management for wireless and mobile communications systems is proposed in [10].

Of critical importance in establishing operational and stable environment for AAA database services in WLAN will be the secure access to AAA database(s) over combination of wireless and wired network infrastructure. To accomplish this task, IEEE 802.1x has been increasingly implemented in many new 802.11 WLAN devices [1]. The IEEE 802.1x standard defines a mechanism for port-based network access control to provide compatible authentication and authorization mechanisms for devices interconnected by various 802 LANs [2,6]. IEEE 802.11i incorporates 802.1x as authentication solution for WLAN. There are three main components in the 802.1x authentication system: *supplicant*, *authenticator*, and *authentication server* (AS). A supplicant is usually a mobile node (MN) requesting WLAN access. An authenticator represents the network access server (NAS). In 802.11 networks it is normally an access point (AP). A Remote Authentication Dial In User Service (RADIUS) [7] server is commonly used as the authentication server, although other types of AAA servers such as DIAMETER [9] could also serve as the authentication server.

Although the proposed database structure is essentially independent of type of AAA server, in the rest of this paper the term AAA server will be related to RADIUS server as type of AAA server mostly used to control network access in contemporary WLANs. Most of AAA protocols allow users to maintain user profiles in a central WLAN AAA database that potential remote servers can share. Having a central WLAN database also means that it's easier to track usage for billing and for keeping network statistics. Thus database used for AAA purposes in WLAN represent central repository of access control, accounting and user management information offering many statistical parameters about user and their logins in the network. This paper discusses design considerations that must be taken into account during development of Structured Query Language (SQL) databases for WLAN AAA purposes.

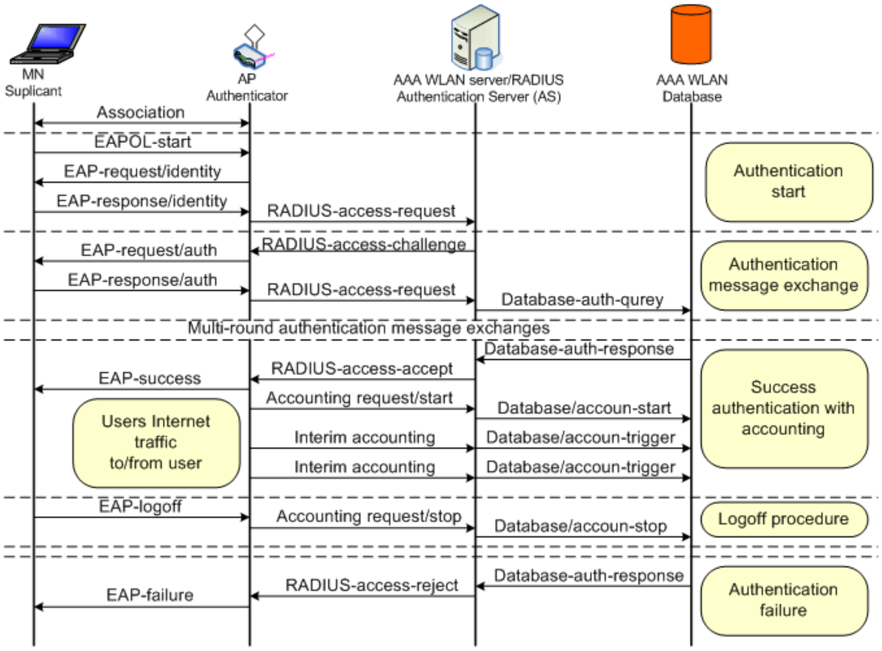

Figure 1. Example flows of IEEE 802.1x message exchange with AAA database

Distributed SQL database network architecture with capabilities of centralized storage of AAA information for Mobile Network Operators (MNO) offering WLAN services are proposed.

The rest of the paper is organized as follows: in section II. description of IEEE 802.1x framework in relation to database used for AAA operations is shown. Section III. describes proposed SQL database network architecture for AAA services in large WLANs using principles of MySQL master/slave database replication. Finally, some concluding remarks are given in section IV.

