DISCOVERING ARCHITECTURE FORMALISM OF GEO-LOCATED WEB SERVICES FOR NEXT GENERATION OF MOBILE NETWORKS

André Claude Bayomock Linwa [1] and Samuel Pierre [2]
[1,2] LARIM, École Polytechnique, C.P. 6079, succ. Centre-ville Montréal (Québec), H3C 3A7, Canada

Abstract: Geo-located web services are web services offered in a particular geographical region. In mobile application design, a geo-located web service can be mapped to a set of mobile network location areas. As a mobile client roams in a mobile network, if he has a geo-located web service in execution progress at a supplier application server (SAS), he will lost its session in case when its current location is not covered by this SAS. With the next generation (third and up) of mobile networks, the geographical position of a mobile client will be sent back by a LoCation Server (LCS) to an application which requests it. As many geo-located web services will be deployed in the future, the great challenge for a mobile client will be to discover and maintain a geo-located web service when he is roaming. We propose a new system named Geo-Located Web Service Architecture (GLWSA) that aims to discover and maintain a geo-located web service with or without QoS at the nearest SAS of a mobile client current location. The GLWSA is a set of discover servers named GLWSMs (Geo-Located Web Service Manager) which are distributed in the topology. The GLWSA extends the UDDI and MLP protocols to add the GLWSM topology management and the thematic location of a mobile clients group, respectively. A thematic location consists of sending to a LCS, a chain of characters that represents a theme or a subject linking a group of mobile clients. In this paper, we present the GLWSA concepts and its mathematical formalism. Tests executed to evaluate the system performance prove that the GLWSA concepts are adequate to discover geo-located web services.

Key words: Geo-located services; Next generation mobile networks; Service discovery; Web services; Quality of services.

1. INTRODUCTION

In the development perspective of future mobile service applications, geo-located web services will be more and more deployed [5]. By definition, a geo-located web service is a web service offered in a specific geographical region. The geographical area restriction of a geo-located web service induces one main problem: how to maintain the service in execution progress when a mobile client leaves the geographical region covered by a geo-located web service? By adding to this problem the scalability deployment of geo-located web services, the great challenges for a mobile client that roams over a mobile network will be to discover and maintain a service in execution progress at the nearest supplier application server (SAS) of a mobile client current location. The service discovering and the service execution maintainability could be requested with or without QoS requirements.

An example of geo-located can be to do a virtual visit of a particular museum when the mobile client is in a specific geographical area.

In related work, the proposed architectures are either adapted to discover services where fixed clients are involved [6] and those which consider the mobility put the emphasis on code and agent mobility [7, 10] rather than data or task mobility to the nearest SAS (as the web services are service oriented and use a client/server model) based on the location context of the mobile client and maintainability of service execution.

To resolve these challenges, we proposed a new system named Geo-Located Web Services Architecture (GLWSA) that aims to lookup, publish a geo-located web service and coordinate the geo-located web service migration at the nearest SAS of a mobile client current location. The GLWSA also provides thematic location methods to locate a group of mobiles related by a theme or subject instead of locate a group mobiles by sending with a location request a list of mobile identifiers. The GLWSA is composed of a set of distributed Geo-Located Web Services Manager (GLWSM).

An example of thematic location is to determine the position of all *railroad trains of Via Rail Canada* which are in Montreal area. In this request, the "*railroad trains of Via Rail Canada*" is the subject or theme.

To discover a service with QoS (cost of service, network bandwidth and SAS utilization rate) and maintain a service execution with QoS, we defined a new mechanism that collects the network bandwidth of a particular geo-located web service and the SAS utilization rate at a specific GLWSM. For a specific service, the network bandwidth and SAS utilization rate are collected periodically in a particular GLWSM domain by sending a collect traffic message to all SAS that offer the concerned service in this domain. Each concerned SAS collects and sends back to the requestor GLWSM, the bandwidth and the SAS processor rate. Collected QoS parameters are used in the SAS selection and migration process.

The main contribution of this paper is to present the GLWSA concepts and its mathematical formalism. The mathematical formalism shows how the GLWSA properties, relations and functionalities (lookup, publish a geo-located web service and coordinate a geo-located web service migration at the nearest SAS) can be transform to algebra logics. Thematic location and the proposed mechanism to collect QoS are not in the scope of this paper.

The organization of this paper consists to present related work in discovering services, to explain the GLWSA concepts, to transform the system concepts in mathematical formalism, then evaluate the GLWSA system and give a brief conclusion.

2. RELATED WORK

Related work in the service discovery can be classified in two categories: classic and non classic protocols. Classic service discovery protocols are protocols which are commercially known and generally used in the service discovery process. SLP (Service Location Protocol) and Jini are the most popular classic service discovery protocols in literature. SLP (Service Location Protocol), a protocol developed by IETF, uses three agents: a UA (User Agent), an SA (Service Agent), and a DA (Directory Agent) [6]. On the other hand, Jini is a technology developed by Sun Microsystems for discovering services. Just as SLP, it involves three actors: the client, the service broker server, and the service supplier server [3]. However, SLP and Jini protocols are not designed for mobile clients and do not allow to discover a service based on a client's location context. Non classic discovery protocols are protocols proposed by research group to discover services but are not huge used. In [12], authors defined an architecture to locate mobiles and query databases based on their location context. The proposed architecture is a central middleware where services are published and discovered through a user service agent. The main component of the system is a location dependent service manager. The location dependent service manager controls the system. It analyzes the query and binds the pseudo-codes sent in the request to the correct predicates (e.g., 'nearest' can take the value 'five miles'). It verifies the granularity of the query, dispatches the request to the corresponding databases and returns the results to the client in the desired format.

