A Demand-Driven Architecture for Web-Based Tracking Systems

BING-FEI WU, YING-HAN CHEN, and PO-CHIA HUANG
Department of Electrical Engineering
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu
TAIWAN
winand@cssp.cn.nctu.edu.tw http://cssplab.cn.nctu.edu.tw

Abstract: - With the convergence of the Internet and the Global Positioning System (GPS), a common approach called GPS tracking system has been developed nowadays. A GPS tracking system uses GPS to determine the precise location of a vehicle, person, or other assets and shares information with remote clients through the Internet. In the outdoors, the device usually connects to the Internet through mobile networks like GPRS or 3G. However, the weaknesses of a mobile network are low bandwidth and high charging strategies compared to Ethernet or Wi-Fi. In this paper, a demand-driven architecture for web-based tracking systems is presented. By integrating several technologies nowadays, this architecture is designed to improve the efficiency for the location-aware applications. The proposed architecture focuses on two problems. One is the redundant connection, causing the waste of bandwidth and fees during the connection period in the mobile network. The other is the data asynchronization during the period of transmission. We not only addressed the connection scheme to solve above issues, but also designed a prototype on an embedded platform for web-based tracking systems. The performance has also been evaluated for the total latency time when a user exploits different communication medium, and the results showed that the user can monitor the status of target in real time even in the worst case.

Key-Words: - GPS, redundant connections, asynchronization, tracking

1 Introduction

Recently, Location-Based Services (LBS) have become one of the fastest-growing areas with enormous modern devices capable of Global Positioning System (GPS). By providing information and services depending on the location of users or mobile objects, various applications have been developed to benefit people in many situations. Numerous studies and experiments have investigated the potential of LBS in fleet management, vehicle monitoring, individual life-log, tourism information, navigation system and other areas [1-6]. In addition, a tracking or monitoring system in LBS is one of the most common applications used for locating any interested people or items. An intelligent tracking system [7] have been developed which can provide extremely accurate location information for some specific persons. In [8], the user with a mobile phone can upload its current position to a server and obtain the specific information from the server with a query text. Goo-Tracking [9] uses an embedded device with a GPS/GPRS module as a client to periodically transmit its location to a server and display the location using Google Earth software. To assist people to locate their family and friends, a client server system designed for a mobile application has been proposed [10].

However, the connection architecture based on the traditional client-server model has some shortcomings. Under this model, a client has to periodically send its location to a server because the server needs to maintain the latest location of the client to provide real-time information. Therefore, if the server desires a truer location, the client has to increase its transmission rate. In a practical application, the client is usually a mobile target with a GPS-equipped device to provide its location in the outdoor environment. It is reasonable for a mobile device to be powered by batteries and connected to the Internet via the mobile network such as GPRS and 3G. Thus, frequent transmission of unnecessary information not only quickly drains energy from the mobile device’s battery, but also occupies the limited mobile network resource. Furthermore, high communication costs on the mobile network could decrease the willingness of using this service for the user. To solve the problems mentioned above, several methods for reducing location updates to a server while maintaining the real-time responses have been proposed.

To reduce the number of GPS fix transmissions to a server and the position recalculation on a mobile device, the Critical Point (CP) algorithm and the location-aware state machine is designed in [11]. By the dynamic management with these two designs, the mobile phone can reduce the amount of transmission costs and extend the usage life of a battery. Different from the improvement on the mobile site, [12] has proposed a solution based on Ajax push pattern using on the server site. By utilizing this technology, the connection between the server and the client is kept alive with this persistent connection feature in HTTP/1.1. Therefore, once the latest location is uploaded by the client, the server can do the response actively to the browser without polling as traditional methods.

Although both methods provide solutions on the mobile site and the server site, they still use the traditional connection scheme, that is to say, the client has to send the location to the server first and the delay of the real position is decided by the update period. In brief, using the traditional approach to develop a web-based GPS tracking system has two following drawbacks.

1) Redundant connection–Each client has to upload its location periodically to a server for viewers to monitor the latest location. However, it is difficult to define an ideal interval to avoid redundant connection because the client even does not know when viewers will connect to the server to request the client's position.

2) Data asynchronization–There are two intervals in the traditional data connection scheme. One is the interval that the client uploads its location to the server, and the other is the one that the viewer requests the location from the server. Both intervals have to be set and either restricts the location update rate for the viewer.

We extend our previous work [13] and propose a demand-driven architecture for web-based tracking systems to solve the above two drawbacks. The main contribution of this design is as follows:

- Provide the mobile site full controllability of location sharing and connection permission.
- Provide the viewer site full controllability of monitoring interval.
- Provide the server site flexibility to coordinate the mobile site and viewer site with efficient data transmission.

In the traditional connection scheme, the above features are difficult to provide without facing the mentioned drawbacks. On the other hand, if an application has above requirement, e.g., a web-based tracking system, the proposed method is a serviceable solution.