2. THE IEEE 802.1x FRAMEWORK

802.1x is a port-based protocol. In 802.1x, the *port* represents the association between MNs (supplicant) and APs (authenticator). This port restricts access to the Internet and network resources until the MN provides credentials through authentication. To control access to a network, the AP uses "controlled" and "uncontrolled" ports. Both of them are logical and virtual, but they use a single wireless association (link) between the supplicant and the AP called port access entity (PAE). The AP PAE controls the authorized/unauthorized state of its *controlled port* depending on the outcome of the authentication processes. Before the MN is authenticated, the AP uses an *uncontrolled port* to communicate with the supplicant. The AP will block all traffic except 802.1x messages before the supplicant is authenticated. The 802.1x standard leverages Extensible Authentication Protocol (EAP, IETF RFC 2284) to provide a number of authentication schemes, including MD5 *(IETF RFC 1321)*, TLS *(IETF RFC 2716)*, TTLS, PEAP, and smart cards such as EAP SIMs [6]. 802.1x also defines EAP over LANs (EAPOL) that encapsulates EAP messages between the supplicant (MN) and authenticator (AP). EAP messages from the MN are relayed over AP to the AAA server such as RADIUS or DIAMETER server. In order to let the RADIUS server authenticate users using EAP, the AP encapsulates the same EAP messages in RADIUS packet format and sends them to the RADIUS server (assuming it has been adopted as the authentication server). The encapsulation is known as RADIUS-encapsulated EAP with the EAP-Message attribute, which is defined in RADIUS Extensions (IETF RFC 2869) for supporting EAP within RADIUS. Once the MN is authenticated successfully, the controlled port in the AP is authorized. Packets from the supplicant will now go through the controlled port of the AP to backend networks to acquire the necessary services.

Fig. 1. depicts a typical 802.1x message exchange. After the MN and AP complete the 802.11 association, the port is unauthorized, and the 802.1x authentication process just starts. As indicated on Fig. 1., the MN sends an EAPOL-Start frame to the AP to initialize the authentication process. When the AP receives EAPOL-Start, it replies with an EAP-Request/Identity to obtain the MN's identity. The MN then sends back an EAP-response/identity containing the MN's identity in response to the EAP-request/identity. If the AP receives the EAP-response/identity, the authenticator PAE encapsulates the EAP-response/identity message in RADIUS-access-request as an attribute (EAP-Message attribute) and sends it to the RADIUS server. In response to the RADIUS-access-request, the RADIUS server will challenge the MN by sending a RADIUS-access-challenge to the AP, which then relays the message in the form of EAP-request/auth to the MN. After the MN receives EAP-request/auth, MN replies with an EAP-response/auth, which is also relayed to the RADIUS server by the AP in the format of RADIUS-access-request. Depending on the authentication scheme, there might be some more message exchanges.

The RADIUS server sends query to AAA database in order to compare the user supplied authentication data with the user-associated data stored in WLAN AAA database. If the credentials match, the user is granted network access, otherwise access to the network is denied. Described system relays on authentication that is performed in (by) AAA WLAN database. Based on results of database query, decision whether the MN should be accepted or denied access to the network services will be processed by RADIUS server.

Figure 1. depicts three cases thereafter that are separated by dotted lines. If authentication succeeds, the RADIUS server sends a RADIUS-access-accept to the AP. On receipt of RADIUS-access-accept the authenticator sends an EAP-success message to the MN to indicate the success of authentication. The controlled port of the AP is thus authorized. After receiving EAP-success, the MN is authenticated to network and the whole authentication process is completed. On the other hand, RADIUS-access-reject is sent by the RADIUS server and relayed to the MN by the AP in the message of EAP-failure if the authentication fails. In this case, the MN is not authenticated by RADIUS server, and the whole authentication fails. The controlled port is thus still unauthorized. If the MN is authenticated and wants to perform a logoff procedure from the current AP, the MN originates an EAPOL-logoff packet to the AP. After that, the controlled port of the current AP transits to unauthorized state immediately.

