International Global Navigation Satellite Systems Society
IGNSS Symposium 2015

Outrigger Gold Coast, Australia
14-16 July, 2015

Real-time position tracking via Internet communication protocols in LBS.

Izwan Idris
Queensland University of Technology, Brisbane, Australia
+61 415646549, izwan.idris@hdr.qut.edu.au
Charles Wang
Queensland University of Technology, Brisbane, Australia
+61 731389111, cc.wang@qut.edu.au
Yanming Feng
Queensland University of Technology, Brisbane, Australia
+61 731381926, y.feng@qut.edu.au

ABSTRACT

The most enticing feature of a location based service (LBS) system is the ability to distribute and personalise useful and relevant information to its subscribers with regards to their current spatial location. In order to sustain a responsive and an uninterrupted service, LBS system requires consistent, efficient and reliable network connectivity between the central server and mobile devices. Currently, most of the connectivity is done through wireless mobile Internet connection which is not resistance to known issues such as service outages, limited bandwidth and intermittent signals. The connectivity in LBS is also impacted by the Internet communication protocols used in different applications. LBS system generally requires frequent message transmissions between the subscribers and the central server via Internet communication protocols particularly where real-time position tracking is required. Therefore, it is imperative for LBS applications to adapt to communication protocols that can optimise the connectivity performance particularly in wireless mobile Internet connection. This paper presents a study on the performance of device connectivity in LBS by measuring the latency in data transmission between mobile devices and a central server via Internet communication protocols. The study examines the impact of the different type of Internet communication protocols towards system latency in real-time position tracking in kinematic and static positioning. In kinematic testing, the result shows that all protocols are trending in the same pattern which suggests that all protocols are affected by the same mobility factors such as communication stability and reliability. The communication factor is clearly proven in the benchmark experiment where static test with stable 3G mobile connection shows significant improvements in the average latency due to stability in the communication link.

KEYWORDS: LBS, Decimetre positioning, Communication Protocols

1. INTRODUCTION

Real-time vehicle tracking system is one of the fastest growing and emerging technologies applied in various critical applications such as fleet management systems (FMS), road safety and driver assistance systems, and advance vehicles cooperative systems. These critical systems are also sometime referred as Intelligent Transport System (ITS). These systems are referred to as 'intelligent' because their capabilities allow them to perform higher order operations such as situational analysis and adaptive reasoning (Jonathan Raper, *et al., 2007*). Tracking a vehicle's spatial location in real-time by a central server with high processing power and resource, and access to unlimited information, will allow the server to offer countless informative services to the tracked vehicles. For an example, a vehicle tracking server which has access to the traffic information can efficiently compute and analyse alternative routes for its individual subscribers based on the vehicles real-time location, current traffic congestion and any reported incident, hence promote lower energy consumption and better living lifestyle.

In the context of providing services based on real-time vehicles tracking, Location Based Services (LBS) share the same aspiration as ITS systems. The term LBS refers to an IT service which provides information that has been filtered, selected, compiled, or created, taking into account the current locations of the device, other people, or mobile objects (Brad McKenna, *et al., 2011*). The key technologies that can enable this ITS vision are many of the same technologies that underpin LBS in general: geo-positioning, wireless communications, mobile computing platforms, and spatial databases (Jonathan Raper, *et al., 2007*). Most of the current LBS applications utilise Internet connection in their wireless communication between a server and mobile devices due to the wide area coverage of service, reliable and accessible infrastructures, and more cost effective compare to other means of wireless communications such as satellite systems and radio wave transmission. However, Internet connection suffers considerable amount of latency due to several factors such as network traffic congestion, limited data bandwidth and unsuitable type of communication protocols used in the device connectivity. In wireless network communication, latency is the delay in transmitting data via a wireless connection between a server and a mobile device. High latency in LBS will result in poor responses and feedbacks between a central server and mobile devices. Low latency means short delays, while high latency means long delays (Lehpamer, 2004).

In this paper, a study in the performance and agility of Internet communication between mobile devices and a central LBS server is being carried out. The study measures system latency in device connectivity within the three major Internet communication protocols; HTTP, TCP/IP and UDP. Experiments were conducted using a prototype vehicle monitoring platform utilising Open-Source GPS Tracking System (Open-GTS) as the central server and an on-board vehicle tracking unit which is consisted of a mobile computing unit and a Global Navigation Satellite System (GNSS) positioning system. In kinematic testing, the result shows that all protocols are trending in the same pattern which suggests that all protocols are affected by the same mobility factors such as communication stability and reliability. The communication factor is clearly proven in the benchmark experiment where static test with stable 3G mobile connection shows significant improvements in the average latency due to stability in the communication link.

The rest of this paper is organised as follows. Section 2 discusses on common latency issues in LBS applications and introduces internet communication protocols that are later applied in this study. Then section 3 presents the research methodology applied in this study which includes a prototype design of vehicle monitoring platform utilising a web based GPS tracking services application called Open GPS Tracking System (OpenGTS), a mobile tracking device and a device communication server utilising Internet communication protocols to establish and maintain device connectivity in real-time. Section 4 presents results and analyses from field tests. Finally, section 5 concludes this study.

2. LATENCY AND INTERNET COMMUNICATION PROTOCOLS

The term latency refers to a measureable time interval between a defined starting point and a completion point. In communication, network latency has been defined as the time delay between the sending of a packet until the entire packet is received by the receiver through a medium of network communication. Packet transmission latency is the data transmission time from when the packet enters into the network to the time it reaches the destination (Ming Qu, 2011). Communication is a basic functionality of location based service (Yubin Xu and Xiuwan Chen, 2010). In general, LBS systems require communication for various purposes such as to transmit devices locations to the central LBS server, to receive GNSS positioning corrections and to distribute information to subscribed mobile users. As the number of mobile computing users grows exponentially, LBS solutions have becoming more relevant and adding values to the everyday life of mobile users. Despite their early failure (A. May *et al.*, 2006), location-based services (LBS) are making a comeback due to the emergence of new mobile phones with increased processing power, high-resolution colour screens, faster data connections, high performance positioning technologies, and a greater emphasis by the telecom operators on data services (A. May *et al.*, 2006),.

