Distributed Mobile Traffic Monitoring System Based on Free and Open Source Software

Velislava Spasova, Grisha Spasov

Abstract: *In recent years the development of technology and the evolution of mobile phones made possible their integration and usage in a wide range of mobile systems. In this paper it is presented a mobile distributed system which monitors the traffic in areas without monitoring cameras and gathers series of photographs which could be used to determine whether there is congestion in a given region. The system is realized using Free and Open Source Software (FOSS) which contributes to its flexibility and portability.*

Key words: *Distributed Mobile System, Traffic Monitoring, Free and Open Source Software, Java Micro Edition, Global Positioning System.*

INTRODUCTION

During the last several decades traffic in big cities and agglomerations has become increasingly heavy and congestions have become an everyday part of many people's lives. Different strategies have been created to prevent congestions or to quickly eliminate them once occurred ([4], [5]). One such measure that is becoming increasingly popular is to put monitoring cameras in places that are known to be points of regular congestion occurrences. The monitoring information from these cameras is then processed and in case of heavy traffic some kind of alert message is broadcast so that the majority of the drivers could easily notice it. Although highly efficient this measure is not a perfect solution because sometimes congestions occur in places without cameras and in that case they cannot be handled properly.

An efficient supplement to that congestion prevention strategy is to have a number of movable cameras which could be used dynamically in case of a remote congestion i.e. cameras that provide information only when such is needed. The question is where to put these cameras and how to connect them to a monitoring center. This paper presents a solution in which the cameras are placed inside a car and the connection to the center is realized through the Internet over GPRS i.e. using a mobile operator's network. The presented solution employs mobile phone with an integrated or external camera and with an integrated or external GPS module to take regular snapshots of the road and to determine the location at which these snapshots have been taken. Using a mobile operator's network makes the proposed solution very flexible as the only requirement to send the information is to have network coverage at the place where the picture is taken. As in most developed countries mobile operators have covered the whole country, this requirement is easily fulfilled. The software modules of this solution have been realized with free or open source software (FOSS) which gives additional level of flexibility and portability of the system [1].

ARCHITECTURAL OVERVIEW

The architecture of the proposed distributed mobile traffic monitoring system is presented in Figure 1.

Every Mobile Observation Point (MOP) includes a mobile phone with a SIM card of a mobile operator, a camera (which could be integrated in the mobile phone) and a GPS module (which also could be integrated in the mobile phone). Every MOP could use SIM card of a different mobile operator – the only requirement is to be able to send data through the Internet. When the MOP application is started it opens an Internet connection to the Monitoring Centre (MC) and keeps it alive until it exits. The Internet connection is realized in accordance to the following procedure: the MOP connects to the nearest mobile operator's base station over GPRS; the base station connects to the mobile operator's network; internal routing in the network is realized to reach the Internet gateway (which often is a Network Address Translation box as well); through the gateway a connection is realized to the MC server which listens on a previously defined port. As far as the MOP is concerned, it has to connect over GPRS to a base station. All the other internal and external routing and NAT is hidden for the MOP.