Three major roles appearing in our scheme have to be defined first. One is the target who owns a tracking device used to provide the location of mobile site. Another is a remote server providing the location received from the target. Last is the viewer, who is a user monitoring the target's location through general browsers. The concept of our scheme is simple. The tracking device only sends the location to the remote server when the viewer requests. In other words, the tracking device does not actively and periodically upload the location to the server, but passively wait for the request instead. Hence, the redundant connection is eliminated because every data becomes meaningful. The data asynchronization does not occur because the process only relates to the transmission delay time. The remainder of the paper is organized as follows. Section 2 addresses the detail of the designed system architecture and subsystems. The prototype and demonstration are presented in Section 3. Finally, a brief conclusion and possible extensions are given in Section 4.

2 System Architecture

The proposed connection scheme aims to prevent the drawbacks of redundant connection and data asynchronization by creating a direct channel between the viewer and the target. The system architecture designed to perform this connection scheme consists of three subsystems: central management subsystem (CMS), mobile tracking subsystem (MTS), and web monitor subsystem (WMS). Three subsystems firstly abstracted as specified below and then the detail functional components inside each subsystem are addressed in the following subsection.

CMS interfaces with MTS and WMS to process and transfer the request and response between them. It stores the targets' profiles, locations and status into the database to provide the authorized connection services, location cache services and message delivery services. It not only accepts the registration of network status from MTS and receives requests from WMS, but packages the data for communication.

MTS provides the target's location information on demand to CMS. To save the bandwidth and ease the connectivity, standard HTTP connection protocol is used for data transmission. MTS processes the GPS data to obtain the location information, and packages the response content with the standard HTTP header. It supports the auto-registration mechanism when the mobile network breaks, and will update the registration information to ensure CMS can access to MTS with the correct network address.

WMS is a web-based interface for users to monitor the position of the interested target. It offers users a convenient and friendly environment to interact with the map showing the position or path of target. Moreover, users can decide the location update interval by selecting a desired period. Shorter interval presents the real-time status while longer interval is suitable to observe the trajectory for a long duration. For privacy issues, the authentication of the user is considered and the user will be asked for entering the correct password if MTS of selected target registers to CMS through authorized connection services.

2.1 Central Management Subsystem (CMS)

CMS transfers the request and response between MTS and WMS, and integrates with designed functional services. It runs on a central server with a stationary IP address, so that MTS and WMS can connect to transfer necessary data. The main functions for CMS are to store the content, query the status and forward the request. Therefore, a web server, a database and a proxy are included to achieve the requirement as three functional components.

The web server is used for receiving the request from the user and returning the response. Because the system interface is designed by web technologies, the advantages and limitations have to be considered carefully for performing the services. Both front-end and back-end web technologies are adopted for the dynamic and efficient data exchange. For the target who does not allow any people to monitor its location, CMS provides authorized connection services to resist the connection from normal users. This mechanism not only protects the privacy of the target, but prevents the possible waste of bandwidth and extra costs of communication. To reduce any possible redundant connections is always the most important principle for our system design. When MTS first registers its network information to CMS, a customized password can be set by the target when the authorized registration is enabled. Then the password and the target's information are stored into the database on CMS. After the target completes the registration, a dynamic field for entering password appears when the user selects the target for monitoring.

In addition, more than one user may monitor the same target simultaneously, and MTS may receive more than one request in a short time even the chosen update intervals are not the same. In this case, MTS sends the near location data many times according to the number of request. This behavior generates the redundant connections because the content of transmission data is almost the same. Therefore, location cache services are designed on CMS to reduce the number of request for the same location of target. When MTS send the current location data to CMS at the first time, the longitude and the latitude are record in the database with the current timestamp. Next, when a new request from WMS comes in, CMS compares the time difference between the arriving time of the request and the timestamp in the database. If the period difference is shorter than a threshold, CMS fetches the location in the database as the response data. On the other hand, if the period difference exceeds a threshold, the location data in the database is considered as out of date. Thus, CMS forwards the request to MTS to obtain the update location data. After acquiring the data, CMS overwrites the last record longitude and latitude with a new current timestamp as cache data for the next request.

What is more, CMS provides the message delivery services to share the information between WMS and MTS. Viewers can directly input the messages on the web page to communicate with the target. Not only the messages can be shared, but also the path planning result of MTS can be transferred to CMS for sharing.

2.2 Mobile Tracking Subsystem (MTS)

MTS, running at the target side, returns the data passively for external requests instead of uploading the data to CMS periodically. This is the key feature for reducing the redundant connections. A suitable update interval of data is different for various applications. For fleet management, the driving distance is usually so long that knowing whether the vehicles are at the route is more important than their exactly locations. In contrast, for applications about people finding, the information of real-time position would be more beneficial. Usually the target cannot know what the update interval is suitable for users. To provide real-time data, the target has to transfer the data in a short interval. However, the connection becomes redundant if no users need the information at the moment. Too many redundant connections produce a waste of bandwidth and extra costs of communication. Therefore, the design strategy of proposed connection scheme adopts the transfer on demand, i.e., MTS only transfers the data when receiving the request from CMS.