2.1 Type of latencies

However, the ever increasing demand in LBS poses challenges to LBS players to provide smooth and responsive user experience to the mobile users considering wireless communication suffers high latency issues. Some of the possible causes of high latency in a mobile network are as follows.

- **Network traffic congestion:** Number of concurrent mobile users connecting to a central LBS server will increase latency due to several possible factors such as limited hardware resources, exhausted data bandwidth consumption and network topography design.
- **Limited wireless communication coverage**: Wireless Wide Area Networks (WWAN) covers from 100m to 35 km. This network encompasses GSM/CDMA/UMTS, which is called 2G, and CDMA2000/WCDMA/TD-CDMA, which is so called 3G (). Mobile devices which are located further away from the coverage radius will most likely experience high latency in communication.
- **Interrupted connection**: Intermittent signal in wireless communication can cause high latency between mobile devices and the central server due to the frequent re-establishment process in device connectivity.
- **Network communication protocols**: Mobile devices have several ways or methods in communicating with the central servers or LBS applications via wireless communication. These methods of communication are called communication

protocols. It is imperative for LBS systems architects and developers to identify the nature of their solutions and understand the behaviour of each protocol as different communication protocols are designed for specific types of solutions. Choosing inappropriate communication protocols may lead to unnecessary latency in device connectivity.

2.2 Network communication protocols

Hypertext Transfer Protocol (HTTP) is a web application protocol which is mainly designed to ease communication between a client and a web application hosted by a server. Although HTTP would be the simplest way for a client to communicate with the server, HTTP may introduce unnecessary latency due to its interfacing process with the underlying and reliable transport protocol called Transmission Control Protocol (TCP). HTTP can also use unreliable transport protocol such as User Datagram Protocol (UDP).

Transmission Control Protocol (TCP) offers reliable, ordered, and error-checked delivery of a stream of data between a server and clients (Ming Qu, 2011). Unlike HTTP, TCP is a low level communication protocol which provides raw socket connection between a server and a client. Each of the TCP packets consists of a TCP header of length 20 Bytes, as well as a TCP payload. The TCP header contains information necessary to guarantee data packet can be transmitted to the destination, while the TCP payload contains the information to be delivered to the destination. TCP requires connection to be established via a three-way handshake before sending and receiving streams of data. Furthermore, TCP has flow control capability which manages the rate of streaming data and it also ensures arrival of all sent data by re-transmitting lost packet.

Another widely used low level communication protocol is called User Datagram Protocol (UDP). Similar to TCP, UDP offers raw socket connection between a server and a client. But unlike TCP, UDP does not require a proper connection to be established before a packet can be sent from a server to a client. Furthermore, UDP header doesn't have a sequence number, acknowledgement number or flags in comparison with TCP. Apart from a smaller 8 Bytes packet size compare to TCP's 20 Bytes packet, UDP only uses the length to indicate the size of the entire datagram and the checksum to verify the header data. UDP's main concern is the speed of sending the packet rather than securing the packet or retransmitting any lost packet. Information on the TCP and UDP header structures is shown in Figure 1.

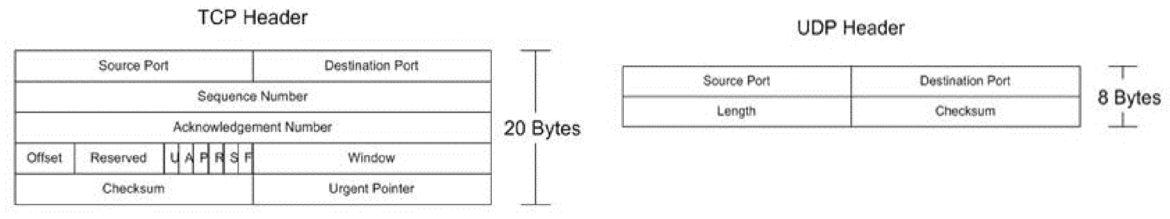

Figure 1. TCP and UDP header structures

3. RESEARCH METHODOLOGY

In order to study the performance of each communication protocol, a prototype of vehicle monitoring system has been developed where real-time positions of vehicles are tracked from a remote server. The term real-time refers to the concept where vehicle positions will be updated at a certain update frequency, for example 1 Hz and only the latest position will be sent to the monitoring server. Under any circumstance that prevents a position to update, such as Internet connection dropout or device malfunctions, the stale positions will not be consolidated or replicated back to the central server. Though data replication is quite a useful feature to ensure a complete tracking history of a vehicle, this is not the focus for this study. The main objective for this vehicle monitoring system is to measure latency in kinematic position updates via Internet communication protocols using mobile Internet connection (3G) and to benchmark the results against static positioning and land-line based Internet connection (ADSL).

3.1 Development of Vehicle Monitoring Platform

The monitoring platform consists of two main components; **monitoring server** and **vehicle tracking unit**. The monitoring server establishes communication with a vehicle tracking unit via internet communication protocols such as HTTP, TCP and UDP. In order to do relative comparisons of the latency results between these communication protocols, every update on the vehicle's position is sent to the central monitoring server via all protocols consecutively. Each communication protocol will be individually time-stamp before data is sent from the tracking unit and after data is received at the monitoring server. Details on how the latency is calculated for each protocol is further explained in section 3.5, Field test campaigns. Figure 2 below shows the overview architecture of the vehicle monitoring platform.

