Cloud-based Real-time location tracking and messaging system: A child-care case study

Cong-Thinh Huynh
Department of Computer Engineering
Kyung Hee University
Yongin, South Korea
thinhhc@gmail.com

Huu-Quoc Nguyen
Department of Computer Engineering
Kyung Hee University
Yongin, South Korea
huuquoc09@gmail.com

Xuan-Qui Pham
Department of Computer Engineering
Kyung Hee University
Yongin, South Korea
pxuanqui@gmail.com

Tien-Dung Nguyen
Department of Computer Engineering
Kyung Hee University
Yongin, South Korea
ntiendung@gmail.com

Eui-Nam Huh*
Department of Computer Engineering
Kyung Hee University
Yongin, South Korea
johnhuh@khu.ac.kr

ABSTRACT

Cloud Computing represents an attractive and cost-efficient of server-based computing and application service provider models. Together with Cloud Computing development, we have witnessed the rapid increasing of mobile devices industry. Smart-phone is now a well-functioning system of GPS navigation, 3G/4G mobile network, wifi technology and much more. By using Cloud Computing, traditional mobile applications often involve with many Cloud Services such as online data storage, collaboration, real-time monitoring, web, email, push messaging, database processing, compute processing and so on. As the extension of mobile applications, real-time location tracking and messaging system has become particularly important. In this paper, we present a cloud-based real-time location tracking and messaging system (CRLTMS), that consists a cloud-based push messaging service, namely Google Cloud Messaging (GCM), web server, database and GPS navigation data. An Android-based child-care application case study shows that the effectiveness of the system, which enables the server tracks and communicates synchronously to smart-phones in real-time manner.

Categories and Subject Descriptors

[**Networking/Telecommunications**]: [Ubiquitous computing, Mobile computing, Mobile networks, Grid and cloud computing]

*Eui-Nam Huh is the corresponding author.

IMCOM '15, January 08-10, 2015, BALI, Indonesia
Copyright 2015 ACM 978-1-4503-3377-1 ...$15.00
http://dx.doi.org/10.1145/2701126.2701161.

General Terms

Application

Keywords

Real-time system, cloud computing, mobile cloud computing, cloud messaging, cloud push service, Android, Google Cloud Messaging (GCM)

1. INTRODUCTION

Recently, there has been a remarkable upsurge in Cloud Computing development, where computational processing, real-time monitoring, data sharing, storage, access and manipulation can be achieved in cost-effective, secure, low upfront costs and pay-as-you-go models are considered as advantages of clouds [1]. On the other hand, with the development of mobile technology and communication network, mobile internet has been gradually incorporated into people's daily lives, transforms the Cloud internet services and empowers end users with rich mobile experience.

By using mobile cloud computing (MCC), health-care services are being upgraded, achieve many good results and provide many services for human life. More particularly, in child-care service, we love nothing more than our children, we have to keep an eye on them and want to know that they are safe and doing well at any time. Technology has reduced the distance problem like never before, communication technology has revolutionized the way we communicate today. With the advent of smart-phone, mobile application and cloud internet as well, we can have a powerful tool to track our kids on the move anytime and anywhere. Hence, the real-time application and system has attracted more attention to developers. To address this motivation, we develop a cloud-based real-time location tracking and messaging system (CRLTMS) and apply for child-care scenario.

The CRLTMS, as illustrated Figure 1, is the combination of child's phone, which connected through internet to a monitoring server, Google cloud push service and by using smart-phone, tablet or laptop, parents can watch their children's move and communicate with them in real-time de-

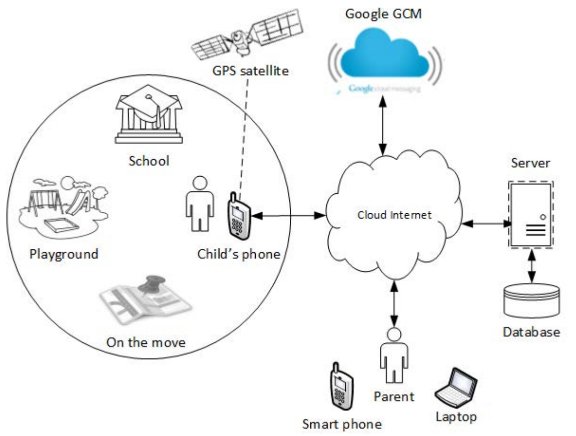

Figure 1: The cloud-based real-time location tracking and messaging system (CRLTMS)

mand via web browser or mobile application. We investigate two objectives as follows:

- send and receive messages between parents and their kids in real-time behaviors
- real-time location tracking, automated emergency notification based on location.