When MTS connects to the Internet at the first time, it has to send a specified message containing a network address and identification data to CMS as registration. Without this action, WMS cannot show the target's ID for users to select, and CMS cannot connect to MTS to transfer requests. If the target intends to enable the authorized connection services, an additional flag with a customized password is included in the message for registration. In this condition, when a user selects this target to monitor, a matched password and a nickname of the user for identification are needed to input.

To achieve the connection scheme for response on demand, a typical request-response protocol, HTTP, is used on MTS. HTTP protocol is a stateless and connectionless protocol, and is quite suitable for our connection scheme. Using HTTP protocol makes subsystems cooperate easily and efficiently because CMS and WMS also associate with web technologies. Therefore, a server containing a part of HTTP/1.1 protocol is implemented on MTS. It analyses GET and POST methods from CMS and performs the response with a standard HTTP header.

MTS provides both the current location information from the GPS module and the path planning results from the digital map. A general GPS receiver offers the data in National Marine Electronics Association (NMEA) format, which contains the longitude, latitude, altitude, speed, bearing and other interested values for usage. MTS picks necessary values according to the request from CMS. If users only request to monitor the location of the target, the longitude and latitude are sufficient for the response message. Similarly, more interested values will be selected if users demand more information. Only necessary information is transferred to CMS to avoid the waste of bandwidth and communication costs. To offer an extended format for information sharing, all selected values are packaged in Extensible Markup Language (XML) format. With XML format, MTS can accommodate new requirements readily. In the XML document, the text enclosed by the tags may contain an arbitrary number of XML elements. Therefore, the designed response message is placed with the interested values as the elements to complete a structured XML document. The packaging result is shown in Fig. 1.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <response>
    <status>1</status>
    <longitude>120.9998</longitude>
    <latitude>24.78866</latitude>
</response>
```

Fig. 1. The packaging result in XML format.

MTS provides navigation function by assistance of the digital map. By sharing the path planning result to CMS, users can know the target's destination and route as advanced information. Furthermore, MTS activates the rerouting function when the target deviates from the planning route, and a new path planning result is transferred to CMS to update the old path data. To collaborate with WMS, the path planning result is packaged in Keyhole Markup Language (KML) format, which is an XML-based language schema for expressing geographic annotation and visualization.

Furthermore, auto-registration mechanism ensures that MTS can be reached by CMS. Although MTS registers to CMS when connecting to the Internet at the first time, the connection may be broken due to the bad network quality or other uncontrollable factors. Therefore, auto-registration mechanism is introduced to solve this situation. When the reconnection and the subsequent IP change occur, MTS will transfer the new network information to update the old one on CMS.

2.3 Web Monitor Subsystem (WMS)