In addition to authentication and authorization, RADIUS server provides a technique for collecting accounting information specific to the end user's communication session and storing it on an accounting (AAA) server. If the AP is configured for RADIUS accounting, it forwards Accounting-request (with Acc-status-type set to Start) message to the accounting AAA server with completion of the authentication procedure. When Accounting-Request reach the RADIUS server, the server will then send an SQL query. This will create an entry in the accounting database. Before sending an acknowledgment back, another SQL query is sent to get the unique session ID created by the RADIUS server. The RADIUS server acknowledges the Accounting-request by sending Accounting-replay where only attribute present is Acc-Session-ID. After that, database starts collecting information about the session, including: input and output octets, input and output packets, session duration, and termination cause (such as user request or idle timeout). When the session is terminated, the NAS sends an Accounting-Request (with Acc-status-type set to Stop) message to the server indicating that the session is over. RADIUS accounting is described in RFC 2866 [8]. Accounting information's are stored in accounting database. This information's can be used for billing and charging purposes depending of ISP preferences.

3. SQL DATABASE NETWORK ARCHITECTURE FOR AAA SERVICES IN LARGE WLAN'S

The database management proposed is this article is built on a centralized-distributed database architecture that defines required network components as well as processes to run a distributed AAA database management between multiple MNO. In the considered scenario several mobile networks, representing a generic platform for the provisioning of IP based services, are assumed. The network components of the architecture include (Fig. 2.): distributed AAA servers with corresponding Slave Databases (SDB) on the some machine, Network Access Servers (NAS) in form of APs, centralized Master Database (MDB) server and Backup Master Database (BMDB) server.

Database administration and maintenance is based on replication capabilities between master database and slave database. Last version of MySQL (v5.0) [4] support for one-way replication, in which one server acts as the master, while one or more other servers act as slaves. Replication offers benefits for robustness, speed, and system administration:

- Robustness is increased with a master/slave setup. In the event of problems with the master, it is possible to switch to the slave as a backup.
- Better response time for clients can be achieved by splitting the load for processing client queries between the master and slave servers. Thus queries may be sent to the slave to reduce the query processing load of the master.
- All modifications (updates, deletes, etc.) should be done on master database and using replication process sent to slave. databases simplifying database maintenance and administration.
- Another benefit of using replication is possibility of performing database backups using a slave server without disturbing the master. The master continues to process updates while the backup is being made.

The generic master database (MDB_A) shown in Figure 2. is divided into two parts. On one hand, the Administration Data Storage (ADS) contains information required for business processes of an operator.