The remainder of this paper is organized as follows: Section II briefly introduces the background, techniques of this research. Section III comes up with the system architecture and design while section IV explains the system evaluation. Finally, section V summarizes this work as well as presents the conclusions and proposes future research investigation.

2. RELATED WORKS AND PROTOCOLS

2.1 Related works

Researchers have made significant efforts to Google GCM development and implementation. Many studies and applications were made and applied to different purposes. Penghui Li *et al* [2] implemented the original GCM into an Android application handle data synchronization between Android and server-side, the result from this study show that it enhances the timeliness and effectiveness of the information, reduces bandwidth traffic and saves battery power. As another approach, Shetty *et al* [3] developed a cloud GCM enabled online quiz application, this allows students to answer an online quiz from anywhere and the results and score instantly generated and sent to students. Gurek [4] developed an Android based home automation system, that allows users to control the appliances by an Android application or through a web site; in this paradigm, Google GCM cloud platforms are used to support messaging between home appliance devices, web server and mobile device running Android application.

In health-care services, Das *et al* [5] and Koufi [6] proposed two health-care services based on Google GCM cloud service, location and Android application, it is customized asynchronous notification feature whereby caregivers are notified on critical data updates and text to get the nearest healthcare centers from user's current position during any emergency situations. Their solutions either provide a number of benefits to the patient and the health service or manage and share their health records also provide cost-efficient, high-quality care in real world problems. In this paper, we take advantage of cloud computing, mobile devices and implement the algorithm for distance calculation and real-time notification.

2.2 Existing popular protocols

There are quite a number of protocols using client-server structure and some others peer-to-peer architecture to exchange data between network components as following.

Simple Object Access Protocol (SOAP): This is a light-weight protocol used for exchanging information in the implementation of web services in computer network environment. This protocol is based on XML and client/server architecture, it consists of three sections: SOAP-EVN envelope which defines a framework for identification of the message contents and how it can be processed, a set of encoding rules for expressing instances of data types, and a convention for representing remote requests and responses. SOAP specifies HTTP header and an XML file so that program can call a program in another computer and return response [7].

Session Initiation Protocol (SIP): SIP is a client/server architecture based protocol. This was aimed at provision of interactive communication. SIP provides support for applications that provides real-time communication among registered clients [8].

The Internet Relay Chat (IRC): This is one of the most popular chat protocols. This chat protocol is based on the client/server, TCP and TLS architecture. It allows real-time communications between the clients and the web server. IRC suffers from a lack of security features, amount of shared state data, users nickname collision problem, the trade-off in scalability for the sake of real-time user presence information. Network failures are witnessed most times when the system is in use [9].

Extensible Messaging and Presence Protocol (XMPP): It is a very standard protocol with many present and past available implementations. XMPP is based on XML and intended for instant messaging, collaboration, lightweight middleware, content syndication and online presence detection. This protocol can eventually allow users to send an instant messages to anyone else on the internet, regardless of differences in browsers and operating systems. Login procedures and distribution of presence information are managed by a client-server interaction, while messages are been relayed directly to the clients in a peer-to-peer manner [10].

3. BACKGROUND

3.1 Android Operating System

Android is a free, open source mobile operating system (OS) based on the Linux kernel and currently developed by Google. Android platform has four main layers, on top of world Linux kernel, there are libraries and APIs written in C, application framework and application layer. Android uses the Dalvik virtual machine (DVM) and Android Runtime (ART) to translate Java bytecode into Dalvik dex-code [11]. Android is a powerful mobile OS, it is configured to link and work with all hardware devices and sensors. In this study, we develop an Android-based application, which handles GPS information and manipulates messages throughout the system.