WMS is a web-based subsystem running with a web server, and provides friendly web pages for users to interact with CMS over HTTP-based communications. WMS mainly uses front-end web technologies to offer users good experience of the real-time response. The main functions for WMS are to send users' requests, receive the response from CMS and collaborate with other services. The design principle for WMS is easily deployed and manipulated. Hence, the compatibility for different browsers and content response time are optimized through the specified design.

WMS is the entrance of the system for users, and provides users high control ability including the target selection, the mode of showing position, and the position update interval. The user-chosen update interval, which relates to the frequency of requests to MTS, is the main distinguishing feature. With this feature, the redundant connections are eliminated because every connection is responsible to a request accordingly.

Moreover, an adaptive update interval mechanism is provided on MTS to further reduce redundant connections. The target's action is generally divided into two kinds of movement, moving or static state. At first the target is moving toward to the destination, but may stop to wait for traffic lights when crossing the road. Maybe he encounters someone and stops to have a talk with, or something catches his eye and makes him stop to check what happened. In this case, the target is almost immobile, but the requests still come in to MTS according to the chosen update interval on WMS. The following consecutive location data are almost the similar position. From users' prospect, if the update position is the same as the last one, the update process becomes unnecessary.

Therefore, the adaptive update interval mechanism is designed to deal with this situation. Every time the position is updated, the displacement *d* between current and last position is calculated. If d is shorter than a threshold d_{min}, the target is considered as immobile, and a counter *m* used to identify the movement accumulates. On the other hand, if d does not exceed the d_{min} next time, *m* will directly reset to the value of zero. Once the counter surpasses a customized threshold *n*, MTS concludes the target is in a static state. Thus, the update interval *T* will be automatically adjusted to its double. However, once the counter is lower than the threshold, the update interval will soon return to the original one T_{ori} set by the user. The flowchart of the adaptive update interval mechanism is shown in Fig. 2. By extending the original update interval, the redundant connections caused by the similar position are decreased.

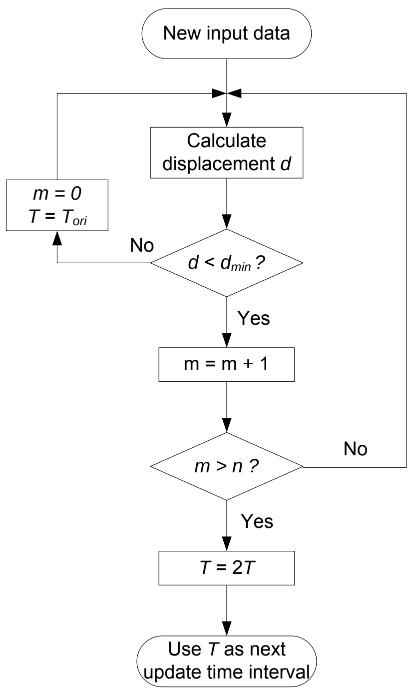

Fig. 2. The flowchart of the adaptive update interval mechanism.

2.4 System Interactive Procedures

This subsection describes several procedures to assist the comprehension of the cooperation among three subsystems. Four major phases, the registration, location monitoring, path displaying and adaptive update interval, are emphasized to explain the different actions in Fig. 3.

1) Registration and Auto-registration

The first action is the registration from MTS, occurring when it connects to the Internet. The target can manually set a name as identification, and enable the authorized connection services if needed. With authorized connection services, a customized password is required to complete the registration process. When receiving the registration information, CMS compares it with the data in the database to determine whether the target has registered or not. If the name of the target exists, CMS update the other registration information. Otherwise, a new record will be inserted into the database. After the target has registered, MTS may disconnect from the network due to some reasons. In this case, when MTS reconnects to the Internet, the auto-registration mechanism retransfers the registration information to update the data in the database.

2) Location Monitoring

