Implementing Mobile Location Protocol

Petr Vlacil
Czech Technical University in Prague
Department of Telecommunications Engineering
Email: vlacip1@fel.cvut.cz

Rober Bestak
Czech Technical University in Prague
Department of Telecommunications Engineering
Email: bestar1@fel.cvut.cz

***Abstract*—Location based services provide new opportunities for service providers to attract customers. Main part of location base service architecture in GSM network is Gateway Mobile Location Center. The essential task of this signaling node is to provide a position of mobile terminals to the location clients that request it. The node also represents a gateway from the IP network side to the mobile network. Standardized interface to communicate with Gateway Mobile Location Center from the IP side is the Mobile Location Protocol. This article describes possible design of implementation of Mobile Location Protocol. Implementation concentrates on providing library that could be easily used by other projects.**

***Index Terms*—Location, Location Based Services, Mobile Location Protocol, XML.**

I. Introduction

Standardized way to implement Location Based Services (LCS) in the mobile networks is to upgrade the radio access network with localization procedures such as Time Difference of Arrival (TDOA) or Enhanced Observed Time Difference (E-OTD) [1] [2]. The next step is to implement a Gateway Mobile Location Center (GMLC). The GMLC queries position of terminals in a network. The GMLC converts position information to the standardized form and provides it to other entities called Location Clients. Location Clients can be service providers or mobile phones.

Standardized interface allows external entities that are not part of mobile network to use services of mobile network without knowing its complicated structure. This yields to the separation of network providers and service providers.

Most widespread communication network technology is IP network. Location Clients communicate with GMLC by the standardized Mobile Location Protocol (MLP) which is designed to be carried in Hypertext Transport Protocol (HTTP) protocol over IP networks.

Mobile location platforms are developed and sold by known companies like Ericsson and Alcatel-Lucent. This platforms are expensive. There is no free opensource implementation of MLP that could be used to develop cheep services by smaller providers.

II. Mobile Location Protocol

Mobile Location Protocol (MLP) is application-level protocol designed for requesting and reporting subscribers location. According to 3GPP standard [3] it is used on the Le interface in Mobile networks as seen on figure 1. MLP is specified by Open Mobile Alliance (OMA). MLP is based on XML and is defined to be transported in Hypertext Transport Protocol (HTTP) or Simple Object Access Protocol (SOAP). Only mapping to HTTP is defined in present days. For security reasons Secure Socket Layer (SSL) or Transport Layer Security (TLS) cryptographic protocols can be used to carry HTTP (HTTPS). Apart from HTTPS Mobile Location Protocol uses authentication by user and password. This information is carried in header part of MLP message.

Last draft of MLP specification is OMA-TS-MLP v 3.3 from March 2009 [4]. Version 3.3 provides these features:

Standard Location Immediate Service (SLIS) Used for requesting the location of one or more Mobile Subscribers (MS). It is used when immediate response is required. Response can be returned in more then one message if the request contained more Mobile Subscribers.

Emergency Location Immediate Service (ELIS) Used for requesting the location of one or more MS involved in an emergency call. Response can be returned in more then one message if the server supports it.

Standard Location Reporting Service (SLRS) This service is initiated by MS by sending Mobile Originated Location Request [TS23271] through wireless network. It is preformed when Mobile Subscriber wants an LCS client to receive MS location.

Emergency Location Reporting Service (ELRS) This service is initiated by network when MS initiates or releases emergency call. Report is generated and sent from Location Server to predefined LCS client.

Triggered Location Reporting Service (TLRS) Service is used when LCS client wants to track several MS. Triggers