3.2 Google Cloud Message (GCM)

GCM is a cloud push message service that helps developers manipulate light-weight data from servers to Android devices in real-time demand [12]. GCM server handles all messages queuing and delivers to the Android application running on the target device as the proxy server. In CRLTMS, application server first receives and sends the message from the parent's device to GCM server, then the GCM server will push this message to appropriate child's device. By using GCM, we can reduce the network traffic and battery consumption of Android devices. We further discuss the mechanism of this process in next section of this paper.

3.3 Google Maps API

Google Maps provides developers set of Google Maps API functions that can be used to operate components or applications. By interpreting Google Maps APIs, we can develop rich-feature applications with three different map views, numerous destinations, street view mode, driving directions and advanced geo-coding. In our research, we mainly focus on geo-coding, location and interact with the longitude, latitude data in embedded map.

3.4 Application server (Apache PHP and MySQL database)

As the core of CRLTMS, application server comprises of Apache PHP web server and MySQL database. Apache PHP web server is developed and maintained by an open community of developers, and it plays an important role in the initial growth of World Wide Web. Apache supports a variety of features, modules and common language interfaces such as Perl, Python, Tcl and PHP [13]. PHP [14] is a lightweight and powerful server-side scripting language for web development. PHP is now the most-used web programming language, e.g. Facebook, Digg, Wordpress, MediaWiki, Joomla, etc. Besides, MySQL [15] is a feature rich, open-source database that powers a lot of websites and application. MySQL supports many SQL functionalities, ensure security, speedy application, data scalable and powerful. In this study, MySQL database is used to store all parents, children and GPS locations data.

4. SYSTEM ARCHITECTURE & DESIGN

In this section, we first introduce the architecture, which our work is based on and then present the design of system that paper is to address.

4.1 System Architecture

The core architecture of CRLTMS is depicted in Figure 1, in which, application web server is configured as important part of the system, that manages the information of parents, stores child's ID and GPS navigation information from child's phone, handles requests and resend them as needed using exponential back-off [16], also works as a communication bridge between parent, Google Cloud Message (GCM) and children. On the other hand, on the top layer of CRLTMS, GCM is used to manipulate the messages between web server and child's device. GCM supports two connections types: Hypertext Transfer Protocol (HTTP) and Cloud Connection Server (CCS) - an Extensible Messaging and Presence Protocol (XMPP) endpoint. We can either use HTTP or CCS separately to send upstream (device-to-cloud) or downstream (cloud-to-device) messages. In this paradigm, parents can monitor the location, moves, also send and receive messages from child's device. As a part of automated system, application server can automatically send an alert message to child's device when they go outside the defined safe range, compare to the center point. This is an implementation of Haversine formula [17], to calculate the distance between two points in the Earth by their longitudes and latitudes. The details of CRLTMS will be represented and analyzed in next section.

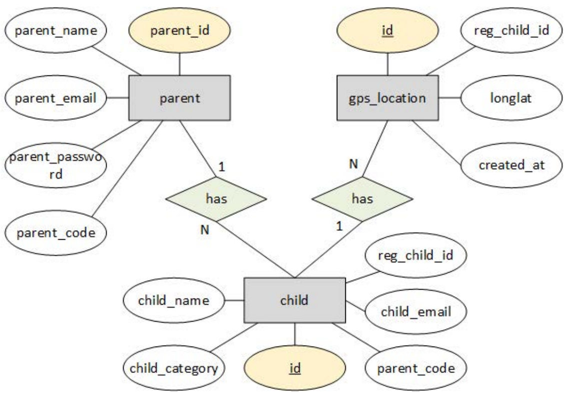

Figure 2: Database ER diagram

4.2 System Design

4.2.1 Database ER diagram

The database has three main tables as depicted in Figure 2. Table **parent** has primary key *parent_id*, defines the parent's name, email, password and code while *parent_code* is an unique value, this value is generated when parent register their account into CRLTMS at the beginning. This unique value is used to differentiate other codes and to activate the child's device into the system. Whereas, table **child** has more specific columns, *reg_child_id* is a unique string, generated by GCM and used to identify child's device for messaging service. In this table, *parent_code* is a foreign key, linked to *parent_code* in **parent** table. The last table is **gps_location**, it stores all the longitudes, latitudes (longlat) location and created date of longlat value, *reg_child_id* maps to the same column's name in table **child**.