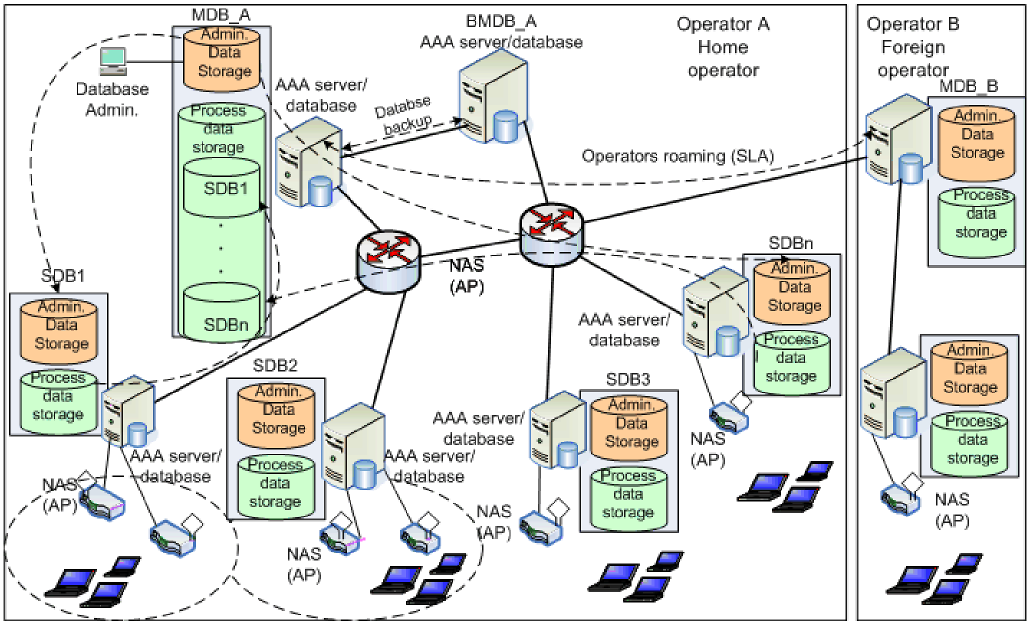

Figure. 2. Proposed network architecture of AAA database servers for large WLAN networks

Information stored in this database serves as input to the configuration of network elements and services. It contains user profiles and Service Level Agreements (SLA), and configuration data required for accounting and charging. On the other hand, the Process Data Storage (PDS) serves as an output database for the accounting and charging processes. It stores accounting and charging records generated by SDBs servers.

3.1 Master/slave replication of MySQL databases

The master/slave database server configuration is set up as chained replication: MDB_A server is configured as master for ADS and as slave for PDS data. Other SDB servers are configured as slaves for ADS and at the same time as masters for their PDS data. Thus, for different parts of generic WLAN AAA database, master/slave replication in both directions between MDB_A and SDBx must be implemented (Fig. 3.).

The MDB_A server keeps track of all changes to ADS (updates, deletes, inserts, etc.) in its binary logs. Also, SDBx server keeps track of all changes to PDS (accounting and billing information's, etc.) in its binary logs.

In master/slave replication, the master server writes updates to its binary log files and maintains an index of those files to keep track of log rotation. Therefore, binary log files serve as a record of updates to be sent to any slave servers. When a slave connects to its master, it informs the master of the position up to which the slave read the logs at its last successful update. The slave receives any updates that have taken place since that time, so that the slave can execute the same updates on its copy of the data. If the master fails, or the slave loses connectivity with its master server, the slave keeps trying to connect periodically until it is able to resume listening for updates. Each slave keeps track of where it left off when it last read from its master server. The master has no knowledge of how many slaves it has or which ones are up to date at any given time.

MySQL replication capabilities are implemented using three threads (one on the master server side and two on the slave server side) shown on Fig. 3. The slave creates an I/O thread, which connects to the master and asks it to send the updates recorded in its binary logs. The master creates a thread to send the binary log contents to the slave. The slave I/O thread reads the updates that the master thread sends and copies them to local files, known as relay logs, in the slave's data directory. The third thread is the SQL thread, which the slave creates to read the relay logs and to execute the updates they contain. As previously mentioned, there are three threads per master/slave connection in each direction. A master that has multiple slaves creates one thread for each currently connected slave, and each slave has its own I/O and SQL threads (Fig. 3.). The slave uses two threads so that reading updates from the master and executing them can be separated into two independent tasks. Thus, the task of reading statements is not slowed down if statement execution is slow. For example, if the slave server has not been running for a while, its I/O thread can quickly fetch all the binary log contents from the master when the slave starts, even if the SQL thread lags far behind.