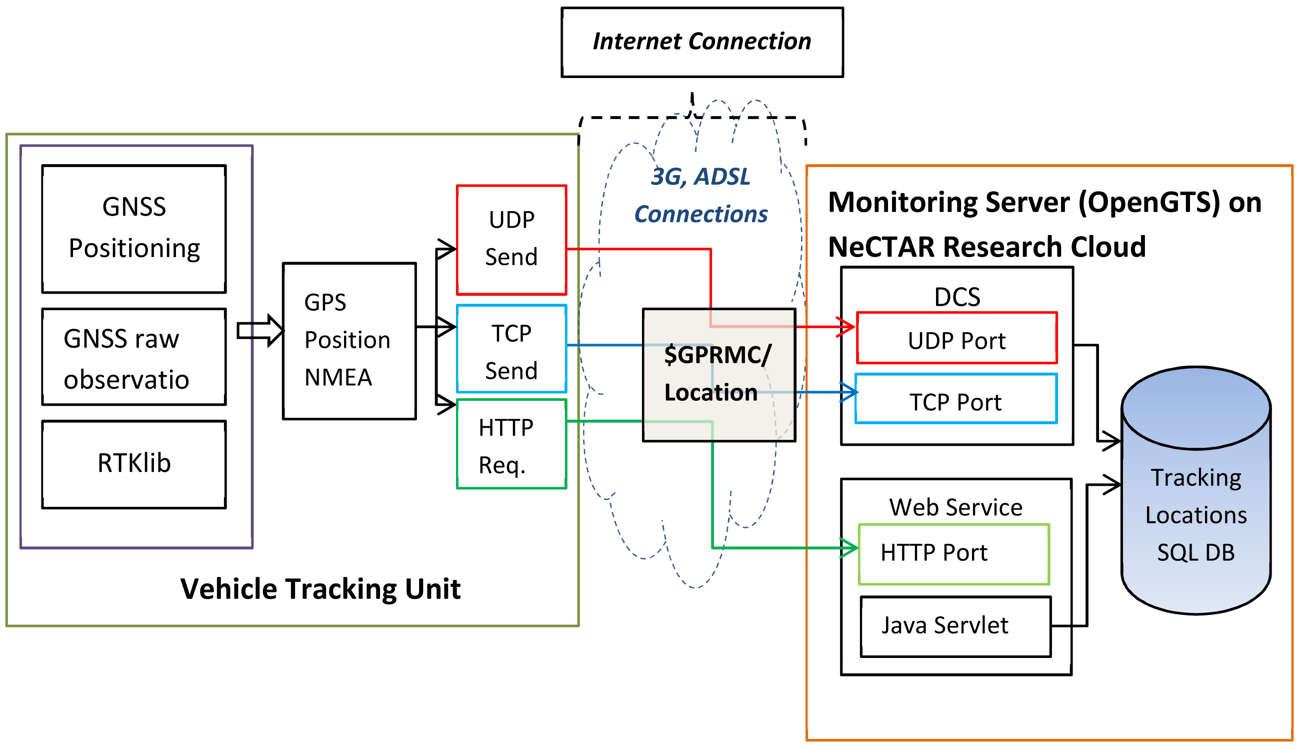

Figure 2. Vehicle Monitoring Platform Architecture

3.2 Monitoring server: Open GPS Tracking System (OpenGTS)

The server component of the monitoring system was implemented using Open GPS Tracking System (OpenGTS) which is an open source project designed specifically to provide web-based GPS tracking services for a fleet of vehicles. It is designed to operate independently of any specific GPS tracking device or protocol, but comes with support for several device protocol formats. The monitoring server was deployed to a research cloud, the National eResearch Collaboration Tools and Resources (NeCTAR), on a virtual Ubuntu machine with 1 virtual CPU and 4GB of RAM.

3.2.1 How OpenGTS works

OpenGTS maintains a simple process flow on the web-server to track fleets of vehicles in real-time. Web-server in OpenGTS receives National Marine Electronics Association (NMEA) messages from remote devices via communication protocols and each message is converted into spatial location before being saved in the SQL database. On the server side, OpenGTS is designed to be device and protocol independent. In order to use the features of OpenGTS, a specific device/protocol communication server will need to be implemented to communicate with the remote device and place the data in the SQL database. OpenGTS ships with support for Open Source Device Monitoring and Tracking Protocol (OpenDMTP, http://www.opendmtp.org) so that OpenDMTP compliant devices will be ready to immediately utilize the services of OpenGTS. Furthermore, the device communication server can be customised to support other devices which are not OpenDMTP compliant. On the web-interface side, the user presentation is easily customizable to fit the individual desired motif. Figure 3 shows an example of the web-interface in OpenGTS.