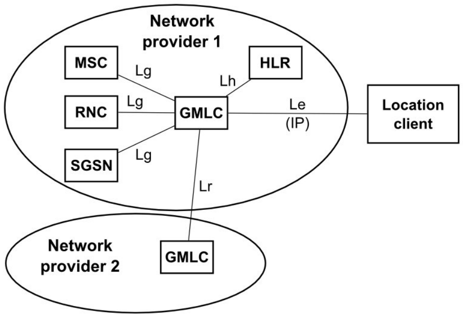

Fig. 1. Architecture of location system

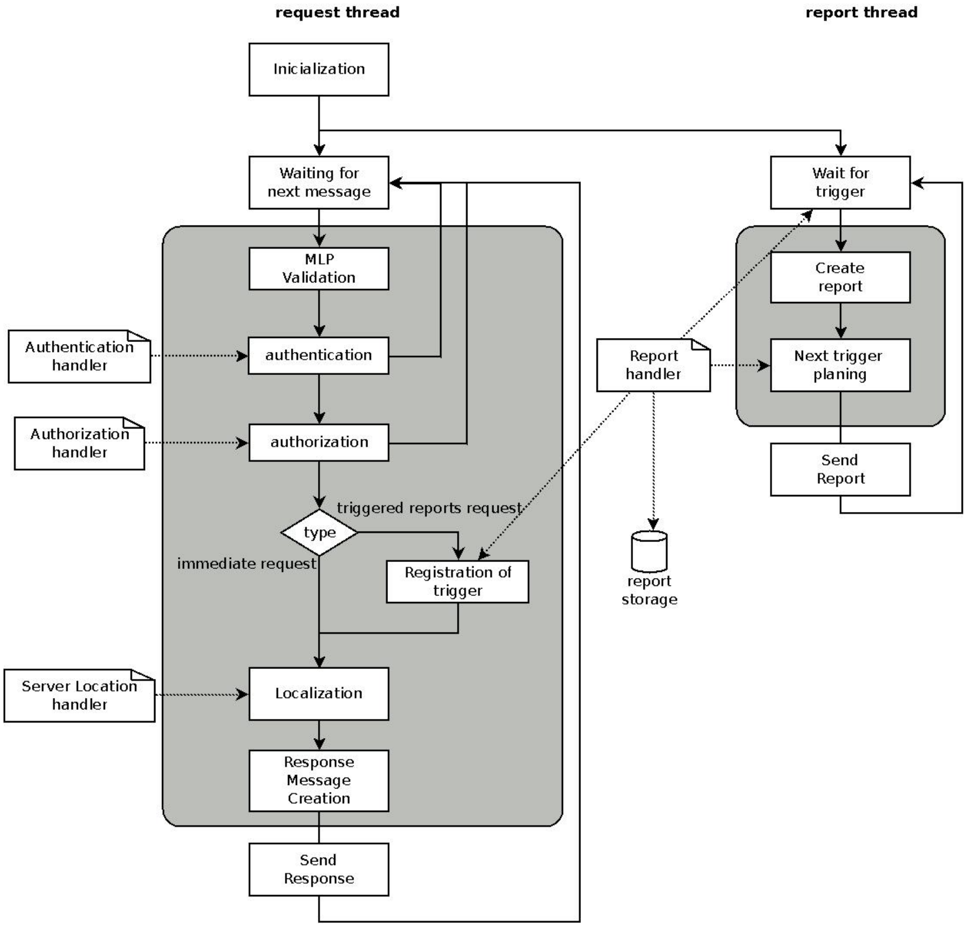

Fig. 2. Location server flow chart (Grey part is processed by MLP library)

can be timers or MS events like change of location.

Historic Location Immediate Service (HLIS) Is used to request reporting of historic locations of one MS when large amount of data is expected.

Mobile Location protocol provides synchronous and also asynchronous mode of communication.

A. Synchronous mode

In this mode location client sends the HTTP POST request that contains any MLP service request message. Service request have the attribute res_type="SYNC". Location client expects exactly one response message which contains the result.

B. Asynchronous mode

In this mode client sends the HTTP POST request. MLP service request have the attribute res_type="ASYNC". In this case client expects one answer message and zero one or many report messages, that are sent as the HTTP POST messages.

III. Implementation

MLP Protocol is based on XML. Processing of XML could be done in two ways, using Document Object Model (DOM) or Simple API for XML (SAX). DOM parsing is more memory consuming. Whole parsed XML document is loaded into memory. Main benefit of DOM parser is possibility to randomly jump through document. SAX parser reads XML documents consecutively and allow to pick only information of our interest. Therefore it is less memory consuming. SAX uses call-back functions which are called at the different events in a document, for example start of XML element.