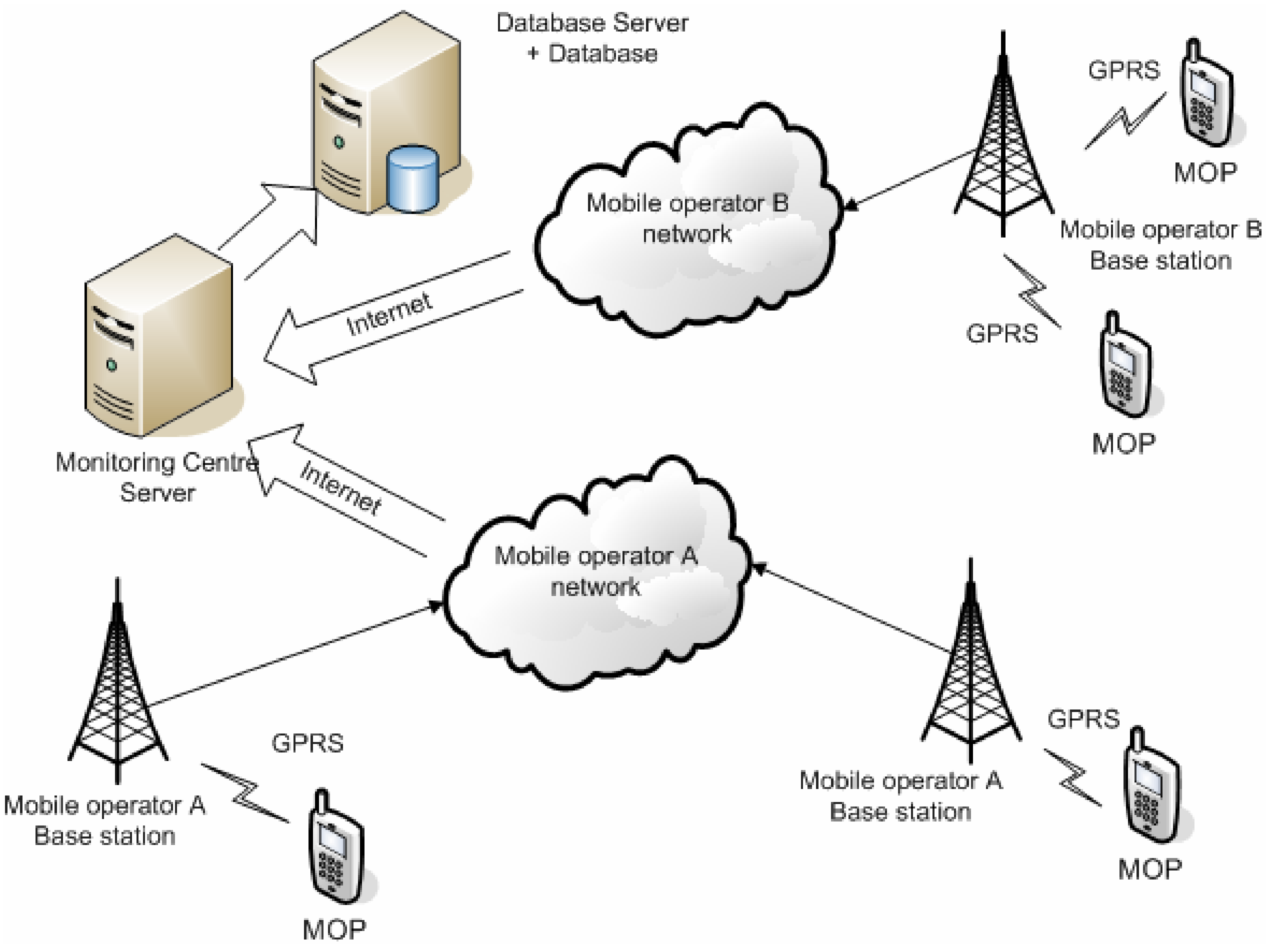

Figure 1. Architecture of the distributed mobile traffic monitoring system

After the connection to the MC server is realized the application running on the MOP waits for command from the server. When command to send data is received, the MOP takes picture of the road, then takes the location where the picture is shot and sends both to the MC. The MC server receives the data from the corresponding MOP and writes it into a database along with the time when the picture is taken. It is the MC server's responsibility to determine when a picture has to be taken from a specific MOP. Also the MC server should be able to handle multiple MOPs in the same time i.e. some kind of parallelism should be implemented in the server application. In the proposed solution two possibilities are suggested for the server application – to develop a custom-designed server and to use a standard HTTP server. Both approaches are discussed in the following sections.

THE TCP/IP APPROACH

This approach uses TCP/IP for MOPs to connect to a custom-designed MC server. TCP/IP is used because a reliable way of transmitting the information from a MOP to the MC server is required. UDP/IP could be used as well but in that case the application should consider and assure the reliable data transmission. In this case the infrastructure of the mobile operator network also puts some limitations. In comparison with HTTP which is another way of reliable data transmission, the advantages of TCP/IP approach are:

compact MOP and server applications source code, more flexible and customizable server and less traffic overhead. This is important issue as access to the Internet is realized through a mobile operator's network where the prices are more expensive then these of the other wireless internet service providers and even more: the prices are based on the traffic rather than on the time spent in the Internet. The disadvantages of the TCP/IP approach are exposed when it has to be integrated in complex traffic control systems where middle layers should be introduced.

The proposed TCP/IP approach is presented on Figure 2.