4.2.2 CRLTMS workflow diagram

To ensure the seamlessness of whole system, all entities are configured and data is sent follow the sequence of system. The workflow diagram of our CRLTMS is illustrated in Figure 3 as follow:

Step 1. Register account: At the beginning, parents have to register an account on application server website, the system will generate and random code (*parent_code*) for them. This code is used to add and verify a child into system.

Step 2. Store record: All parents' information, e.g. name, email, password are saved into database. These information will be called in **step 5**.

Step 3. Request reg_child_id: This step is to request

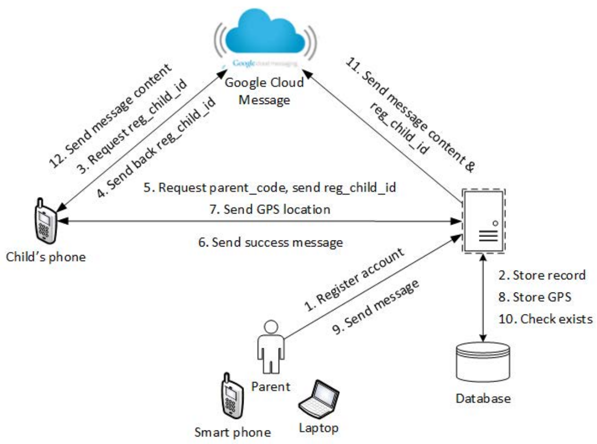

Figure 3: CRLTMS network paradigm

a device ID from GCM and register to application server.

Step 4. Send back reg_child_id: GCM sends back client registration IDs (*reg_child_id*) to device.

Step 5. Request parent_code, send reg_child_id: Device requests *parent_code* and *reg_child_id* which derived from previous step to application server.

Step 6. Send back success message: At this time, application server checks for *parent_code* whether it exists or not, if yes *reg_child_id* will be stored into database and mapped to *parent_code*. After finishing this step, parents can check for their children from web browser or smart-phone using email and password from **step 1**.

Step 7. Send GPS location: In this step, child's device continuously send GPS location through internet to application server. In order to save battery, we set the sending condition based on time scale or changed GPS data.

Step 8. Store GPS location data: Application server stores all the GPS location and its timing to database. By using AJAX, calls the SQL query so that newest GPS data will be assign into an queue array and real-time update on embedded Google Maps without refreshing the webpage.

Step 9. Send message: Parents, in order to send message to child's device, they first select *reg_child_id* then enter and send the message content to application server.

Step 10. Check reg_child_id exists: Application server handles the request from parent and check for existence of *reg_child_id* in database.

Step 11. Send message content & reg_child_id: Application server continues to send message content and *reg_child_id* to the GCM servers.

Step 12. Send message content: GCM enqueues and stores the message in case child's device is offline. If the device is connected to internet, GCM sends the message to target device through HTTP or CCS method. Because GCM broadcasts the message to the specified Android application (has exact *reg_child_id*) via Intent broadcast, so that application does not need to be running beforehand to receive the message, this saves batter consumption of background service.

4.2.3 System Implementation

In this section, we present our system implementation to ensure a) the real-time location tracking, b) emergency notification functions of CRLTMS.

a) AJAX real-time location tracking. By setting *setInterval* time to intercept *get_update_location*() function in Algorithm 1, application server periodically call *getChildGPS.php* for checking changed values in database then add new longitudes, latitudes data in to queue array through *addLatLng* function. Those updated GPS data will be plotted on embedded Google Maps without refreshing the webpage.