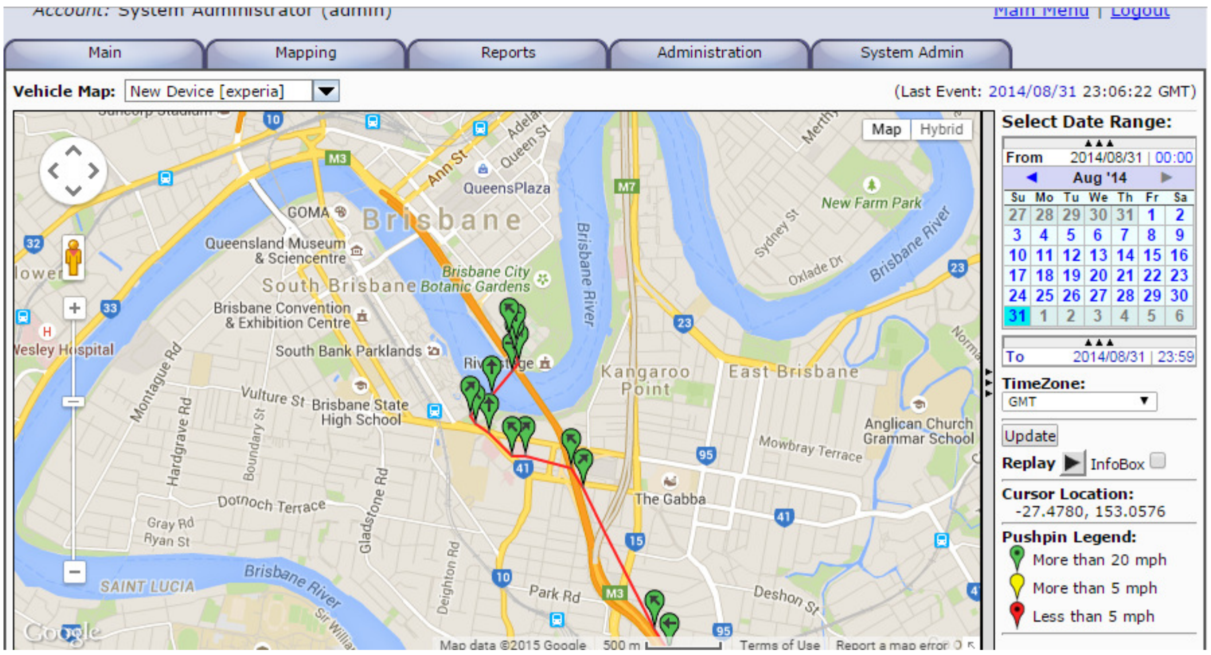

Figure 3. OpenGTS's user web-interface

3.2.1 Supported Platforms

OpenGTS is completely implemented in Java and should run fine on any system that fully supports Java Runtime Environment. As OpenGTS's implementation requires a SQL database server, the chosen platform is limited to the compatibility with the chosen SQL database server. In this study, MySQL has been chosen as the SQL database server and the server operating system is Ubuntu version 14.04. MySQL is also known to be compatible with most of the major operating systems (OS) such as Linux, Mac OS X, FreeBSD, OpenBSD, and Windows-XP/Vista/20xx platforms.

3.2.2 System Architecture

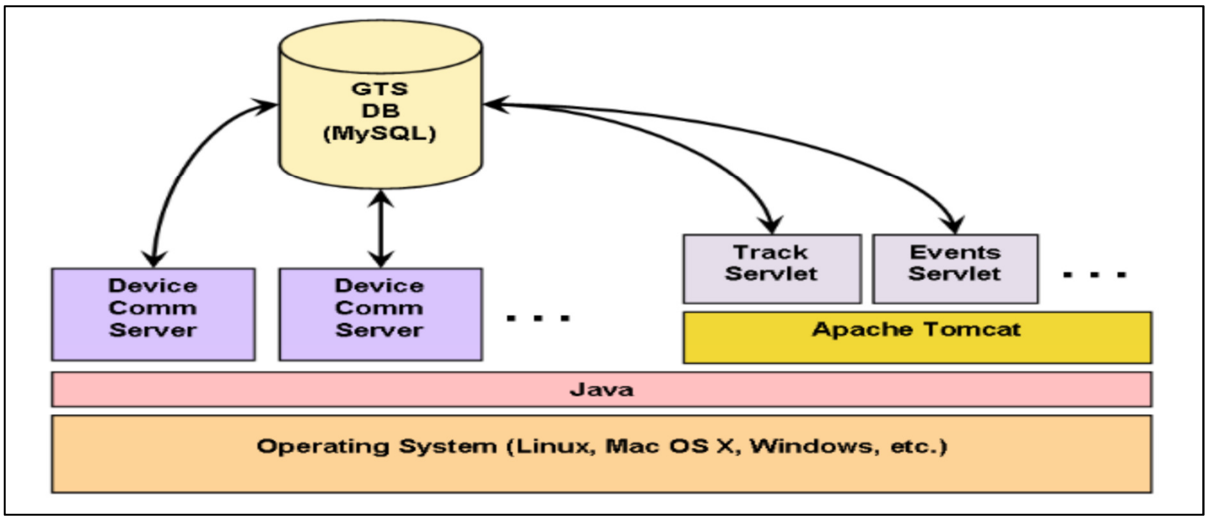

Figure 4. OpenGTS's system architecture

Figure 4 above describes the basic system architecture of the OpenGTS system. The various device communication servers (the modules which listen for incoming data from the remote GPS tracking devices) run as separate processes on top of Java. The Track servlet, as well as other servlets (including any HTTP-based device communication server), run within a Servlet Container, such as Apache Tomcat.

3.2.2 Device Communication Server (DCS)

In order for OpenGTS to receive data from a device, a customised Device Communication Server (DCS) will need to be implemented that understands the protocol used to communicate with the remote device, and insert received events into the SQL database. The method used by remote devices to transport events to the server varies greatly with the manufactures of the device. Some transport data to a server via SMS messages, some use an SMTP email transport to send data to a server, some use an HTTP-based protocol which encode data in the request to the server, and many use form of raw-socket based communication (via TCP/UDP) to connect to a listener on the server to transmit data. In order to create a device communication server that is able to parse incoming data from a device, an intimate understanding of the specifics of the protocol used by the device manufacturer is required. In this study, a new DCS has been developed to parse incoming data from the vehicle tracking unit and the new DCS supports HTTP, TCP and a UDP communication protocols.

3.2.3 HTTP based Device Communication Server - OpenGTS

HTTP based communication is typically the easiest to implement as it only requires configuring message that will be embedded in the HTTP request message to the "gprmc" servlet. The servlet name refers to one of the message type in NMEA 3.0 format. Here is an example of a HTTP request message sent to "gprmc" servlet to update device's location:
"http://localhost:8080/gprmc/Data?id=1&code=0xF020&gprmc=$GPRMC,132832.00,A,2743.2205301,S,15312.1515650,E,0.00,0.00,050814,0.0,E,D*2E".
The "gprmc" message represents the NMEA-0183 $GPRMC record straight from the GPS receiver which will be converted to location on map. Refer to table 1 for a complete list of NMEA sentences.