Main criteria for discussed MLP implementation is speed, efficiency and conformance to standards. SAX parsing method was chosen to process MLP messages for two reasons. There is no need to jump through MLP message randomly. SAX parser saves memory which is important in server applications. Xerces-C library was chosen to work with XML. C/C++ programming language is suitable for server applications because it produces more effective programs that do not need any middleware runtime environments. As stated in [5], implementation of MLP without middleware is faster.

Developer that want to use Xerces SAX parser need to program so called handlers for different events. Default handlers are concentrated in DefaultHandler class. StartElement, characters, and endElement handlers of Xerces-C will be used to implement MLP parser. These handlers are called on start of all XML elements, text content of XML elements or at

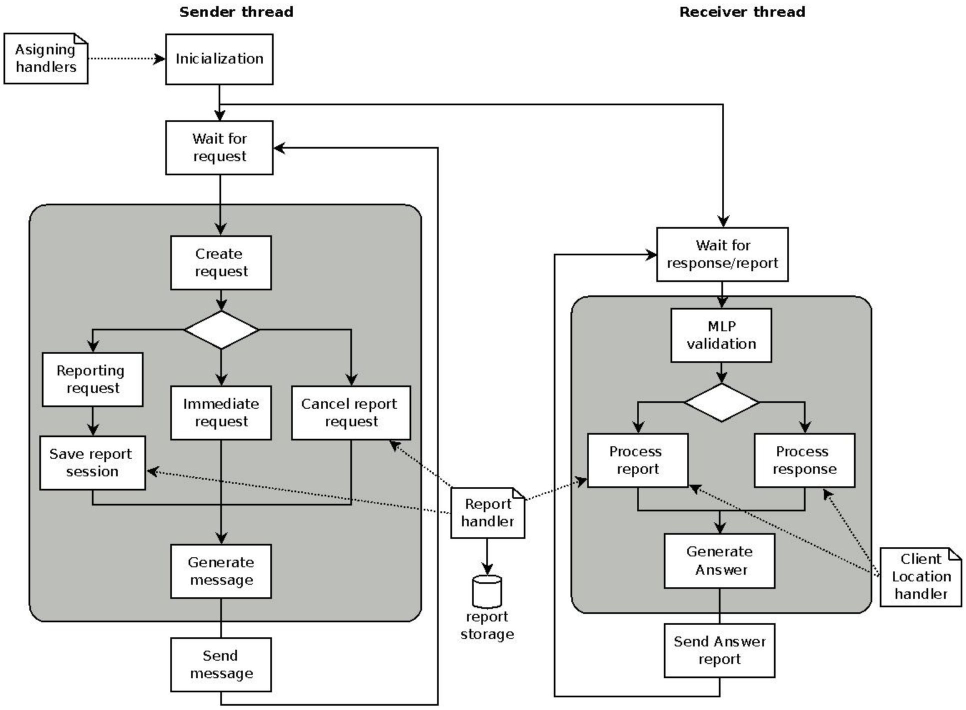

Fig. 3. Location client flow chart (Grey part is performed by MLP)

the end of every XML element. These handlers will decide what to do with these XML elements and their content. Other handlers that need to be implemented are warning, error and fatalError handlers. These handlers take care of melformed XML messages.

Xerces SAX parser [6] validates incoming XML messages against Document Type Definition (DTD) definitions or XML Schemas. OMA provides DTDs for MLP so DTDs will be used. Every received message should contain DOCTYPE section that specifies which DTD should be used for the validation. Xerces automatically downloads the DTDs from address specified in DOCTYPE. For security reasons GMLC might not be accessible from Internet or might not have access to OMA web-pages where DTDs are stored. Because of this. Default behavior of Xerces have to be modified so it loads DTDs locally from hard drive. DTDs should be cached in memory so it does not have to be loaded from file every time the message arrives. To achieve this, modified Xerces EntitiResolver object need to be created. This object inspects Uniform Resource Identifiers (URI) of DTDs received in messages and changes them to proper local ones. Then DTD can be stored locally.