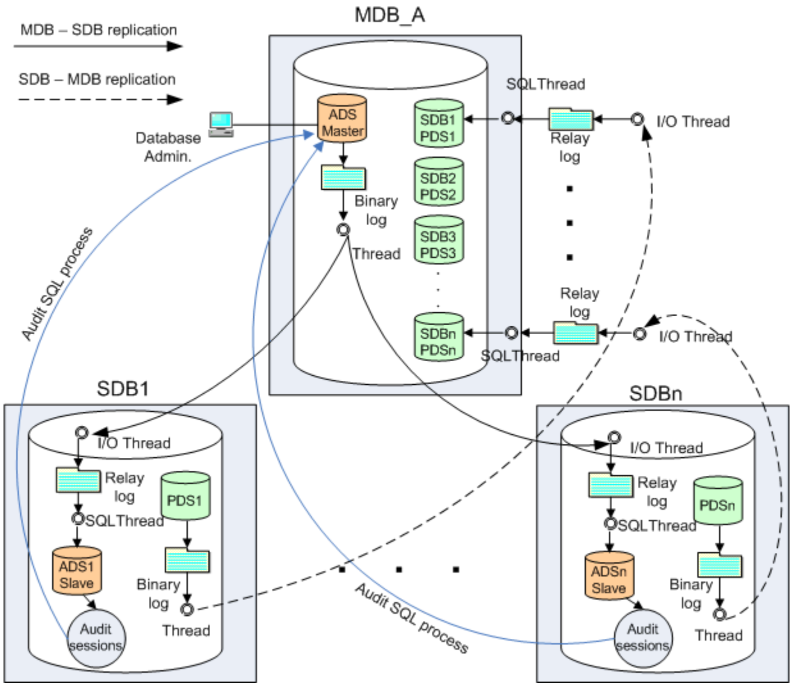

Figure 3. MySQL master/slave database replication process

If the slave stops before the SQL thread has executed all the fetched statements, the I/O thread has at least fetched everything so that a safe copy of the statements is stored locally in the slave's relay logs, ready for execution the next time that the slave starts. This enables the master server to purge its binary logs sooner because it no longer needs to wait for the slave to fetch their contents.

Relay logs have the same format as binary logs. The SQL thread automatically deletes each relay log file as soon as it has executed all events in the file and no longer needs it. Since we set up multiple slaves, each one must have a unique server-id value that differs from that of the master and from each of the other slaves. We can define server-id values as something similar to IP addresses where these IDs uniquely identify each server instance in the community of replication partners. For PDS data master/slave roles are reversed (i.e. the master MDB_A server for ADS data is now acting as slave server for PDS data). Therefore we have chained database replication MDB_A -> SDB1 -> MDB_A.

Normally, a slave does not log to its own binary log any updates that are received from a master server. It is important to emphasize that ADS database on master MDB_A server is read/write (R/W) type of database while ADS databases on SDBx servers are read-only (R-O) type of database. On the other hand, in reverse direction, PDS databases on SDBx servers are R/W type of databases while PDSn databases on MDB_A server are R-O databases.

3.2 Centralized-distributed AAA SQL database

With proposed centralized-distributed database architecture all AAA database information are stored on central location which is in our case master database server MDB_A (Fig. 2 and 3). Also, every AAA information that cross specific SDBx server are stored (distributed) on corresponding server. In order to further enhance robustness of system and data storage redundancy, third location for storage of AAA database information of complete system is backup master (BMDB_A) server. All AAA information stored in MDB_A server are also stored on backup BMDB_A server by means of backup procedure supported in MySQL server processes. This enables resistance of the complete AAA database system in situations of master server failure or master server upgrades.

The generic centralized MDB_A contains only one database for administrative purpose, called ADS, and *n* PDS databases which represent replication from *n* RADIUS server AAA slave databases (from SDB1-PDS to SDBn-PDS) as shown on Fig. 3. The master MDB_A server running one way replication will replicate all changes in master ADS database to the other RADIUS server slave ADS databases (from ADS1 to ADSn) as described previously. Only ADS data will be replicated to slave servers in this direction. Also, the slave SDBx server runs one-way replication in opposite direction, replicating all changes in slave PDS databases to corresponding PDS databases of master MDB_A server. In this replication process only PDS data from different SDBs will be replicated to corresponding PDS database of master server. Each AAA server (RADIUS) with slave database will run a local instance of ADS database for authorization and authentication and it will also run PDS database for accounting. Since every slave ADS database is R-O database, user authentication and service authorization functionalities will be performed in relation to this database.

Thus, for users placed in coverage area of APs which use AAA services from corresponding RADIUS server with AAA slave database, all authentication and authorization functionalities will be performed by means of slave ADS databases. In large WLAN networks this procedure unloads ADS database of MDB_A server enabling better response time for wireless clients. On the other hand, every slave PDS database is R/W type of database and all accounting processes will be performed in relation to this database. Thus, all accounting information generated by users (activity) placed in coverage area of APs served by related AAA server with slave database, will be placed in PDS database. Number of SDBs depends of number of AAA servers, because accounting information will be stored locally on each AAA server slave database and then forwarded to corresponding PDSn database of the MDB_A using one way replication (SDB - MDB). This will enable storage of all accounting information's from every WLAN user on centralized location, independently of its current position and movements in the WLAN network. All data from master server will be saved on special backup server using backup procedures supported in MySQL.