In non classic discovering services protocols, we selected some relevant protocols.. In [7], an architecture named Application Module Request Broker (AMRB) is presented. This architecture enables clients to discover an application module (AM). The system is a distributed AMRB. Each AMRB has two main components: location and migration. Location and migration components allow to determine the current location of an application code and to migrate an application code to another host. The system proposes two kinds of migrations: host and code migration. The host migration uses a network location detection by sending periodic polling; code migration detects the migration of an AM and diffused the location change in a multicast process to all AMRB of the system.

In [11], an architecture for reconfiguration control and service provisioning platforms were proposed. This is a middleware system which mediates between a provider service called Value Added Service Provider (VASP) and the network resources in order to deliver services to end users according to their location context. The system informs the mobile clients of the communication cost as they change location area. Services are published and discovered through a service manager. Although this strategy seems similar to our approach, it fails to maintain a nearest service execution compared to the location context of the mobile client.

Other projects, such as Globe, use an architecture called "Globe Housing Service" [2]. This architecture allows locating the mobile users and services. Globe defines and implements distributed objects in a hierarchical topology tree. When a client looks up an object in a leaf node of a given location area, the object can be present or not. If it is absent, the system returns the contact address of the Globe object requested. This contact address redirects the request to the next node of the path tree. The procedure is repeated until the request is fulfilled. Globe is not adapted to discover services based on the location context of a mobile client.

3. THE PROPOSED SYSTEM CONCEPTS

3.1 UDDI protocol

The Universal Description, Discovery, and Integration (UDDI) protocol provides a standardized method for publishing and discovering information about web services [13]. Building a UDDI protocol is an industry initiative in order to create a platform-independent, open framework for describing services, discovering businesses, and integrating business services. The UDDI main objective is a web service discovery process in a service-oriented architecture. Basically, a UDDI protocol is associated with a single registry that can be distributed among many nodes. Each UDDI registry stores business information of a specific supplier or organization entity.

Conceptually, a business organization can register three types of information into a UDDI registry: white pages, yellow pages and green pages. The white pages are basic contact information and identifiers about a company, including business name, address, contact information, and unique identifiers. This information allows others companies or clients to discover a business organization web service based on the business identification. The yellow pages are information that describe a web service using different categorizations (taxonomies). These information allow others to discover a web service based upon its categorization (such as food business). The green pages are technical information that describe the behaviors and supported functions of a web service hosted by your business. These information include pointers to the grouping information of web services and where the web services are located [13].

The UDDI data structure is composed four elements: *businessEntity, businessService, bindingTemplate* and *tModel.* A *businessEntity* structure represents a business organization basic information. These information include contact information, categorization, identifiers, descriptions, and relationships to other businesses. A *businessEntity* contains one or more *businessService* structures. A *businessService* represents a description of one web service. This web service can contains one or many published methods. A *businessService* contains one or more *bindingTemplate* structures. A *bindingTemplate* contains pointers to technical descriptions and the access point URL, but does not contain the details of the web service's specifications. A *bindingTemplate* contains an optional text description of the web service, the URL of its access point, and a reference to one or more *tModel* structures. A *tModel* is an abstract description of a particular specification or behavior to which the web service adheres. A *tModel* is linked to a WSDL (Web Service Description Language) document that determines specifically how to interact with a particular web service. Clients or others organizations can use the information pointed a *tModel*'s WSDL document to determine whether a web service is compatible with their requirements.

3.2 MLP protocol

The MLP protocol is application level protocol that aims to allow the interoperability of location requests with location servers LCS [1]. The MLP format language is developed using XML language. The MLP architecture has three main layers: service, element and transport [9]. On the lowest level, the MLP transport layer defines how XML content is transported. Possible

MLP transport protocols include HTTP, SOAP and others. The Element layer (second layer) defines all common elements used by the services in the service layer. The Service layer (top layer) defines the services offered by the MLP. The services are classified into five categories: standard location immediate service, emergency location immediate service, standard location reporting service, emergency location reporting service and triggered location reporting service. The standard location immediate service allows applications to request a single location response from the Location Server LCS. The request can also be served by asynchronously sending the location to the application until a timeout limit is reached. The emergency location immediate service is used when a mobile client initiates an emergency call. This service is mostly used by *911* applications. In the standard location reporting service, the position is periodically sent to an application until a timeout is reached. The emergency location report is used when the mobile network automatically initiates the position determination for an emergency call. The position and related data are then sent back to the emergency application. The difference between the emergency location immediate service and emergency location report is that the first one is initiated by the subscriber, while the second service is automatically triggered by the provider. The triggered location reporting service is an event-based service where the mobile subscriber's location is reported on the occurrence of a specific event.

3.3 Description of the proposed system

The GLWSA system is a distributed system composed mainly of three entities: the GLWSM server, the UDDIM database and the ClientInfosDB database (Figure 1). The GLWSM (Geo-Located Web Services Manager) server is the main component of the system. It keeps the implementation of the system functionalities and manages the geo-located web service operations (lookup, publication, coordination of service migration, service execution, mobile location, etc.). The UDDIM (UDDI for Mobiles) is an extension of the UDDI to adapt the UDDI registry to the mobility context by adding APIs and structures of the GLWSA topology (GLWSM nodes), the service agreement of a geo-located web service and the geo-located web service agreement per node. The ClientInfosDB database stores user personal data such as identification (name, address, username, password, etc.), equipment (mobile station identifier, phone number), subscription and quality of service data. A GLWSM interacts with three external entities: mobile client, the SAS (Supplier Application Server) and the LCS server. A mobile client is a person who has a mobile equipment (cell phone, PDA, etc.) and has subscribed to a specific geo-located web service. The SAS is the end server where a geo-located web service is really implemented and executed. The LCS is the location server which determined the geographical position of a particular mobile client. To interact with the LCS, the GLWSM uses a MLPe protocol (MLP Extension) which is an extension of the MLP protocol to realize the thematic location of a group of mobile clients.