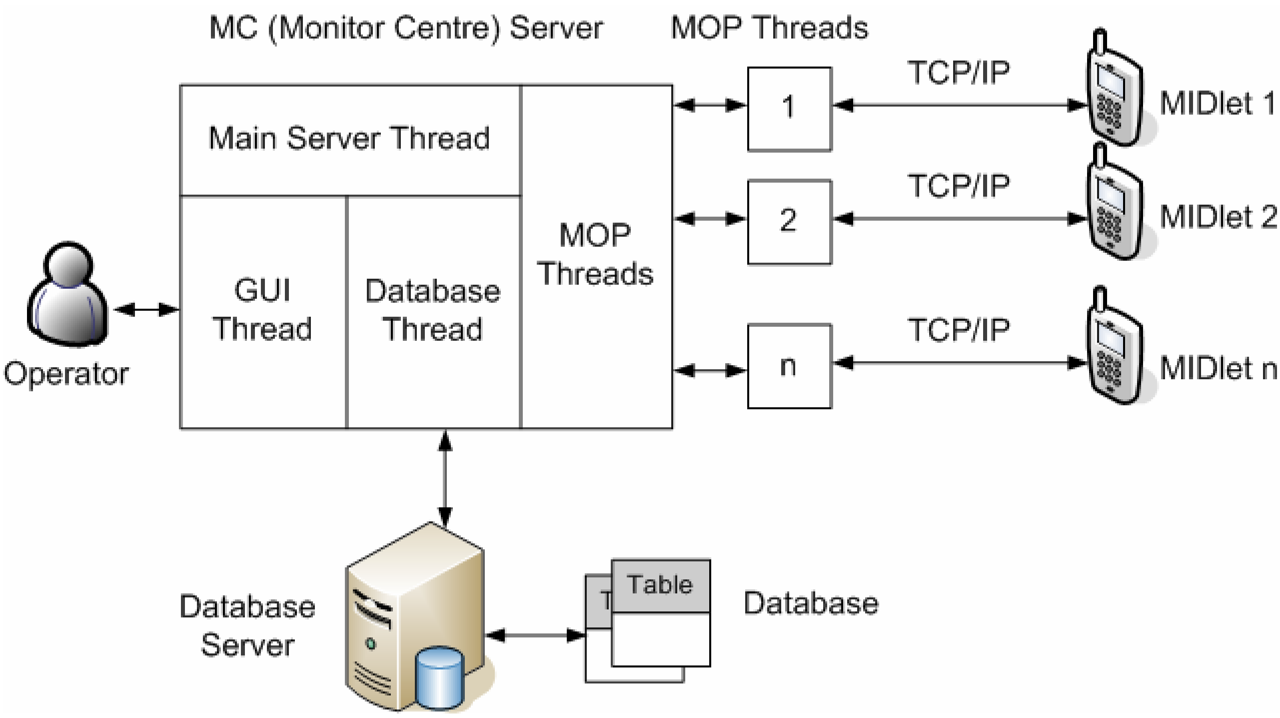

Figure 2. The proposed TCP/IP Approach

On each MOP a MIDlet (a Java 2 ME application) is running which realizes the functionality of the MOP. It opens a SocketConnection ([2], [3]) to the server. The server is developed using Java SE and has one main thread which is responsible for listening on a previously specified port for incoming socket (TCP) connections from MOPs. When one such connection is detected, another thread is started which deals with that MOP. The number of the MOP threads running on the server machine equals the number of the currently connected MOPs. There is another thread that deals with the database server and is responsible for writing data to and reading data from the database. There is one more thread which is responsible for the graphical user interface (GUI) so that the operator(s) in the Monitoring Centre could easily interact with the server through a graphical interface.

THE HTTP APPROACH

This approach uses HTTP to access the server. As previously mentioned HTTP has more overhead than TCP/IP but it has also some advantages: there are ready standard HTTP servers which could be used as a MC server; when using such a product firewalls running on the server machine could be bypassed. Firewalls could be a problem in the TCP/IP approach when the server is running on a lower layer in the network protocol stack.

The proposed HTTP approach is presented on Figure 3.

Here MIDlets are running on every MOP but the MIDlets open HttpConnection ([2], [3]) to the MC server instead of SocketConnection. On the server side all the functionality regarding the network communication and the GUI is realized by the HTTP server, so it is not necessary to develop that part of the application. The only modules to develop are the input interface and the connection to the database. Any script language that is supported by the server such as PHP or JSP (Java Server Pages) technology could be used. In our example JSP is employed. Two JSP pages are proposed – one realizing the interface to the MOPs and another for communication with the database. The advantage of this approach is that pictures from hot congestion areas could be published in real time to a website and everyone could have access to them.