NMEA Sentence	Details
$GPBOD	Bearing, origin to destination
$GPBWC	Bearing and distance to waypoint, great circle
$GPGGA	Global Positioning System Fix Data
$GPGLL	Geographic position, latitude / longitude
$GPGSA	GPS DOP and active satellites
$GPGSV	GPS Satellites in view
$GPHDT	Heading, True
$GPR00	List of waypoints in currently active route
$GPRMA	Recommended minimum specific Loran-C data
$GPRMB	Recommended minimum navigation info
$GPRMC	Recommended minimum specific GPS/Transit data
$GPRTE	Routes
$GPTRF	Transit Fix Data
$GPSTN	Multiple Data ID
$GPVBW	Dual Ground / Water Speed
$GPVTG	Track made good and ground speed
$GPWPL	Waypoint location
$GPXTE	Cross- track error, Measured
$GPZDA	Date & Time

Table 1. NMEA Sentence Information (excerpted from http://aprs.gids.nl/nmea)

3.2.4 Raw-socket based Device Communication Server - OpenGTS

OpenGTS provides an example template server which can be modified to parse received data and insert into the SQL database. This server type runs as a separate process listening on a selected socket port for incoming TCP/UDP connections. Similar to the HTTP-based communication, $GPRMC message and other information will be packaged and sent to a listening port and the message will be parsed and inserted to the SQL database. Both of the TCP and UDP ports are assigned with different port number as they are required to listen to the incoming data concurrently.

3.3 Vehicle tracking unit

The main objective of having a dedicated vehicle tracking unit is to customise the tracking unit with the requirements of this study. The tracking unit is required to calculate its own position and consecutively send each of the fixed position to OpenGTS via HTTP, TCP and UDP communication protocols.

3.3.1 Vehicle tracking unit (Hardware): Beaglebone Black, U-Blox receiver and antenna

The main hardware components for the tracking unit are mobile computing platform, and GNSS antenna and receiver. The hardware specifications are Beaglebone Black with 1GHz ARM@ Cortex–A8 processor, 512 MB RAM, IO interfaces and 32 bit micro-controller.

Beaglebone Black is a small size computing board with excellent computing power, low power consumption and good size storage, which is a perfect choice not only for computation purposes but also for vehicle mechanical assembly. The board does not have an on-board network card or 3G Modem, but it does have a USB port where an external 3G Modem or WIFI dongle can be plugged-in to the board. As for the positioning component, the compact GNSS module of U-BLOX –NEQ-M8N receiver and U-BLOX GPS antenna is sufficient enough to do single positioning for the tracking unit.

3.3.2 Vehicle tracking unit (Embedded Software): GNSS positioning solution and inter-process communication programming.

The main software requirement for the tracking unit is an embedded software application to process GNSS observation data and positioning corrections into standard and precise positioning algorithms. Then, every positioning solution is sent to the central monitoring server via Internet communication protocols. RTKlib is open-source software that provides standard positioning solution and precise real-time positioning solutions such as DGPS, RTK and PPP. RTKlib supports large database of GNSS receiver formats and I/O data communication protocols (for example serial port). RTKlib was configured to use observation data from U-BLOX's receiver via serial interface and the positioning solution was set to single positioning.

In order to send or update positions to the remote server via network communication protocols, inter-process communication between distributed machines was established. HTTP request was being sent to the server using UNIX's CURL command; and both TCP and UDP were using inter-process communication programming provided by the standard C++ library in UNIX.

In order to simplify the integration between GNSS positioning and positions updates to a tracking server, the on-board's RTKlib has been extended with OpenGTS options. The idea is that, a single instance of an RTKlib-based application such as RTKRCV can send each positioning solution via NMEA's GPRMC format message to OpenGTS without involving complex data interfacing and convoluted implementations which could potentially cause more delays or unnecessary latencies in the system. GPRMC is one of the sentences in NMEA which contains the minimum recommended data on position information or PVT (position, velocity, time). Refer to table 1 for a complete list of NMEA sentences.

Example of additional OpenGTS options in the RTKlib's configuration file:

```
ogts-enable    = 1                    # to enable OpenGTS location updates
ogts-deviceid  = test02               # to identify the device in the OpenGTS server
ogts-acctid    = sysadmin             # to identify the receiver
ogts-hostip    =  130.56.549.144      #OpenGTS's IP address
ogts-portnum   = 8080                 # listening port
ogts-interval  = 5                    # Position updates frequency
```

3.5 Field Campaigns

As mentioned in the section 3, the main objective for this vehicle monitoring system is to measure latency in kinematic positioning mode where vehicle's position updates are done via Internet communication protocols using mobile Internet connection (3G). The experiment data are expected to show how various communication protocols perform with regards to latency when a vehicle is in high mobility and using mobile Internet connection (3G) to update its position to the central server. The secondary objective for these experiments is to study the impact of mobility and Internet connection bandwidth towards latency. Therefore, kinematic positioning is benchmarked against static positioning and mobile Internet connection is benchmarked against land-line based connection.

3.5.1 Method to calculate latency

Figure 5 below shows the process that was used in the experiment to measure latency during real-time location update by a vehicle tracking unit to the OpenGTS server. **This method does not measure a network's ping time between the tracking unit and the monitoring server, rather to measure the time that it takes to update its location or position at the central tracking database via network communication**. Each time a mobile device send a packet to update its current location via any of the communication protocol, it will include a time-stamp (**Time-Stamp #1**) taken prior of commencing the packet sending process. Once the packet is completely received by the server, a second time-stamp (**Time-Stamp #2**) is taken. Latency for each location update is calculated by the difference between the second and the first time-stamp. In order to reduce any possible variation in the network communication or system environment, latency for each communication protocol is measured using the same GPS position consecutively.