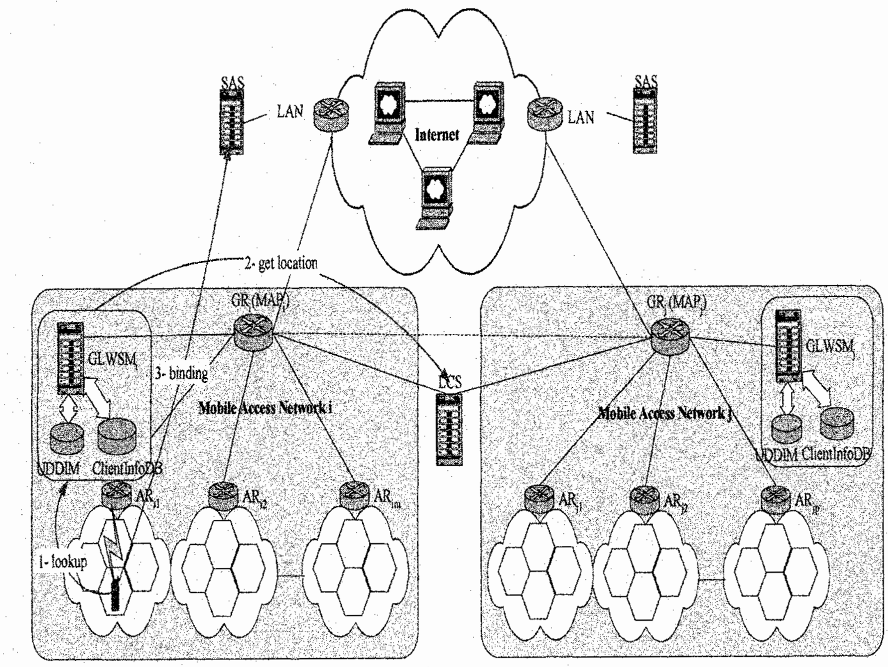

Figure 1. GLWSA architecture

3.4 GLWSA Topology

The GLWSA topology is bi-level hierarchical system. We chose a bi-level system to reduce the message exchange between GLWSMs in the service publication process and the propagation of the collected QoS data update.

The root level is mainly used to control the data coherence induced by the service publication in the system. There is only one GLWSM node. The leaf level has many GLWSM nodes. Each GLWSM leaf node is attached to a mobile access network. A GLWSM covers a geographical region that represents the location areas of the mobile access network with which it is associated. Two different GLWSM nodes cannot cover the same location areas. A SAS can be associated with many GLWSMs. A supplier's particular web service can be distributed over many SAS. The main

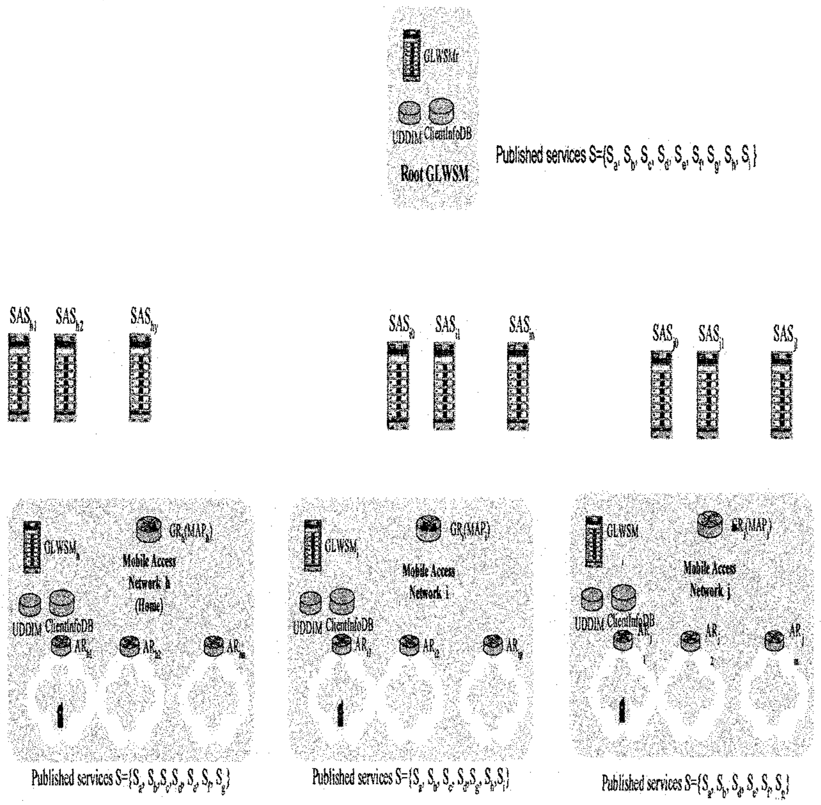

Figure 2. GLWSA topology

characteristic of the leaf nodes is that they are dependent on the location areas of the network operator to ensure the mobility tracking, to maintain service execution in a SAS, and to coordinate the service migration when a mobile client moves to location areas controlled by another GLWSM node where the service in execution exists. Otherwise, the service execution will continue to be provided by the SAS in progress. We impose a migration delay constraint of 2 seconds to migrate a service.