Algorithm 1: *get_update_location*() - Get Updated Location

Input: *reg_child_id*
Output: push *LatLng* values to queue array
begin

```
$.ajax({
url:'getChildGPS.php?reg_child_id='+id,
type:'get',
success:function(res){
try {
res = JSON.parse(res);
console.log(res);
for(i=0; i<res.length; i++){
 var point = res[i];
 addLatLng(point);
}
} catch (ex){}
}
});
```

b) Safe range calculation and emergency notification. The purpose of this step is to find the distance between child's location and a defined center point, this distance is called safe range (e.g. school, playground, home), and when children go out of this range, our system will send the emergency notification to child's device also parent's device. We employ and implement Haversin formula, a well known trigonometric formula to calculate the distance between any two points on the earth. According to [18], given two points with associated latitude ϕ, longitude λ: $A(\phi_1, \lambda_1)$ and $B(\phi_2, \lambda_2)$, the distance d between A and B is calculated as the below equation, where r is earth's radius ($r = 6371km$), assuming all angles are measured in radians.

$$d = 2r \arcsin \sqrt{sin^2(\frac{\phi_2 - \phi_1}{2}) + cos(\phi_1)cos(\phi_2)sin^2(\frac{\lambda_2 - \lambda_1}{2})} \quad (1)$$

Let:

$$a = sin^2(\frac{\phi_2 - \phi_1}{2}) + cos(\phi_1) * cos(\phi_2) * sin^2(\frac{\lambda_2 - \lambda_1}{2}) \quad (2)$$

We have: $d = r * c$, where: $c = 2 * atan2(\sqrt{a}, \sqrt{(1-a)})$ (3)

We implement the Haversin formula in our system to perform emergency task in Algorithm 2 as following.

In Algorithm 2, $lat1$ and $long1$ is defined latitude, longitude of center point while $lat2$ and $long2$ is current location data, which is sent from phone device to the system. Variable r and s is *mean radius* of earth and *safe range* in km, value of a, c and d are calculated as in equation (1), (2) and (3). System checks for the update of location data every 4 seconds, calculate the distance to the center point, compare to safe range radius then give alert message if child's device is out of range.

Algorithm 2: *distanceCalc*() - Haversine Implementation

Input: $A(\phi_1, \lambda_1)$ and $B(\phi_2, \lambda_2)$
Output: distance d, alert message
begin

```
lat1 = 37.242965
long1 = 127.080052 /*LatLong of center point*/
/* define values */
pi80 = M_PI / 180 /*Radian*/
lat1 *= pi80 /*convert to radians*/
long1 *= pi80
lat2 *= pi80
long2 *= pi80
r = 6372.797 /*mean radius of Earth in km*/
s = 0.2 /*safe range radius in km*/
dlat = lat2 − lat1
dlon = long2 − long1
a = sin(dlat/2).sin(dlat/2) +
cos(lat1).cos(lat2).sin(dlon/2).sin(dlon/2)
c = 2 ∗ atan2(sqrt(a), sqrt(1 − a))
d = r ∗ c
if d > s then
    echo "alert"
echo "safe"
```

5. SYSTEM EVALUATION

To evaluate our study, we deploy a PHP website on Apache web server as application server, parents can request for there account, register child's device into system and start tracking child's device via web browser. Besides, an Android application was developed and installed on child's device in order to send location data message from CRLTMS. As seen on Figure 4, our system plots every move of children on the embedded Google Maps through AJAX call without refreshing the page. The blue circle is the defined safe range, pink marker and bluish marker is starting point and current point of child's device respectively. The blue circle is the defined safe range, pink marker and bluish marker is starting point and current point of child's device respectively.

When children go out of safe range, an automated message will be sent from application server to child's device as seen on Figure 5. On the other hand, messages are manipulated fast and properly as well.

5.1 Security and privacy of CRLTMS system

Our CRLTMS system represents strong security characteristics in the combination of clients' devices, web server and cloud environment. Parent's accounts are encoded by MD5 hashing algorithms in MySQL database. While children ID (*reg_child_id*) is an unique string, generated, managed and secured on Google Cloud Platform.

5.2 Performance and battery test

In order to evaluate our study, we setup a test-bed environment for two popular Android mobile push messages: Google GCM and HTTP. For Google GCM side, we use implemented application as above and it is installed on an Android device. The delay time for this architecture is represented in Figure 6. For the HTTP side, we use Comet [19] style long polling as depicted in Figure 7. Basically, the client send a HTTP request to the application server and it is kept open until the server has data to response to the client. Server sends it back to the client and closes the connection between client-server.