4. CONCLUSION

IEEE 802.11x has become an indispensable technology used in public WLAN environment to provide secure user access offered through AAA service. In this paper we propose new network architecture of database servers with emphasis on AAA database server allocation in large WLANs administrated by MNO. Proposed AAA database server network is type of centralized-distributed network architecture offering high level of robustness, reliability, availability and scalability. Centralized database network architecture means that all large WLAN system AAA database information will be stored on central database server called generic master database server (MDB). All operations associated with database administration will be accomplished on single central location simplifying database maintenance and administration. Backup of this database is performed on generic backup database server providing protection of the overall AAA database system in situations of MDB server upgrades or failure. Distributed database network architecture means assignment of AAA database processing loads on distributed database servers called slave database servers (SDB) located at network edges. Distribution of database processing loads reduces processing load of central MDB server offering better response time of AAA database services to the wireless clients. Novel centralized-distributed network structure of database servers for WLAN AAAA functionalities is realized using one-way replication process. By usage of replication process supported in MySQL, we accomplished to replicate part of database between MDS and SDSs creating chain database replication. Proposed database architecture is created in order to provide efficient AAA service in large WLAN networks using SQL data source and it can also be implemented for billing purposes.

REFERENCES

[1] IEEE Std. 802.1X-2001, "Port-Based network Access Control", June 2001., www.ieee.org

[2] Jyh-Cheng Chen, Ming-Chia Jiang, Yi-Wen Liu, "Wireless LAN security and IEEE 802.11i", *IEEE Wireless Communications,* vol. 12, no. 1, Feb 2005 pp. 27-36

[3] C. de Laat, G. Gross, L. Gommans, "Generic AAA Architecture," IETF RFC 2903, August 2000.

[4] MySQL 5.0 Reference Manual, Copyright 1997-2007 MySQL AB, http://dev.mysql.com/doc/refman/5.0/en/replication.html

[5] Henry Haverinen, Jouni Mikkonen, Timo Takamäki, "Cellular access control and charging for Nmobile operator wireless local area networks", IEEE Wireless Communications, Dec. 2002, pp.52-60

[6] Jyh-Cheng Chen, Yu-Ping Wang, "Extensible Authentication Protocol (EAP) and IEEE 802.1x: Tutorial and Empirical Experience", IEEE Radio Communications, December 2005, pp.26-32

[7] C. Rigney, S. Willens, A. Rubens, W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," IETF RFC 2865, June 2000.

[8] C. Rigney, "RADIUS Accounting," IETF RFC 2866, June 2000.

[9] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko "Diameter Base Protocol," IETF RFC 3588, September 2003.

[10] Frank Eyermann, Peter Racz, Burkhard Stiller, Christian Schaefer, Thomas Walter, "Diameter-based accounting management for wireless services", WCNC 2006, vol. 7, no. 1, April 2006 pp. 2304-2310

[11] W. Haidegger, "X.500 Type Databases For Flexible and Highly Available Common and Logically Centralized User Data Storage in Telecommunication" (ICIMP2007), July 2007

[12] Toni Janevski, Aleksandar Tudzarov, Meri Janevska, Perivoje Stojanovski, Dusko Temkov, Goce Stojanov, Dusko Kantardziev, Mine Pavlovski, Tome Bogdanov, "Adaptive solution of a billing system for internetworking of mobile networks and wireless LAN", SoftCOM 2005, pp.215-219

[13] Juha Ala-Laurila, Jouni Mikkonen, and Jyri Rinnemaa, "Wireless LAN Access Network Architecture for Mobile Operators", IEEE Communications Magazine, November 2001, pp.82-89

[14] Weider D. Yu, Sunita Sharama, "A mobile database design methodology for mobile software solutions", COMSAC 2007.