The service publication is controlled by the GLWSM root node and others functionalities (authentication and authorization, subscription, tracking mobile position, coordination of the migration, lookup) are offered by the leaf nodes. These are the relation properties of GWLSA:

Covered Areas: In the topology suggested, the location areas covered by two different GLWSM leaf nodes are disjoined.

Visibility Relation: Every GLWSM node that offers a service S_i knows all GLWSM nodes which offer the same service.

Home Node is the GLWSM leaf node where the mobile client is registered. Every mobile client is associated with a single home node. The mobile client used the home node to authenticate himself and to lookup for a service.

Visited Node is a GLWSM leaf node other than the home node of a mobile client.

Nearest Node is a GLWSM leaf node where the mobile client is located.

Nearest SAS is the SAS attached to the nearest GLWSM node and where the service requested by a mobile client is in execution phase.

4. GLWSA MATHEMATICAL FORMALISM

In order to facilitate the interactions implementation of the principal functionalities of the system with the databases UDDIM and "ClientInfosDB", we formalized the GLWSA concepts in an algebraic language.

4.1 Sets of discovering servers and location areas

Let E be the set of discovering servers GLWSM deployed at the leaf level of the topology tree. By definition:

$$E = \{GLWSM_1, GLWSM_2, ..., GLWSM_p\} \quad \text{(eq. 1)}$$

where p represents the cardinal of E or the number of GLWSM deployed in topology.

Let Z be the set of location areas covered by the GLWSA topology. By definition:

$$Z = \{Z_1, Z_2, ..., Z_t\} \quad \text{(eq. 2)}$$

where t represents the cardinal of Z or the number of the mobile network location areas associated the GLWSA topology.

<u>Mobile access network</u>: A mobile access network is a set of mobile network resources giving access to the network IP. The location areas covered by a mobile access network constitute a subset of Z which share the same MAP (Mobile Anchor Point).

<u>Domain</u>: We call domain a mobile access network controlled by a node GLWSM.

<u>Adjacency</u>: Two discovery servers GLWSM_j and GLWSM_{j+1} are contiguous if they have a common border of location areas.

Properties

Finished sets: At any moment, the sets E and Z are finished and determined sets. The cardinal of E or Z can increase or decrease dependently if there are an addition or a withdrawal of a $GLWSM_i$ node or a location area Z_x in the GLWSA topology.

Topology control: The root node is the single discovering services server which manages (adds, withdraws, modifies) the topological structure deployed in system GLWSA.

Location areas covered by a leaf node GLWSM: Each leaf node $GLWSM_i$ is associated with a domain D_i. This association means that the node $GLWSM_i$ covers the location areas D_i.

Disjunction of covered location areas: Two distinct nodes $GLWSM_i$ and $GLWSM_j$ cover disjoined location areas. Thus, if ZL_i and ZL_j represent the location areas covered respectively by $GLWSM_i$ and $GLWSM_j$, then:

$$ZL_i \cap ZL_j = \phi \qquad \text{(eq.3)}$$

4.2 Sets of deployed services and application servers in GLWSA topology

Let F be a set of supplier applications servers SAS deployed in the GLWSA topology. By definition:

$$F = \{SAS_1, SAS_2, \dots, SAS_m\} \qquad \text{(eq. 4)}$$

where m represents the cardinal of F or the number of SAS deployed in topology.

Let G be the set of Web services deployed in the GLWSA topology. By definition :

$$G = \{S_1, S_2, \dots, S_z\} \qquad \text{(eq.5)}$$

where z represents the cardinal of G or the number of deployed Web services in the topology.

Definitions

Services vector: We call a services vector $\overrightarrow{V_k}$, the set of deployed services deployed at GLWSMk node. We have:

$$\overrightarrow{V_k} = [S_a S_b \dots S_g] \bullet \overrightarrow{GLWSM_k} \qquad \text{(eq. 6)}$$

where $\{S_a, S_b, \dots, S_g\}$ are a subset of G.

service presence: A service Web S_j is present at GLWSMk node if and only if S_j is provided by at least SAS associated with GLWSMk. Consequently, if Sj is present y time at GLWSMk node, then the measurement of presence p is:

$$p = y \qquad \text{(eq.7)}$$

Vector of service presence: We call vector of service presence $\overrightarrow{PV_k}$, the vector materializing the set of web services present at the GLWSMk node.

$$\overrightarrow{PV_k} = [p_{1k} p_{2k} \ldots\ldots p_{zk}] \bullet \overrightarrow{GLWSM_k} \qquad \text{(eq. 8)}$$

where $p_{1k}, p_{2k}, \ldots, p_{zk}$ represent the presence of web services $S_1, S_2, \ldots, S_z$ of G at $GLWSM_k$ node. Matrix of service presence: We call matrix of service presence, the matrix M allowing to identify the presence of web services to any node GLWSM in the GLWSA topology. By definition:

$$M = \begin{bmatrix} p_{11} & p_{12} & p_{1p} \\ p_{21} & p_{22} & p_{2p} \\ & & \\ p_{z1} & p_{z2} & p_{zp} \end{bmatrix} \qquad \text{(eq. 9)}$$

where p_i represents the presence of the service S_i at $GLWSM_k$ node.