After the registration from MTS, the user can select the registered target's name to monitor through WMS. Once the user has chosen the target, CMS checks if the selected target has enabled the authorized connection services. If MTS successfully receives the request from CMS, the location data will be immediately transferred to CMS. At this time, the location cache services are invoked to serve the multi-requests at the same time.

3) Path Displaying

Path displaying only works for users after the target has uploaded its path planning result to CMS. Sometimes the target may just wander around, but sometimes may have a destination to go. MTS provides the navigation function for the target by the assistance of the digital map. The path planning result can be packaged into a KML format file and transferred to CMS, while the geography information is stored into the database too. After the user selects this target, an additional option to display the target's path is shown. Because the planning path is already stored in CMS, the request for path displaying from other users does not make extra connections to MTS. WMS can directly fetch the path result from CMS and display it on the map.

4) Adaptive Update Interval

Adaptive update interval associates with WMS to reduce the redundant connections when the target is nearly immobile. At first the location update frequency is according to the chosen update interval by the user. After the location is transferred to CMS, the displacement of consecutive locations is

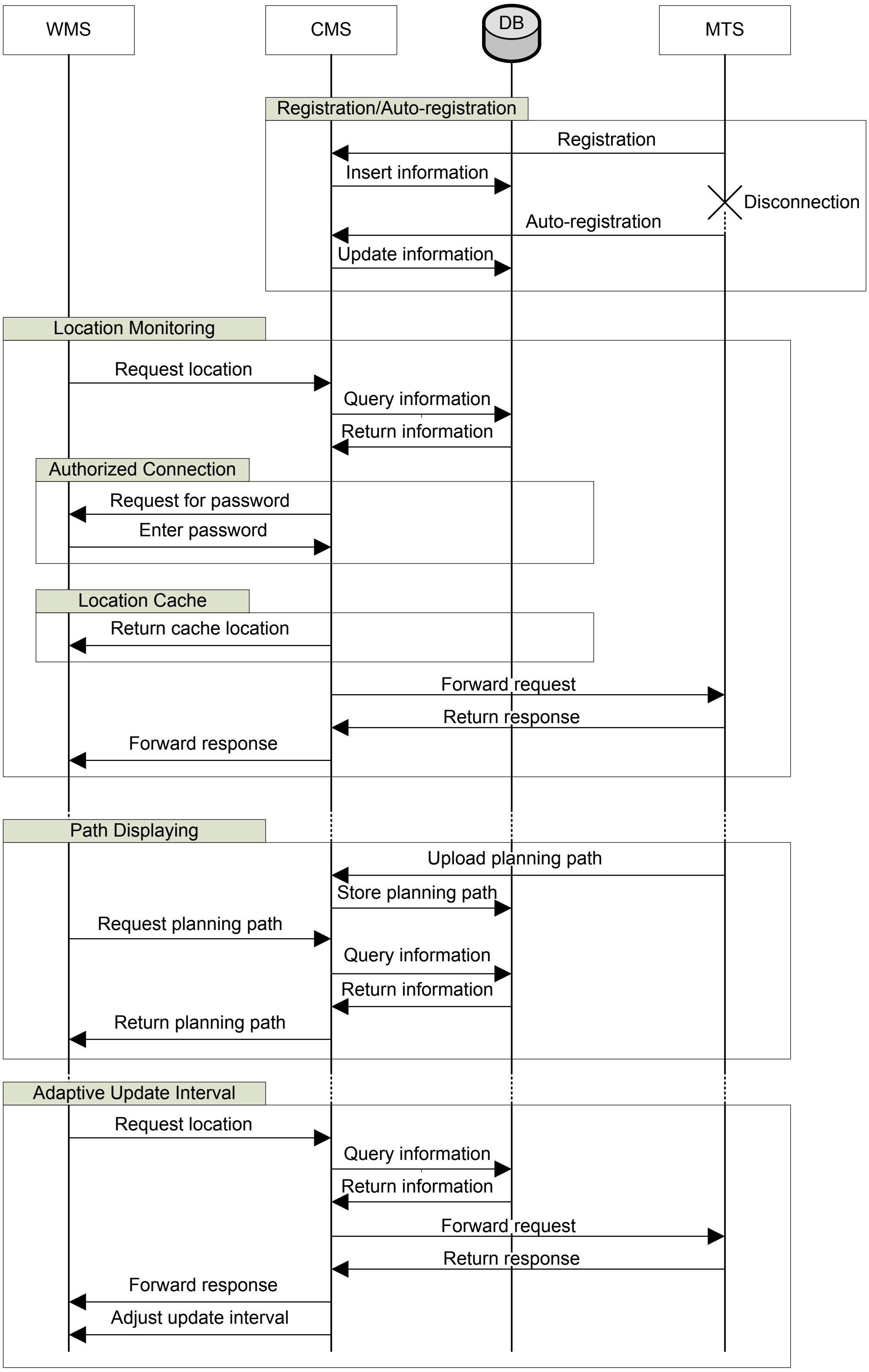

Fig. 3. The action flowcharts in the proposed system.

calculated to identify if the target is moving or not. If WMS determines the target is not on the movement by adaptive update interval mechanism, the update interval is changed to its double comparing to the original one. Therefore, the time for the next request from WMS is extended and the times of connections for requesting the similar location are decreased.

3 Prototype and Implementation

This section describes the implementation details and the experimental results of our prototype. The proposed tracking architecture has been verified on an embedded platform as a web-based tracking system. The system usages and the functional services are exhibited with real demonstrations through some snapshots.

3.1 Prototype Development

The prototype integrated a PC as the platform to develop CMS and WMS, and an embedded platform to serve as MTS. CMS and WMS were built with an Apache web server and programmed with PHP scripts and JavaScript. To offer dynamic interfaces on web pages, Ajax was utilized to process the users' action. However, Ajax is not allowed to make a cross domain request due to the security issue. This restriction failed the connection when CMS intended to connect to MTS. Hence, a proxy implemented with PHP scripts to perform socket communications was designed to conquer this limitation. Because the proposed architecture was used for web-based tracking systems, Google Maps service was used to ease the development of location display. Therefore, WMS also had to associate Google Maps API to obtain the service.

MTS was implemented on an ARM platform with Linux kernel 2.6.22, and was programmed with Qt/Embedded 4.6.2. The specifications of the used platform are listed in Table 1.

Table 1. The specification of embedded platform.

Model	SBC8100
GPP	600-MHz ARM Cortex™-A8 Core
DSP	412-MHz TMS320C64x+™ DSP Core
ROM	128MByte NAND Flash
RAM	128MByte DDR SDRAM

The peripherals of MTS include a Bluetooth GPS module as a location sensor, a general mobile phone as a 3G modem used to connect to the Internet, and a SD card for storing the log data. Fig. 4 shows the prototype of MTS, and Fig. 5 shows the testing configuration consisting of three subsystems: MTS, CMS, and WMS. CMS and WMS were executed on the different PCs and connected with each other through Ethernet, while MTS was executed on the embedded platform and connected to CMS over the air through the mobile network.