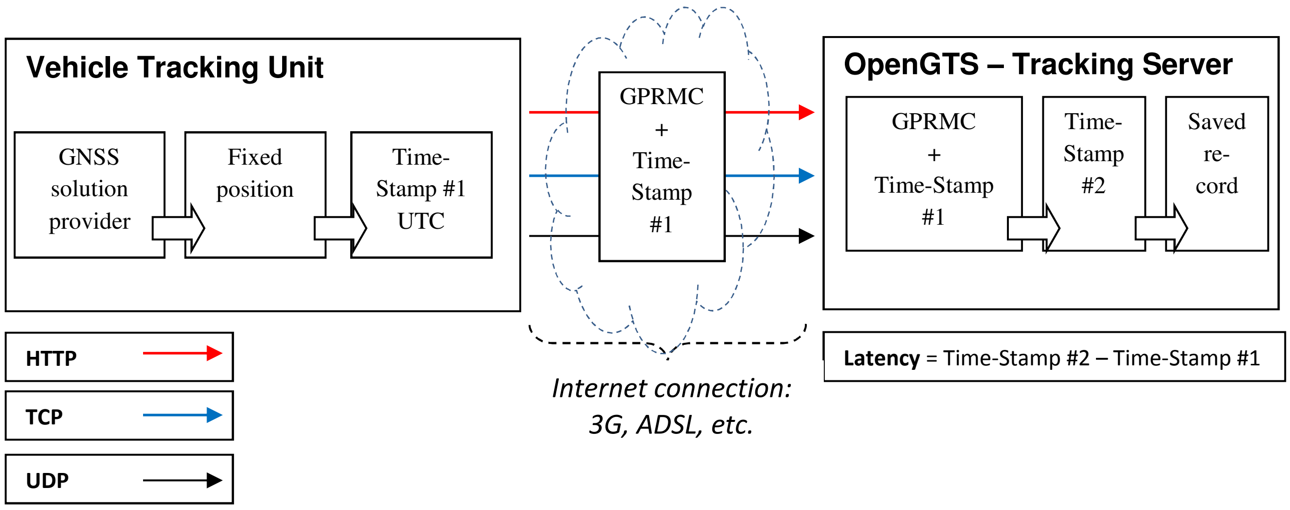

Figure 5. Methods in calculating latency

3.5.2 Field tests setups

There were 3 experiments executed during the field test exercise. Some details of the experiments are shown in table 2. In order to capture high timing precision and to improve visibility in the benchmarks, latency was measured in milliseconds. The kinematic experiment (Test #1) was done using single positioning as high precision solution is not the main focus in this study. Therefore, the experiment's route does not have to take into account location of correction base stations. The estimated total distance of the entire route is about 30 km which covers mostly rural areas. The complete route is shown in Figure 6. Both Test #2 and Test #3 were done in an open area using 3G mobile Internet connection and Asymetric Digital Subscriber Line, ADSL respectively.

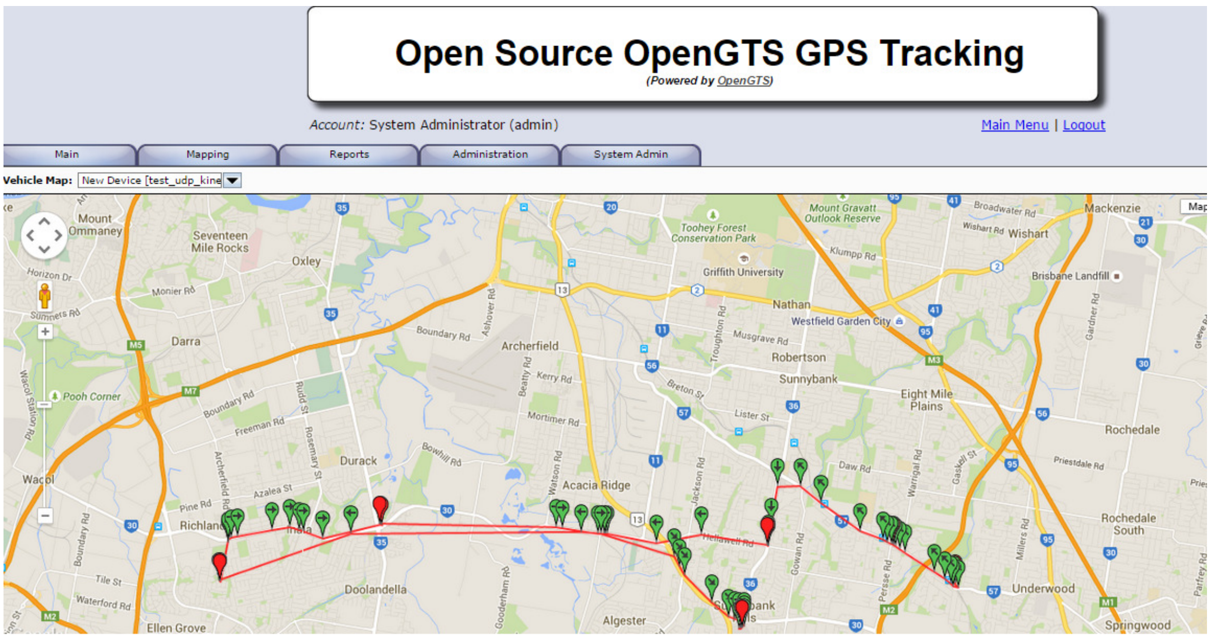
Open Source OpenGTS GPS Tracking
(Powered by OpenGTS)
Account: System Administrator (admin)
Main Menu | Logout
Main
Mapping
Reports
Administration
System Admin
Vehicle Map:
New Device [test_udp_kine

Figure 6. Kinematic experiment route.

	Internet Connection Type		Fix Positioning Type		Objective
Field Tests	3G	ADSL	Static	Kinematic	
Test #1	X			X	Measure latency for kinematic positioning
Test #2	X		X		Measure latency to benchmark kinematic positioning against static positioning
Test #3		X	X		Measure latency to benchmark 3G Internet connection against ADSL connection

Table 2. Experimental Setups

4. FIELD TEST RESULTS

Latency results were calculated by the monitoring server and recorded in OpenGTS's database tables. Figure 7 shows the plot for latencies during kinematic testing for HTTP, TCP and UDP communication protocols. In general, the data trends for all three protocols are quite consistent throughout the timeline, which suggests that HTTP, TCP and UDP protocols are affected by the same factors such as speed of mobility and Internet connection bandwidth. In average, UDP yielded lower latency than HTTP by 46% and lower by 31% than TCP protocols. The lower latency in UDP is consistent with the fact that UDP protocol is not a connection oriented communication protocol where UDP does not have to do three ways acknowledgment and data retransmission as compare to TCP protocol. Although HTTP is a TCP based communication protocol, the average latency in HTTP is noticeably higher than TCP approximately by 22%. HTTP is implemented at web application layer with extra overhead on the application interface whilst both UDP and TCP are transport protocol layers. The overheads in HTTP are mainly introduced to make web communication simpler and easier to use by users.