Properties

Finished sets: At any moment, the sets F and G are finished and determined sets. The cardinal of F or G can increase or decrease, dependently if there are an addition or a withdrawal of a supplier application server SAS_i or a service Web S_x in the GLWSA topology.

Number of offered services: A supplier application can offer one or more web services Web S_i.

Root node root and matrix of service presence: At any moment, the root node $GLWSM_r$ entirely knows the matrix of service presence M.

Relations

Supplier application servers and discovering services servers: Each application server SAS_k is associated with a discovering services server (leaf node) $GLWSM_i$ which covers its geographical position, and possibly other $GLWSM_c$ contiguous to $GLWSM_i$, if its distance compared to the associated discovering service nodes is lower or equal to the maximum distance tolerated between SAS and an associated node GLWSM of theGLWSA topology.

$$D(SAS_k, GLWSM_i) \leq D_{\max} \qquad \text{(eq. 10)}$$

where $D(SAS_k, GLWSM_i)$ is the distance between SAS_k and $GLWSM_i$.

4.3 Service publication

The publication of a service Web S_{z+1} in system GLWSA implies three essential operations:

1. unify the set G and the singleton { S_{z+1} }, we have:

$$G = G \cup \{S_{z+1}\} = \{S_1, S_2, ..., S_z, S_{z+1}\} \qquad \text{(eq. 11)}$$

on the level of the data base UDDIM, this operation consists of adding to the root node $GLWSM_h$ a new record to the service and agreement of service tables.

2. To add a new row L_{z+1} to the matrix of presence M, we have:

$$M = \begin{bmatrix} p_{11} & p_{12} & p_{1p} \\ p_{21} & p_{22} & p_{2p} \\ \\ p_{(z+1)1} & p_{(z+1)2} & p_{(z+1)p} \end{bmatrix} \qquad \text{(eq.12)}$$

on the level of the data base UDDIM, this operation consists of adding to any leaf node $GLWSM_i$ that has a non null presence of service $p_{z+1,i}$ a new record to the tables representing the service S_{z+1}, the agreement of service and the agreement of service at the particular $GLWSM_i$.

3. For any $GLWSM_i$ node having a non null presence of service $p_{z+1,i}$, confirm the recording of the service information published to the root node $GLWSM_h$.

4.4 Lookup service

The lookup of a web service S_d at the home node $GLWSM_h$ of a mobile customer CM implies six essential operations:

1. To locate the mobile client CM by formulating a request of position to the location server LCS.

2. To determine the server $GLWSM_s$ that covers the current geographical position POS_c of the mobile customer. That is equivalent in database algebraic language to select in the GLWSM nodes table, the node $GLWSM_s$ which covers the current geographical position POS_c of the client CM. This operation is equivalent to the following instruction:

$$\{T[GLWSM] \; where \; POS_c \subset AREA\} \quad [GLWSM_s] \qquad \text{(eq.13)}$$

where *AREA* materializes the geographical area covered by the node $GLWSM_s$.

3. To check in $UDDIM_h$ (associate with the home node $GLWSM_h$) if the presence of the S_d service is non null at the node $GLWSM_s$. From the point of view of the database, this operation consists in making a projection on the GLWSM node axis of the *table T[S,GLWSM,ADDRESS]* identifying the

agreement of service to a node when the service has S_d as value and GLWSM has $GLWSM_s$ as value. This operation is equivalent to the following instruction:

$\{T[S, GLWSM, ADDRESS] \text{ where } S = S_d \wedge GLWSM = GLWSM_s\} \quad [GLWSM]$ (eq.14)

4. To select all application servers SAS associated with $GLWSM_s$ and that offer the concerned service S_d. This operation consists in making a projection on the address axis of the table *T[S, GLWSM, ADDRESS]* identifying the agreement of service to a node when the service has S_d as value and GLWSM has $GLWSM_s$ as value. This operation is equivalent to the following instruction:

$\{T[S, GLWSM, ADDRESS] \text{ where } S = S_d \wedge GLWSM = GLWSM_s\} \quad [ADDRESS]$ (eq.15)

5. To check if the QoS criteria meet the QoS level required by the mobile client CM; this phase is optional if the mobile client did not require QoS in his service discovering request.

6. To return to mobile client CM, the URL address of the service S_d that meets the QoS level.

4.5 Coordination of the service migration

The coordination of migration of a web service S_d that is in execution to a current application server SAS_c associated with the current node $GLWSM_c$, begins when the node $GLWSM_c$ is notified by the location server LCS that the mobile client CM (implied in the execution of the S_d service) moves out the $GLWSM_c$ covered area. The following steps are executed thereafter:

1. To determine the next node $GLWSM_n$ that covers the current geographical position POS_c of the mobile client CM. This operation consists in checking which GLWSM nodes of the set E covers the current geographical position of the mobile client CM. This operation is equivalent in the database algebraic language to the instruction eq.13.

2. To check the existence of service S_d and to select all application servers SAS associated with $GLWSM_n$ and that offer the service S_d. This operation is equivalent in the database algebraic language to the instruction eq.15.

3. To check if the QoS criteria meet the QoS level required by the mobile client CM; this phase is optional if the mobile client did not require QoS in his service discovering request.

4. To select the next application server SAS_n where the service migration will be carried out.

5. The node $GLWSM_c$ notifies to the SAS_c about the SAS_n URL address where migration of the service S_d migration must take off.

6. The application server SAS_c notifies to the $GLWMS_c$ of the end service S_d migration at SAS_n server.

7. The *GLWMS$_c$* sends to the *GLWMS$_n$* a message to continue the position tracking of the mobile client CM in its covered area.