Fig. 4. The prototype of MTS.

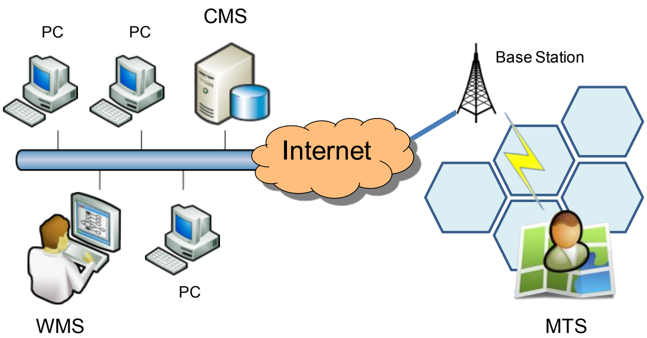

Fig. 5. Testing configuration.

3.2 Functional Services

The tracking system has implemented the proposed connection scheme and collocated with several functional services for a practical application. The usage of the system is performed with the following real demonstrations through some snapshots.

Fig. 6 shows the different responses on WMS for users under authorized connection services. If MTS registers to CMS without enabling authorized connection services, only ID and IP address for the target will be transferred to CMS as shown in Fig. 6(a). After a user uses the pull-down menu to select this target to monitor, the location on the map will be straightly displayed as shown in Fig. 6(b). On the other hand, Fig. 6(c) shows that an additional data, i.e., a customized password, is attached to the original registration information if the target enables the authorized connection services. In this case, the user can not directly obtain the target's information, but has to enter the correct password through WMS. After the user clicks *Check* button, WMS sends the input password to CMS to match the stored one registered by the target. If the input password is not matched, the page inquiring for the password will not vanish as shown in Fig. 6(d).

User-chosen location update interval is the most important feature on the proposed tracking system.

After the user start to monitor the target's location, a pull-down menu contains the different period values, *1 sec*, *5 sec* and *30 sec*, can be arbitrarily selected by the user anytime. The selected new update interval immediately affects the appealing time of the target's next location. The results for the user that has chosen the period of 1 sec and 5 sec are illustrated in Fig. 7(a) and Fig. 7(b), respectively.

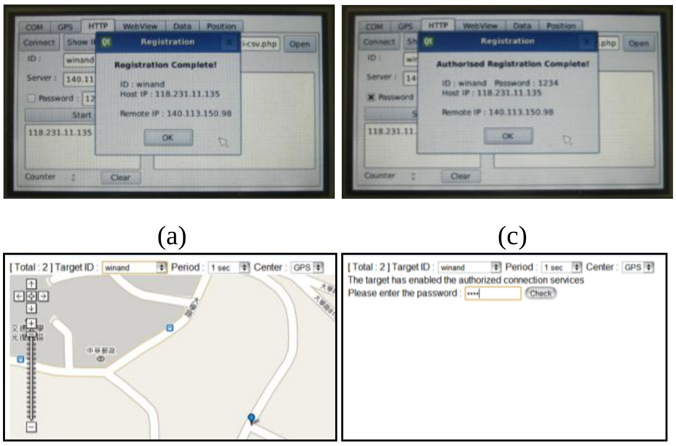
(a) (c)
(b) (d)

Fig. 6. Authorized connection services. (a) Register without the password. (b) WMS shows the map directly. (c) Register with the password. (d) WMS requests the password.


(a) (b)

Fig. 7. User-chosen location update interval. (a) Chosen period: 1 sec. (b) Chosen period: 5 sec.

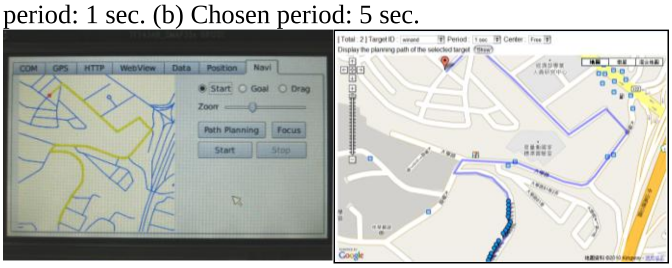
(a) (b)

Fig. 8. Path Displaying. (a) Path planning results on MTS. (b) Path displaying on WMS.

Fig. 8 depicts the usage of path displaying for users. MTS can not only provide its location to CMS, but also the planning path. Because MTS is designed for a navigation device, the basic capacity is to offer the target a stationary map to search routes and the path planning function to show the navigation path. With the support of the network connectivity, MTS can further present this information for users. Fig. 8(a) shows the path planning results on MTS after the target has set the destination. Meanwhile, the path results are packaged in KML format and uploaded to CMS. Once the user selects this target and CMS confirms that the planning path file exists, WMS will reveal a *Show* button for users. Fig. 8(b) shows the path displaying results after the user clicks the button.

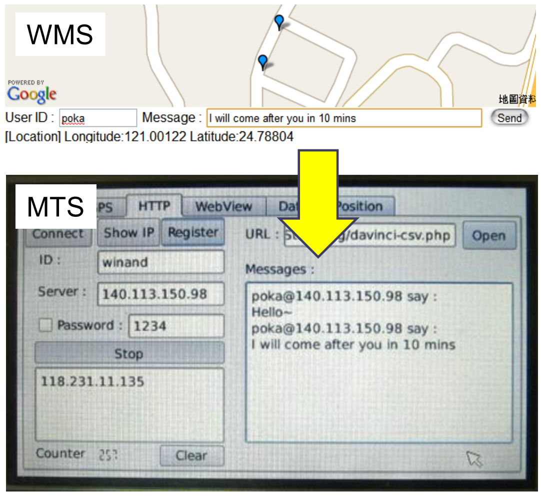

Fig. 9. Message delivery services.

Fig. 9 demonstrates the interaction of the message delivery services. When monitoring the interested target, the user with a requisite user ID for identification can input some words on the webpage to communicate with the target. The input text messages are transferred to MTS as soon as the *Send* button is clicked by the user. Furthermore, the user's information, such as user ID and IP address are combined with the delivery messages and displayed on the screen of MTS.

3.3 Communication Performance