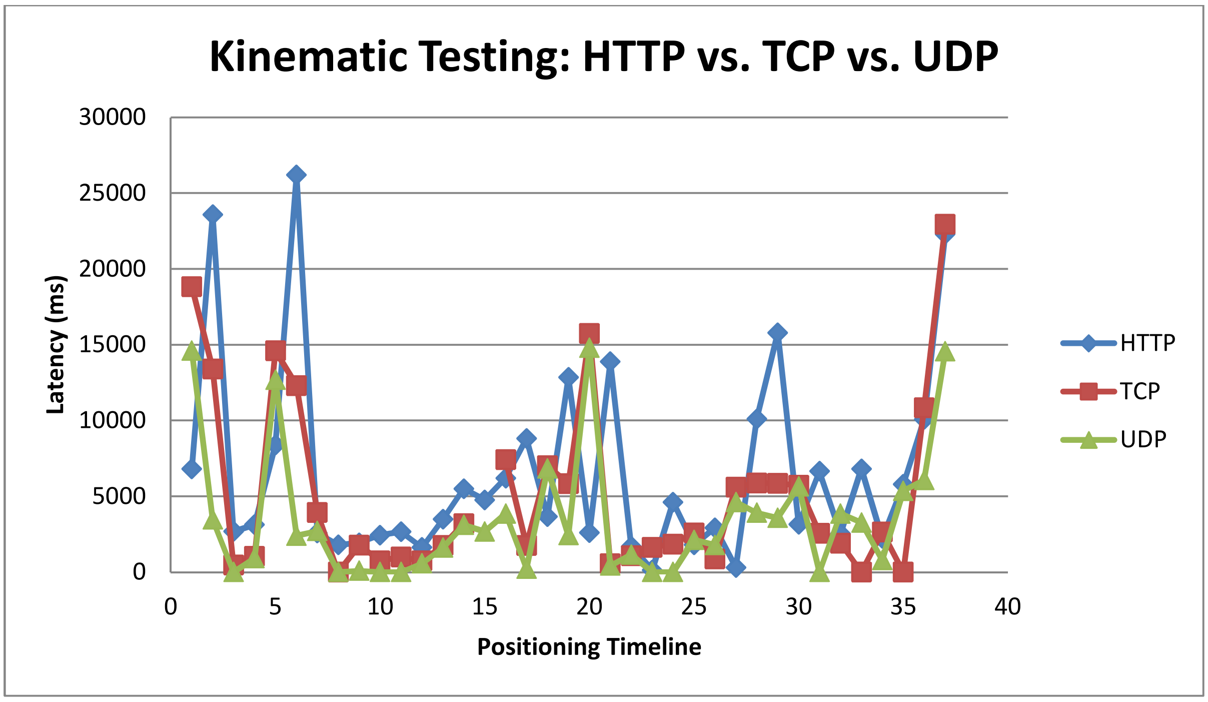

Figure 7. Latency measured in kinematic and 3G mobile Internet connection.

In order to analyse the impact of mobility towards latency in the monitoring system, the latency results from the kinematic experiment is benchmarked against latency results in static position. Table 3 below shows the numerical results from both static and kinematic experiments where the benchmark column is the percentage difference between the two positioning types. All three protocols show an average of 96% of improvement in the latency when positioning is switched to static. Figure 8 clearly shows the significant gap between the positioning methods and the main contributor to the result is suspected to be the inconsistent Internet coverage when the vehicle is in mobile.

	Positioning Scenarios		**Benchmark**
	Average Latency (milliseconds)		
Protocols	**Static**	**Kinematic**	**%**
HTTP	374.4123	6555.216	94.29
TCP	154.0997	5115.25	96.99
UDP	40.6463	3515.568	98.84

Table 3. Latency benchmarks between static and kinematic positioning.

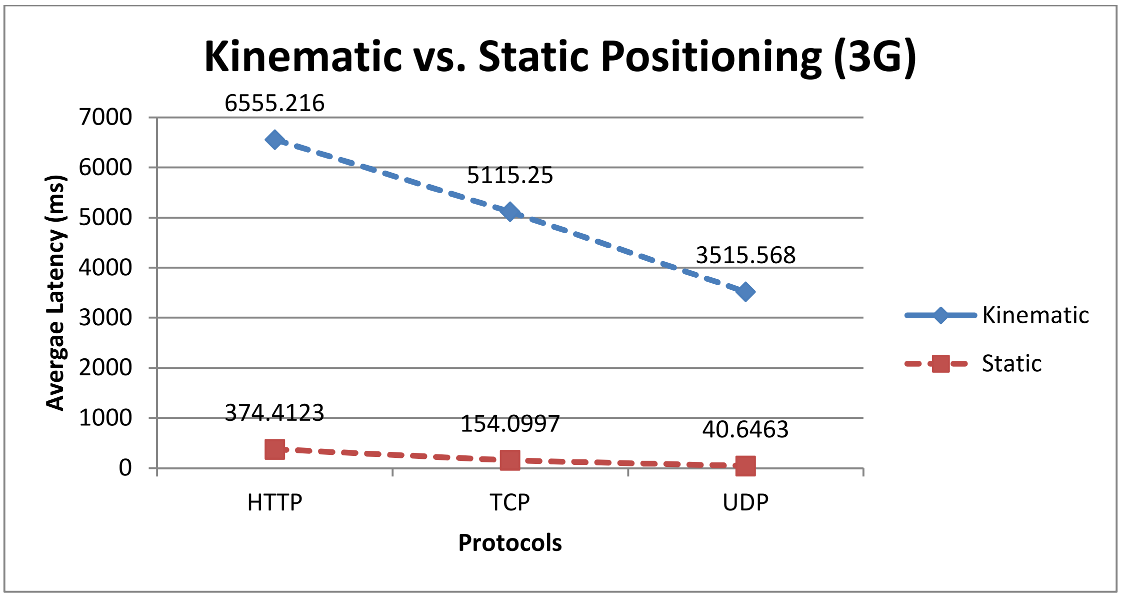

Figure 8. Latency measured in kinematic is benchmarked against static positioning.