5. IMPLEMENTATION DETAILS AND EVALUATION

To evaluate the proposed GLWSA architecture, we built a prototype using the Java programming language (Jbuider 7 and Sun Message Queue 3.5). This prototype implements the functionalities of the GLWSM and communicates with the LCS and the SAS servers. Communication with the LCS was carried out through Ericsson MPS 6.0 emulator. The emulator implements the MLP V3.0 protocol to determine the geocraphical location of a client (or a group of clients) moving over the network topology. We generate a network topology with the network density set to suburban, the distance between two base stations is set to 5000 meters and created a mobile trajectory with a constant speed value.

In Figure 3, the GLWSM$_h$ is the home GLWSM domain of the target mobile client and its geographical covered areas, the GLWSM$_1$ and GLWSM$_2$ are the visited GLWSM domains of the target mobile client with their associated geographical covered areas. In our tests, we used three constant speeds 50 km/h, 100 km/h and 200 km/h. By using the configuration settings described above, a static route file that contains the current cell identifier where the mobile resides, the relative distance between the mobile and the current base station and the mobile position data calculated each 10 seconds and given in geographical system coordinate (latitude/longitude/altitude) is created. The MPS 6.0 emulator calculates the client position between two consecutive offset times of the route file (for example between 0 and 10 seconds) by using the interpolation operations. But the formula used to do the interpolation operations is not given by the MPS tool specifications. At the beginning of the simulation, the emulator starts a clock and reads the mobile position in function clock time in the static route file created. We used the database management system Oracle 9i to store data in UDDIM registry and ClientInfosDB. We used JUDDI [8] and appended UDDIM API to interact with the registry UDDIM. The machines used to materialize the GLWSM, the SAS and the LCS are similar (1.2 GHz Pentium III with 512 Mo RAM). The machines are connected in a wireless LAN. The WLAN has a transmision rate of 11 Mbits/s and is compliant IEEE 802.11b.

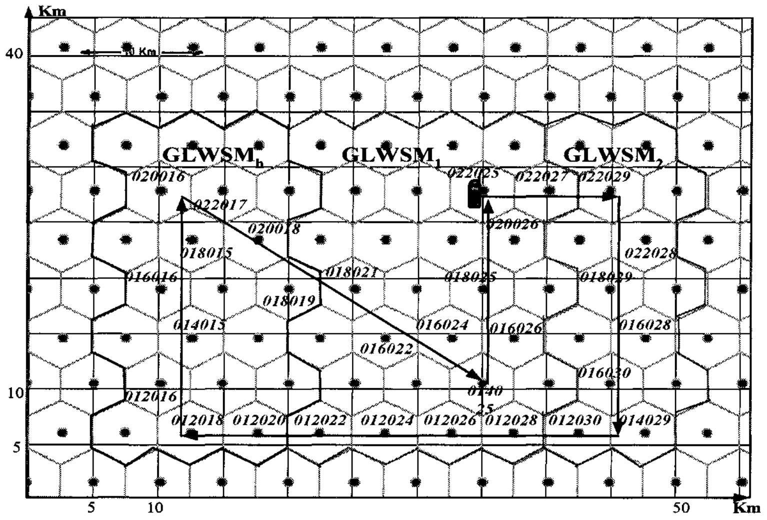

Figure 3. The generated network topology and trajectory of a target mobile client.

We define the round trip time (RTT) as the time difference between the reception time of the response at a client machine and the time when the client sent a request to a server. In Figure 4, we measured the RTT of a nearest service lookup without QoS by supposing that a mobile client sends a request each 2 seconds to his home GLWSM. The nearest lookup service request parameters sent are: service identifier and mobile identifer. The total size of sent parameters is 80 bytes.

We found a minimal RTT value of 20 milliseconds, an average RTT of 28 milliseconds and a maximum RTT of 52 milliseconds. The measure dispersion is 14.16 milliseconds.

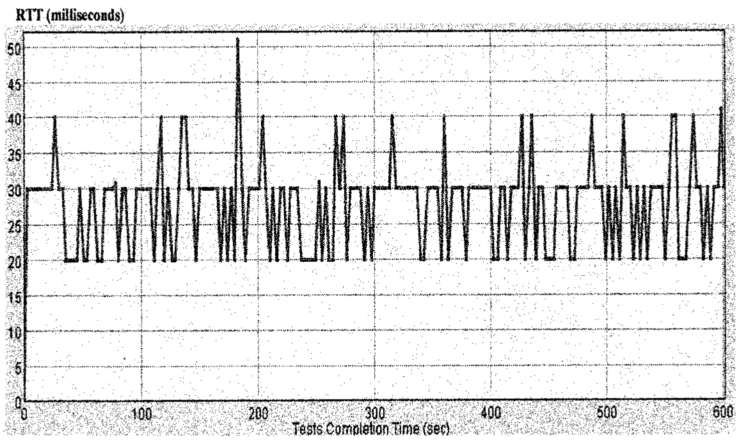

Figure 4. Round Trip Time of a nearest service lookup without QoS in case of one client.