To evaluate the update efficiency of the system, the process of update is divided into four stages including the time for processing and transmission as shown in Fig. 10.

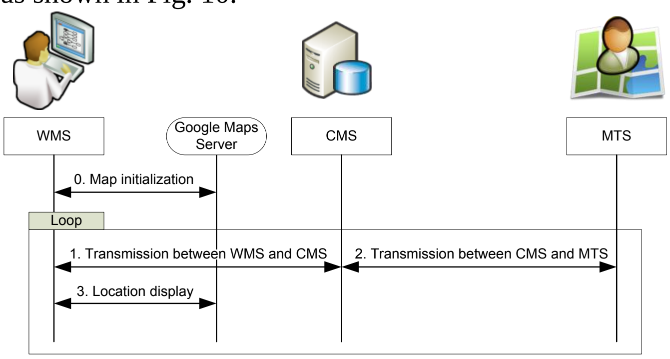

Fig. 10. The flow diagram of update data.

Stage 0 represents the time of initial setting for requesting a map from the Google Maps server. Stage 1 means the data transmission time between WMS and CMS through Ethernet. On the contrary, stage 2 is the data transmission time between CMS and MTS through the mobile network, such as 3G in our experiments. After obtaining the response message from MTS, WMS has to parse the data and draw an icon on the center of map to show the target's current location. Stage 3 represents the cost time for above location display process. Noticeably, stage 0 is only executed once when the user first accesses to WMS, and the period of loop containing stage 1 to 3 is repeated according to the update interval set by the user.

By the default configuration, the user connects to CMS through Ethernet. However, the user may use other methods to connect to the Internet, such as Wi-Fi or 3G. Therefore, an evaluation for the total latency time when the user exploits different communication medium at stage 1 is performed as shown in Table 2 and Fig. 11. An extra stage 4 shown in the figure means the total time cost from the stage 1 to stage 3, i.e., the time needed for the user from sending a request to viewing an icon on the map. From the system architecture, the time spent at stage 0 and stage 3 relates to the interpreter ability of the browser, which is an uncontrollable factor to the user. In our evaluation, the most considerable part is the worst delay time through the different medium, especially for the low-bandwidth medium. The result at stage 4 shows that the user can obtain the update data within 400 ms by using Ethernet or Wi-Fi. Even through 3G, the update data can be retrieved within 700 ms. For a general GPS module, the sampling frequency is 1 Hz. Therefore, this result also guarantees that the user can monitor the status of the target in real time even in the worst case. To clarify the benefit of the connection scheme, we exploit a test scenario to present the usage of connections for two mechanisms as shown in Fig. 12.

Table 2. Latency time at each stage.

Stage	0	1	2	3	4 (1+2+3)
Ethernet (WMS) + 3G (MTS)	17 ms	6 ms	341 ms	13 ms	360 ms
Wi-Fi (WMS) + 3G (MTS)	29 ms	46 ms	336 ms	18 ms	400 ms
3G (WMS) + 3G (MTS)	68 ms	244 ms	379 ms	18 ms	641 ms

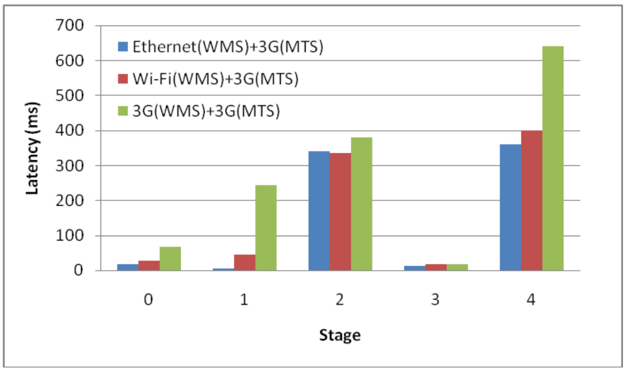

Fig. 11. Latency time at each stage.

The update interval is set to 1 sec in WMS, and the record period is one hour. The viewer is request to monitor the target for 1, 5, 10, 30, and 60 minutes at any moment during the period. Because the rate of sending requests is 1Hz, the monitoring for 1 minute means there are 60 connections in 1 hour. For conventional tracking systems, whether the viewer monitors for 1 or 30 minutes, the number of connections in 1 hour always is 3600 due to the polling mechanism. On the contrary, the number of connections is varying according to the actual monitoring period in our mechanism. We can observe a significant reduction if the monitoring period is shorter than the record period. However, if the monitoring period is longer, the benefit of our mechanism is not obvious comparing to the polling mechanism.

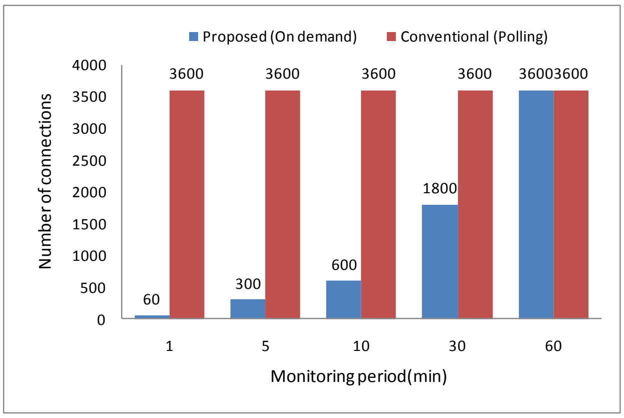

Fig. 12. Usage of connections for two mechanisms.

4 Conclusion