Mobile Location Protocol does not define how the authentication, authorization, localization or other procedures are performed. Therefor MLP implementation will call handlers that will perform this actions. MLP library will provide necessary information from MLP messages that are needed for handlers to work. The handlers will be implemented by programmers that are using MLP library.

External user-defined handlers will connect MLP with other entities like databases, location managers that provide location of subscribers, timers and so on. This way MLP is independent of system architecture, program architecture or other factors.

MLP consists of two logical parts. One part is server that listens to the Location Clients requests and returns results. Other part is performing Reporting service. This reporting service uses triggers to sends location reports to Location Clients.

A. *Server part of MLP library*

Server should be implemented at least as two separated threads or processes. One or more threads are waiting for requests from clients and the other thread is processing reporting triggers.

Flowchart of the server can be seen at figure 2. At the beginning of operation MlpParser have to be created and initialized, than handlers are assigned. After the initialisation program can wait for incoming requests.

Request is passed to the library as a text or stream. Request is validated and Location client is authenticated and authorized. User-defined handlers are used for these operations. Then type of message is recognized. According to the type of message further processing is dissimilar. Request for reporting service is remembered in user-defined storage. Reports are then generated in reporting thread.

User is located using Location handler. Response message is generated from location information. User of the library gets response message in text data form or stream form. After the response message is completed it can be sent in user-specific way to the client.

Reporting thread is waiting for the triggers. Triggers can be timeouts or subscribers location change. Location change triggers must be supported by the mobile network or else Location server would regularly check subscribers location to see if location changes. This would generate enormous signaling traffic. On the other hand timeout triggers can be implemented without network support. There is only one limit on how many time triggers can be registered at the system. This limitation is computation power and available memory size of the server.

B. Client part of MLP library

Location Client flowchart can be seen on figure 3. Location client can be split into two threads or more. One is generating requests to locate clients. The second one is waiting for reports from location server.

Request is generated according to request from outside of the MLP library. Generated message can be Immediate Request, request for Reporting service or request to Cancel Reporting service. Message is returned to the user of MLP library as text or stream. Sending of message is not objective of MLP library. If the user is demanding reporting service, information about this fact is stored in some user-defined storage so it can be canceled later.

Receiving thread is waiting for location responses or reports. Incoming messages are validated and processed. Reports needs to be answered with Report answer message. Location information from incoming messages is handled by user-specific Client Location handler.

IV. Conclusion

Mobile Location Protocol (MLP) is a standardized protocol that is widely used in telecommunications in Location Based Services. It is designed to interconnect the Gateway Mobile Location Center (GMLC) with the Location Service Providers [7].

Companies that develop and implement Location Based Platforms mainly use MLP to communicate from their Location Based platform to a mobile network. Free opensource implementation of MLP could provide a way for smaller service providers to create new value-added services.

This article describes possible design of the Mobile Location Protocol library. Library should be implemented in the way that it can be used in multiple threads to increase performance of the server and client. This means that critical parts of library should be thread-safe. Implementation of MLP library was still under a development during writing of this article.

Acknowledgment

This research work was supported by Internal Grant Service of Czech Technical University in Prague. Number of grant is 11-892130-13132 (CTU0921313).

References

[1] TS 43.059, *Functional stage 2 description of Location Services (LCS) in GERAN*, Release 8 3GPP, 2008-08.

[2] TS 25.305, *Stage 2 functional specification of User Equipment (UE) positioning in UTRAN*, Release 8 3GPP, 2007-12.

[3] TS 23.271, *Functional stage 2 description of Location Services (LCS)* , release 8 3GPP, 2008-12.

[4] OMA-TS-MLP, *Mobile Location Protocol*, draft v3.3, Open Mobile Alliance, 2009

[5] Kapellaki, S.; Tselikas, N.; Chaniotakis, E.; Koutsoloukas, E.; Prezerakos, G.; Venieris, I.S., *The "Le" interface: performance evaluation of 2-tier and 3-tier 3GPP compliant realizations*, IEEE International Conference on Communications, 2005

[6] Apache Software Foundation, *Xerces-C library*, online, ⟨http://xerces.apache.org⟩

[7] LocatioNet, *LBS Middleware*, online, http://www.locationet.com/LBSmiddleware.php