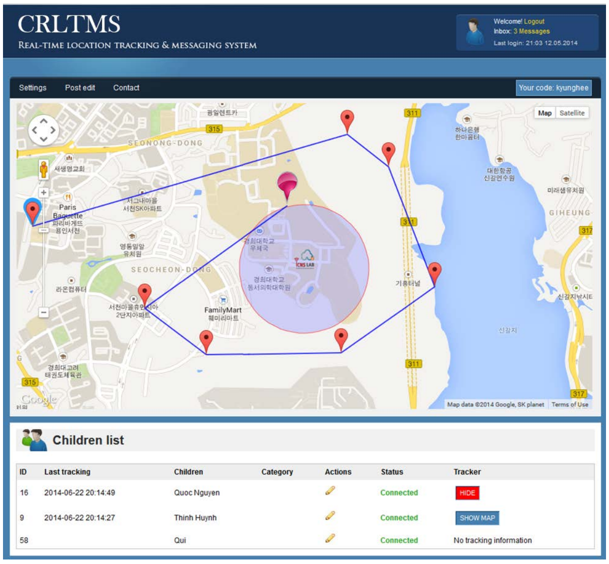

Figure 4: CRLTMS management website

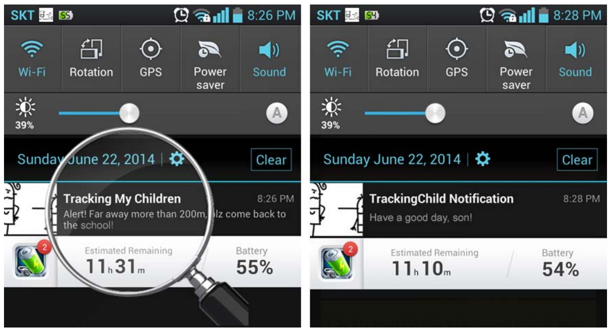

Figure 5: Message notification on mobile

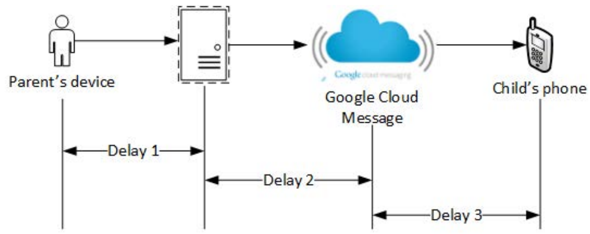

Figure 6: GCM approach delay time

On the other hand, we use PowerTutor [20] to measure the power consumption by major system components such as CPU, network interface, display and GPS receiver. Client's Android device used in this experiment is SKY Vega Iron A870, which has 2150 mAh of battery power. We define two testing plans: 3G mode and Wifi mode, we first set the phone to use 3G connection for the first test then use wifi connection for second test. We start sending 2048 messages (1 byte per message) from parent's devices to client's device and measuring the percentage of used battery in one hour