This paper has proposed a demand-driven architecture of connection scheme for the practical web-based tracking systems with the real-time monitoring performance. Based on the proposed connection scheme, three subsystems have been developed and offer the authorized connection services, the message delivery services and the auto-registration mechanism. The location cache services and the adaptive update interval mechanism are also introduced to further reduce the redundant connections. In addition, the problem for the data asynchronization is solved under this system architecture, because the data is transferred on demand. The transmission latency through different medium has been evaluated to proof the system can maintain the real time performance. We have demonstrated that if a web-based tracking system without the requirement of long-term monitoring, the system based on this demand-driven connection scheme is more efficient and economical. This benefit also encourages the people who are reluctant to spend too much on communication fee to use the application. At present, MTS is developed on an embedded platform to verify the function. In the future, real handheld devices will be utilized to develop peer-to-peer communications. The security, the rich content and the flexible management for sharing information will be interesting extensions.

References:

[1] M. A. Al-Taee, O. B. Khader, and N. A. Al-Saber, Remote monitoring of vehicle diagnostics and location using a smart box with Global Positioning System and General Packet Radio Service, *Proc. IEEE Computer Systems and Applications,* Amman, 2007, pp. 385-388.

[2] D. W. Ryoo, and C. Bae, Design of the wearable gadgets for life-log services based on UTC, *IEEE Trans. Consumer Electron.*, Vol. 53, 2007, pp. 1477-1482.

[3] H. Lim, J. H. Yang, and Y. S. Lee, The development of an ubiquitous learning system based on audio augmented reality, *Proc. IEEE Control, Automation and Systems,* Seoul, 2007, pp. 1072-1077.

[4] M. Modsching, R. Kramer, M. Riebeck, A. Stark, K. ten Hagen, and J. Kawalek, Field trial on the efficiency and user experience of GPS based state of the art navigational systems for pedestrians, *Proc. Positioning, Navigation and Communication*, Hannover, 2007, pp. 129-134.

[5] P. I. Philippopoulos, G. Soulos, A. Krukowski, N. Argyreas, V. Gorgas, D. Kyriazanos, D. I. Axiotis, and S. C. A. Thomopoulos, ToD: an intelligent location-based transport management system, *Proc. IEEE Intelligent Vehicles*, Istanbul, 2007, pp. 793-798.

[6] S. T. S. Thong, C. T. Han, and T. A. Rahman, Intelligent fleet management system with concurrent GPS & GSM real-time positioning technology, *Proc. ITS Telecommunications*, Sophia Antipolis, 2007, pp. 1-6.

[7] A. H. Mazinan, Specific persons surveillance using satellite technique, *Proc. Intelligent and Advanced Systems*, Kuala Lumpur, 2007, pp. 474-478.

[8] D. Titica, O. Fratu, E. Stanescu, and S. Halunga-Fratu, Simple location-based application development for mobile phones, *Proc. Telecommunications in Modern Satellite, Cable and Broadcasting Services*, Serbia, 2007, pp. 15-18.

[9] N. Chadil, A. Russameesawang, and P. Keeratiwintakorn, Real-time tracking management system using GPS, GPRS and Google earth, *Proc. Electrical Engineering/Electronics, Computer, Telecom. and Information Technology*, Krabi, 2008, pp. 393-396.

[10] G. B. Al-Suwaidi and M. J. Zemerly, Locating friends and family using mobile phones with Global Positioning System (GPS), *Proc. Computer Systems and Applications*, Rabat, 2009, pp. 555-558.

[11] S. Barbeau, M. A. Labrador, A. Perez, P. Winters, N. Georggi, D. Aguilar, and R. Perez, Dynamic management of real-time location data on GPS-enabled mobile phones, *Proc. Mobile Ubiquitous Computing, Systems, Services and Technologies*, Valencia 2008, pp. 343-348.

[12] L. Shen, C. Yang, F. Zhang, Y. Ren, L. Wu, and L. Zhu, A scalable web GPS monitoring system based on AJAX push pattern, *Proc. Geoinformatics*, Fairfax, VA, 2009, pp. 1-5.

[13] B. F. Wu, Y. H. Chen, C. J. Chen, C. C. Kao, and P. C. Huang, An efficient web-based tracking system through reduction of redundant connections, *Proc. Advances in Neural Network Research and Applications*, 2010, pp. 671-677.