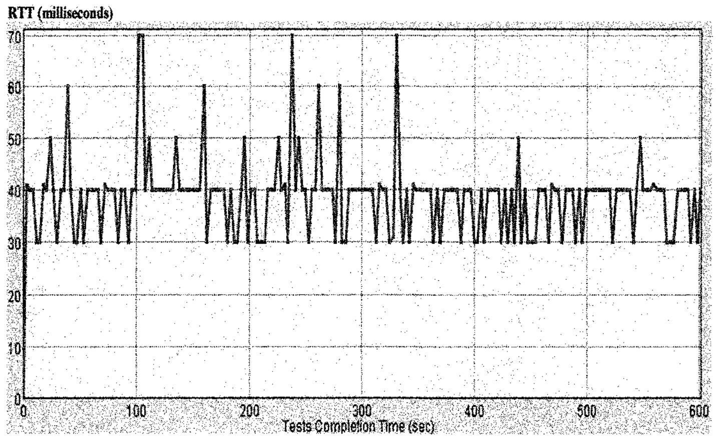

Figure 5. Round Trip Time of a nearest service lookup with QoS in case of one client.

To measure the RTT of the nearest service lookup with QoS, we suppose that a client sent to its home GLWSM a nearest lookup service request containing parameters: service identifier, mobile identifer, required service cost, required bandwidth and required SAS factor utilization. The total size of sent parameters is 104 bytes. We found a minimal RTT value of 30 milliseconds, an average RTT of 41 milliseconds and a maximum RTT of 70 milliseconds (Figure 5). The measure dispersion is 15.57 milliseconds.
The coordination time of a service migration shown in Figure 6 represents the RTT to send the URL of the next SAS (that implement the service in execution of a mobile client) and the mobile identifier parameter of a mobile client (who just changed the SAS domain) to the current SAS in execution. At the receiving of the message, the current SAS just sends back an acknowledgement to the GLWSM sender. Then, the GLWSM sender notifies the next GLWSM to track the target mobile. The coordination time of a service migration has an average RTT of 11 milliseconds, a minimal value of 7 milliseconds and a peak value of 40 milliseconds. . The measure dispersion is 5.80 milliseconds. We varied the speed of the target mobile and we remarked that the speed does not have a direct impact in the coordination of the service migration.
Coordination test of the service migration with QoS (Figure 7) consists first to send the URL address of the next SAS and the mobile identifier to the current SAS if a mobile client moves in the area covered by the next SAS. At the reception of the message, the current SAS just sends back an acknowledgement to the GLWSM sender (current GLWSM). Then, upon receiving the receipt of service migration end, the current GLWSM notifies the next GLWSM to track the position of the mobile client who just enter into its covered area. To track the

mobile client, the current GLWSM sends to the next GLWSM the relevant information to do this operation (mobile identifier, service identifier, next SAS URL, client required service cost, client required bandwidth, client required SAS utilization rate). The total size of information sent to the next GLWSM is 104 bytes. The coordination time of a service migration has an average RTT of 13 milliseconds, a minimal value of 8 milliseconds and a peak value of 65 milliseconds. We varied the speed of the target mobile and we remarked that the speed does not have a direct impact in the coordination of the service migration. The measure dispersion is 6.47 milliseconds. Compare to the service migration

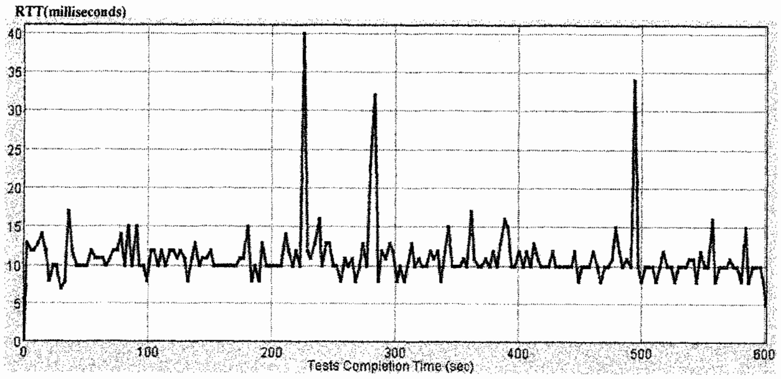

Figure 6. Coordination time of a service migration without QoS.

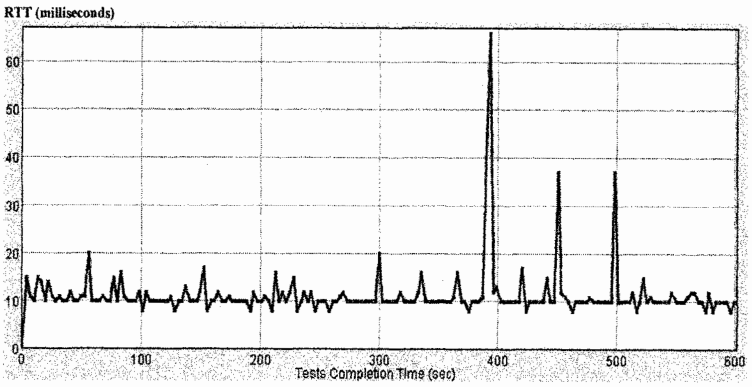

Figure 7. Coordination time of a service migration with QoS.

without QoS, we found a mean RTT variation of 2 milliseconds. This variation is due principally to the increasing of 24 bytes size parameters sent to the current SAS.

Meanwhile, as we imposed that the delay migration constraint must be less than or equals to 2 seconds, if a mobile client has a speed of 300Km/h when the migration is relevated, the target mobile will be at 166,67 meters of the precedent GLWSM domain when the service migration will be terminated. As the service migration for SAS to SAS has a maximum average rate of 300 milliseconds [4], we will have a maximum total service migration time (SAS service migration time and coordination migration time) of 355 milliseconds which is less than the delay migration service constraint. Thus, the system has a margin time 1645 milliseconds for huge applications.