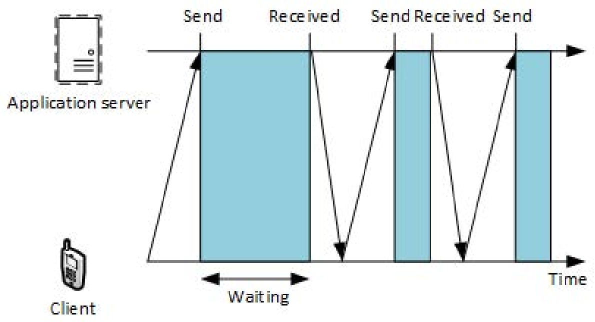

Figure 7: HTTP approach delay time

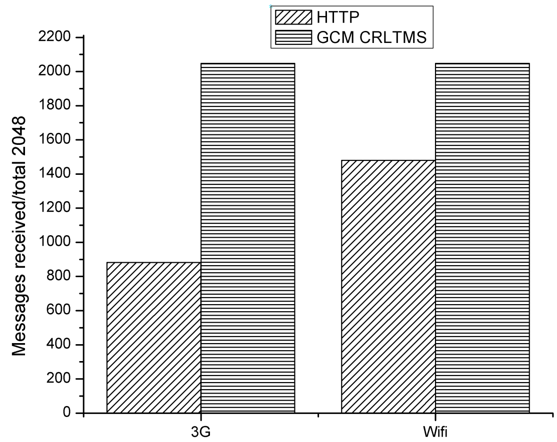

Figure 9: Messages received comparison between HTTP and our GCM CRLTMS approach

and messages received from two testing plans as shown in Table 1.

Table 1: Message pulling and battery consumption

	3G		Wifi	
	HTTP	GCM	HTTP	GCM
% battery/hour	12.15%	9.6%	2.8%	3.14%
Messages sent/hour*	1994	50400	5514	119194
% battery/message	0.006	0.0002	0.0005	0.00003
Messages received**	882	2048	1480	2048

*Messages sent as much as possible.

**Messages received in total 2048 messages sent.

The result shows that by using GCM to manipulate messages, we can send and receive a significant number of messages over two orders of magnitude better than traditional HTTP approach both in 3G mode and Wifi mode. Figure 8 represents the speed of messages were received (averaging only 1994 messages per hour for HTTP approach versus 50400 message for GCM approach).

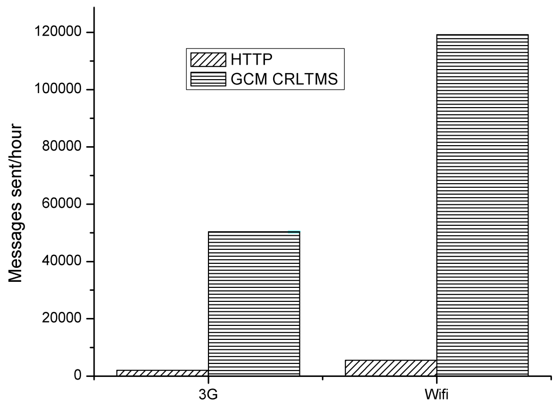

Figure 8: Messages sent comparison between HTTP and our GCM CRLTMS approach

Besides, the experiment shows the reliability of our CRLTMS system, more precisely, the success rate of delivered messages. In 3G mode, all GCM messages are sent, whereas HTTP only handled 882 of 2048 messages in total, the success rate in HTTP mode is 43%. This is caused by message polling mechanism in traditional HTTP approach in Figure 7. In which, a lot of messages are missed in the interval time between when the connection closed with the previous message and subsequently re-established to receive the next message. This rate is mitigated on Wifi, as the connection can be re-established more quickly. The HTTP can handle 1480 of 2048 message as seen in Figure 9. By doing this, our approach can reduce the network latency when sending a specific number of messages to target device.

On the other hand, Figure 10 and Table 1 highlights the efficient in battery consumption of our GCM CRMTMS approach. Both in 3G and Wifi mode, GCM outperforms HTTP by manipulating large amount of message through network and using a small number of battery.

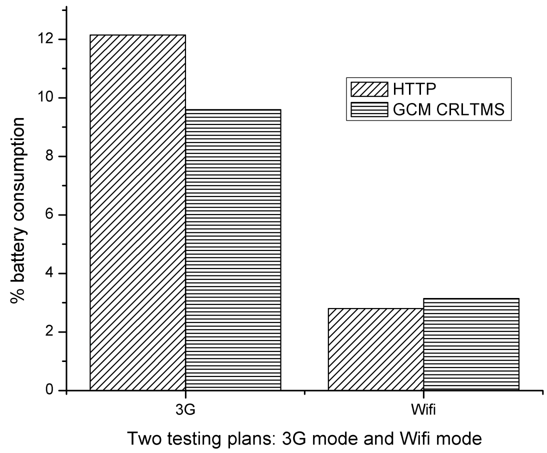

Figure 10: Battery consumption comparison between HTTP and GCM in 3G and Wifi mode