Table 4 below shows benchmark results between a land-line based connection (ADSL) and a mobile Internet connection (3G) where both TCP and UDP protocols yielded only small differences in latencies across the connection types. Figure 9 shows how the differences in latencies start to converge from HTTP to UDP protocols. This result implies that in static positioning, the difference in latencies for both UDP and TCP protocols are negligible whether in land-line based connection or in mobile connection. Although, the average upload speed for Asymetric Digital Subscriber Line, ADSL is about 1.3 Mbit/s which is considerably higher than the average upload speed for 3G mobile connection which is about 0.45 Mbit/s, small packets sent to the server to update location does not seems to cause noticeable difference in latency between ADSL and 3G.

	Internet Connection Types		**Benchmark**
	Average Latency (milliseconds)		
Protocols	**3G**	**ADSL**	**%**
HTTP	374.4123	135.4446	63.82
TCP	154.0997	115.1965	25.25
UDP	40.6463	34.2976	15.62

Table 4. Latency benchmarks between 3G and ADSL connection.

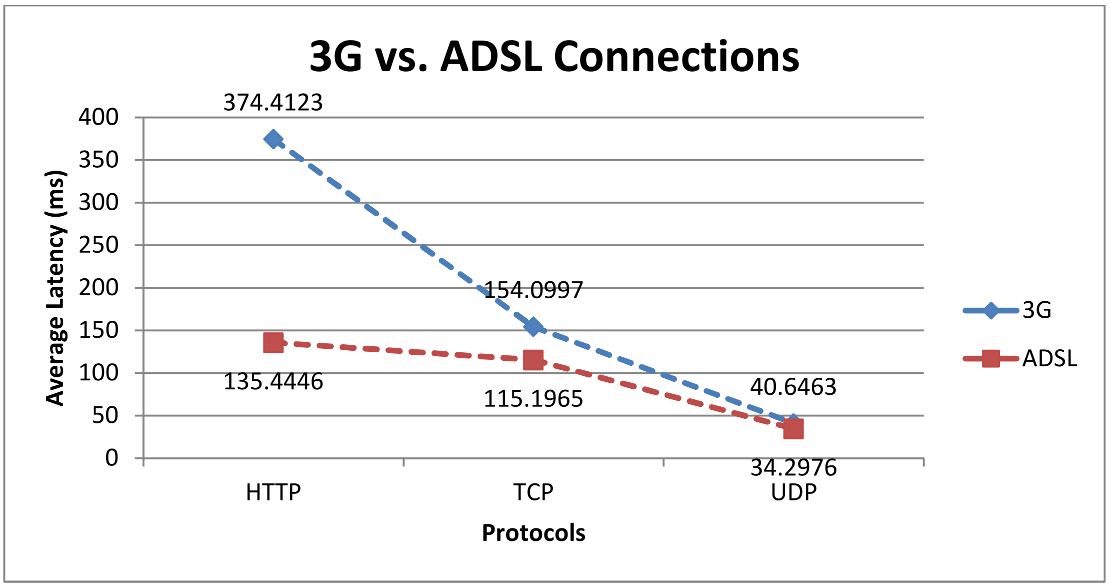

Figure 9. Latency measured in 3G connection is benchmarked against ADSL

5. CONCLUSIONS

Real-time position tracking is one of the key enablers in LBS solutions. In an ITS critical system, quick decisions and responses are highly essential in order to prevent any possible road mishap such as vehicle collisions, dread-lock traffics and road closures. This study develops a prototype of a vehicle monitoring system primarily to study the performance of real-time position tracking by measuring latency in the tracking system.

From the kinematic testing, UDP protocol yielded the lowest average latency as compare to both TCP and HTTP communication protocols. However, UDP is lacking in data reliability and security as compare to TCP based communication. Therefore, despite faster location updates, UDP will not retransmit lost data or secure data like in TCP based communication. From the kinematic and static benchmark experiment, the staggering improvements in latency across all communication protocols shows that reliability in communication link especially with 3G mobile connection is the main factor for lower latency. On the other hand, higher communication bandwidth does not seem to contribute significant differences towards latency in remote position updates. This is mainly because monitoring system requires small network packet to update location or other information such velocity, but the frequency of location update is high. Therefore stability and reliability in network communication even with small communication bandwidth will allow more successful position updates with lower latency.

As conclusions, this study provides vertical and horizontal comparisons on the performance of real-time vehicles tracking in a simple vehicle monitoring system. Reliability and stability of communication link is a known issue for high mobility vehicles as addressed by other research publications. However, depending on the solution requirements, high latency in high mobility vehicles can be reduced by using certain type of communication protocol as long as the implication is fully understood.

6. REFERENCES

Jonathan Raper, George Gartner, Hassan Karim and Chris Rizos. “Application of location-based services”. Journal of Location Based Services, Vol. 1, No. 2, June 2007, 89-111.

Yubin Xu and Xiuwan Chen (2010). LBS Based Disaster and Emergency Management.

A. May, S. H. Bayer, and T. Ross. “A survey of young social and professional users of location-based services in the UK”. Journal of Location Based Services, vol. 1, pp. 112-132, 207.

Brad McKenna, Tuure Tuunanen and Lesley Gardner. “Exploration of Location-Based Services Adoption.”. Proceedings of the 44th Hawaii International Conference on System Sciences 2011.

Ming Qu, Charles Wang and Yanming Feng. “Studies of effects of wireless communication on GNSS positioning performance in a high-mobility vehicle environment”. International Global Navigation Satellite Systems Society IGNSS Symposium 2011.

Lehpamer, H. (2004). Microwave transmission networks: planning, design, and deployment, McGraw-Hill Professional.

OpenGTS Configuration and Installation Manual. GeoTelematic Solutions, Inc.