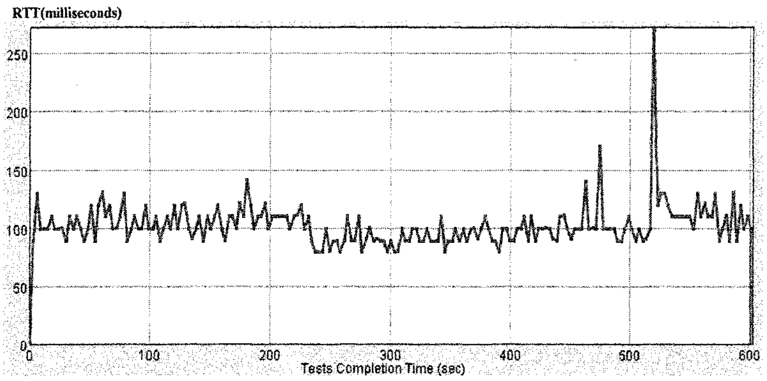

Figure 8. Geo-located web service publication time in two GLWSM leaf nodes.

Figure 8 shows a geo-located web service publication time in two GLWSM leaf nodes. To measure the service publication, we suppose that an authorized supplier sent a size of 1116 bytes parameter data (service, agreement service, agreement node entities) to publish to the root GLWSM machine. Then, the root GLWSM stores the data in his UDDIM and forwards them to the publication queue listened by the two GLWSM leaf nodes. After storing the forwarded data in their UDDIM, each GLWSM leaf node sends back a recept to the root GLWSM. We found an average RTT of 105 milliseconds with a minimal and maximal RTT of 88 and 300 milliseconds during 10 minutes observation. We also analyzed the publication time in function of the number GLWSM leaf nodes (which increases the size of parameters sent in a publication process) where they are published. We found that the publication time increases when the number of GLWSM leaf nodes increases too. We found an average RTT of 110, 147 and 247 milliseconds for 5, 10 and 20 GLWSM leaf nodes, respectively.

6. CONCLUSION

We presented in this paper a discovering architecture for geo-located web services for the next generation mobile networks. With tis architecture, mobile clients can discover geo-located web services and maintain the service execution closest to their location context. We extended the UDDI registry and MLP protocol to reach this goal. Tests executed in the GLWSA system show that the system gives good response in lookup, migration and publication service in the context of a client mobility. Future work will present the mechanism of QoS collection, the thematic location and will consider the adaption of data presentation in different client format machines (xHTML *eXtensible HTML*, cHTML *compact HTML*, WML (*Wireless Markup Language*) and the conviviality of the prototype.

REFERENCES

1. **3GPP**, *Functional stage 2 description of location services in UMTS*, Technical Specifications TS 23.171 V3.9.0, Reference http://www.arib.or.jp/IMT-2000/ARIB-spec/ARIB/23171_300.PDF, September 2002.
2. **G. Ballintijn, A. S. Tanenbaum** and **M. R. Van Steen**, *Locating Objects in a Wide-area System*, PhD Thesis, Amsterdam University, Globe Project, Reference http://www.cs.vu.nl/res/theses/ballintijn_thesis.pdf, 2003.
3. **M. Barbeau**, *Bandwidth Usage Analysis of Service Location Protocol*, Workshop on Pervasive Computing, International Conference on Parallel Processing, Toronto, Reference http://citeseer.nj.nec.com/barbeau00bandwidth.html, August 2000.
4. **S. Bouchenak, D. Hagimont, S. Krakowiak, N. De Palma** and **F. Boyer**. *Experiences Implementing Efficient Java Thread Serialization, Mobility and Persistence*, INRIA Technical Report No. RR-4662, December 2002.
5. **M.A Dru** and **S. Saada**, *Location based mobile services: the essentials*, Alcatel Telecommunications review, pp. 71-76, 1st quarter 2001.
6. **E. Guttman**, *Service Location Protocol : Automatic Discovery of IP Network Services*, IEEE Internet Computing, pp. 71-80, July-August 1999.
7. **N. Harashima, T. Okoshi, J. Nakazawa, Y. Tobe**, and **H. Tokuda**, *AMRB: Toward Location Migration Transparency of Services*, IEEE International Conference on Parallel and Distributed Systems, pp. 305-314, ICPADS 2001.
8. **jUDDI**, http://ws.apache.org/juddi/ .
9. **Location Inter-operability Forum (LIF)**, *Mobile Location Protocol*, LIF TS 101 Specification, Version 3.0.0 6, Reference http://dan.greening.name/profession/manuscripts/LIF%20TS%20101%20v3.0.0.pdf, June 2002.
10. **F. Michahelles, M. Samulowitz** and **B. Schiele**, *Detecting Context in Distributed Sensor Networks by Using Smart Context-Aware Packets*. In International Conference on Architecture of Computing Systems, Reference http://www.vision.*ethz.ch/publ/arcs02.html, ARCS 2002.*
11. **S. Panagiotakis** and **A. ALonistioti**, *Intelligent Service Mediation, for supporting advanced location and mobility-aware service provisioning in reconfigurable mobile networks*, IEEE Wireless Communications, Vol. 9, pp. 28-38 , October 2002.
12. **A. Y. Seydim, M. H. Dunham**, and **V. Kumar**, *An architecture for location dependent query processing*, Proceedings in 12th International Conference on Database and Expert System Applications , pp. 549-555, DEXA 2001.
13. **UDDI**, http://www.uddi.org/specification.html.