6. CONCLUSIONS

Android is one of the most popular mobile operating system today, it has been attracting more attention to developers and mobile industry. Together with the advent of cloud internet, mobile application has revolutionized and came to every edges of our daily lives such as at work, at school, on the move, entertainment, etc. One of the most important aspect is health care services, traditional method of location tracking, pulling message has been replaced by more advanced method, it comprises cloud service and GPS location data. In this paper, we have proposed our system, namely CRLTMS, developed on Google Cloud Messaging service and it solves two real-time objective problems: a) messaging and b) location tracking, emergency notification. The evaluation shows that CRLTMS can help parent keep track and communicate with their children in real-time demand. Google GCM-based application can reduce the network latency and battery power consumption, and brings better user experience for parents to keep watching their children moves everyday. This study can be extended to adapt and apply to many scenarios such as truck, cargo and logistic tracking, location-based emergency medical service, disaster and critical behavior warning system as future investigations.

7. ACKNOWLEDGMENTS

This research was supported by the MSIP (Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014(H0301-14-1020)) supervised by the NIPA (National IT Industry Promotion Agency) and the ICT R&D program of MSIP/IITP, Republic of Korea. [14-000-05-001, Smart Networking Core Technology Development]. Eui-Nam Huh is the corresponding author.

8. REFERENCES

[1] The Magellan Report. The magellan report on cloud computing for science. *U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR)*, December 2011.

[2] Penghui Li, Yan Chen, Taoying Li, Renyuan Wang, and Junxiong Sun. Implementation of cloud messaging system based on gcm service. In *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*, pages 1509–1512, June 2013.

[3] S.D. Shetty, S. Vadivel, and N. Gandhi. A cloud enabled online quiz application deployed on google cloud. In *Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on*, pages 108–113, Dec 2012.

[4] A Gurek, C. Gur, C. Gurakin, M. Akdeniz, S.K. Metin, and I Korkmaz. An android based home automation system. In *High Capacity Optical Networks and Enabling Technologies (HONET-CNS), 2013 10th International Conference on*, pages 121–125, Dec 2013.

[5] R.C. Das and T. Alam. Location based emergency medical assistance system using openstreetmap. In *Informatics, Electronics Vision (ICIEV), 2014 International Conference on*, pages 1–5, May 2014.

[6] V. Koufi, F. Malamateniou, G. Vassilacopoulos, and A Prentza. An android-enabled mobile framework for ubiquitous access to cloud emergency medical services. In *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*, pages 95–101, Dec 2012.

[7] R.Troncon Saint-Andre etal K.Smith. *XMPP: The Definitive Guide*. First edition, 2009.

[8] Wikipedia. Session initiation protocol. *http://en.wikipedia.org/wiki/Session_Initiation_Protocol*.

[9] Wikipedia. Internet relay chat. *http://en.wikipedia.org/wiki/Internet_Relay_Chat*.

[10] Wikipedia. Xmpp. *http://en.wikipedia.org/wiki/XMPP*.

[11] Wikipedia. Android operating system. *http://en.wikipedia.org/wiki/Android*.

[12] Google Inc. Google cloud messaging for android. *http://developer.android.com/google/gcm/*.

[13] Wikipedia. Apache http server. *http://en.wikipedia.org/wiki/Apache_HTTP_Server*.

[14] Wikipedia. Php. *http://en.wikipedia.org/wiki/PHP*.

[15] MySQL. Mysql. *http://www.mysql.com*.

[16] Wikipedia. Exponential backoff. *http://en.wikipedia.org/wiki/Exponential_backoff*.

[17] James Inman. Law of haversines. *http://en.wikipedia.org/wiki/Haversine_formula*.

[18] R. W. Sinnott. Virtues of the haversine. *Sky and Telescope*, December 1984.

[19] Wikipedia. Comet programming. *http://en.wikipedia.org/wiki/Comet_programming*.

[20] Lide Zhang, B. Tiwana, R.P. Dick, Zhiyun Qian, Z.M. Mao, Zhaoguang Wang, and Lei Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010 IEEE/ACM/IFIP International Conference on*, pages 105–114, Oct 2010.