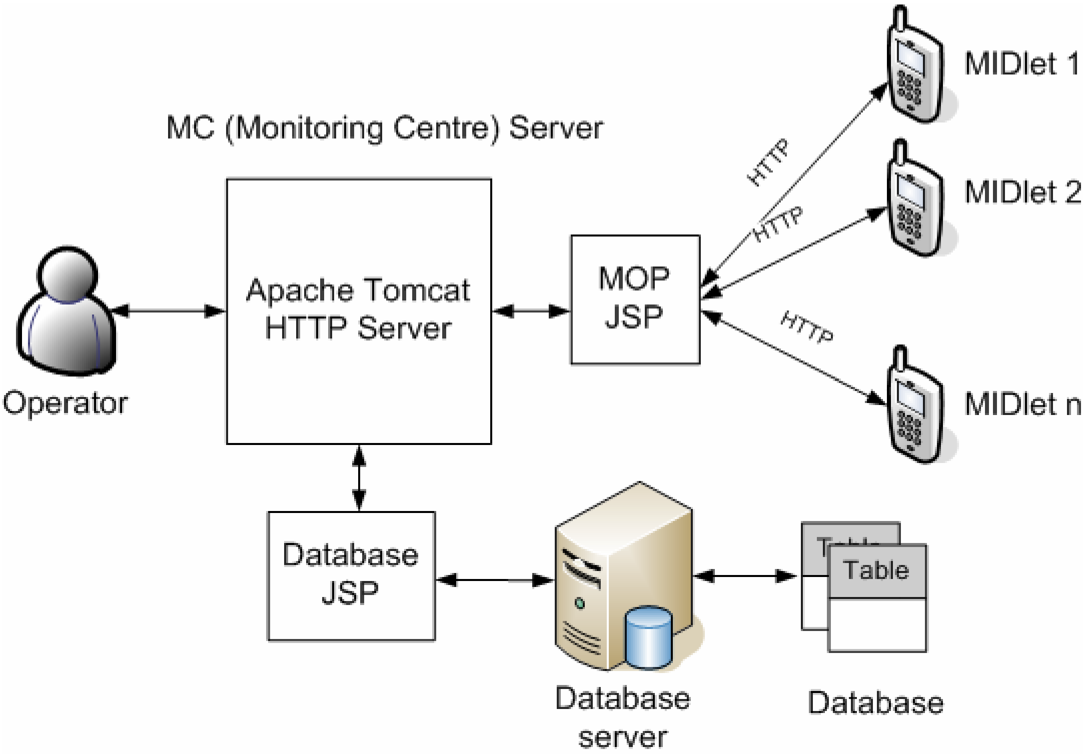

Figure 3. The proposed HTTP Approach

FREE AND OPEN SOURCE SOFTWARE

In this section it is presented the FOSS that is used for the development of this solution.

Programming Language.

The programming language used for the development of this system is Java. For the application running on the MOPs it is Java ME (Java Micro Edition), CLDC 1.1 (Connected Limited Device Configuration), MIDP 2.0 (Mobile Information Device Profile). This standard provides class libraries that every mobile phone manufacturer should implement. This makes the code very portable across different mobile platforms. However, for every specific device the code should be compiled and linked using the development kit provided from the manufacturer because the class library implementations may vary between different models and brands of mobile phones. The optional APIs used are MMAPI (Mobile Media API [3]) and Location API ([2]). The programming language used for the development of the MC server is Java 2 SE (Java Standard Edition). For the GUI it is used Swing API. In the case of the HTTP approach JSP (Java Server Pages) technology is employed.

Java Development Tools.

Although the applications could be developed without IDE (Integrated Development Environment), all the tools such as debugger, linker, text editor, etc. are on disposal in an IDE and the process of development and testing of the code is much simpler and faster. The proposed solution employes NetBeans 6.5 IDE but Eclipse 3.4 could also be used. Both IDEs are free and open source.

Java ME Emulator and Development Tools.

Most of the mobile phone manufacturers distribute also free Software Development Kits for Java ME developers. These kits include the implementation of the classes from the Java ME standard and often emulators on which the software could be tested before being deployed at the phone. Sun also distributes the so-called Sun Java Wireless Toolkit (WTK). Some of the manufacturers' toolkits include: Nokia S60 SDK, Nokia S40 SDK, Sony Ericsson SDK, Samsung SDK, SMTK (Siemens Mobility ToolKit), Motodev Studio (Motorola), etc.

Database Server.

MySQL database server is used because it is free and open source and supports all the needed functionality.

HTTP Server.

In the case of the HTTP approach our suggestion is to use Apache Tomcat 6 which is free and open source "pure Java" HTTP server and implementation of Java servlet and JSP technologies.

CONCLUSIONS AND FUTURE WORK

The presented solution offers an efficient and flexible supplement to the already wide-spread static traffic monitoring systems. The combination of the static and mobile cameras builds complete, effective and functioning congestion prevention and control system. The use of FOSS adds more flexibility and lowers the price of the system. This solution could be extended with a subscription network. Every subscriber will receive regular reports for congested areas in a region of interest.

In the future the mobile phones in the system could be replaced with microcontroller units (MCUs) with an integrated GSM module. The software running on these MCUs could use the same language (J2ME) and the same libraries as the application running on a mobile phone from the same manufacturer. The use of a MCU will provide greater possibilities for connection to other devices such as GPS, sensors, etc. Another aspect in which efforts could be concentrated is to create an image recognition module for the monitoring center server which will make the solution even more efficient because it will eliminate the subjective and slow human factor in the system.

REFERENCES

[1] A. Kalicki, L. Makowski, A. Michalski, Z. Staroszczyk, „Distributed Measurements Systems – A Web System Approach, Part 2", IEEE Instrumentation&Measurement Magazine, vol. 11, No 6, pp. 44 – 50, December 2008

[2] J. Knudsen, "Kicking Butt with MIDP and MSA: Creating Great Mobile Applications", Prentice Hall, December 2007, ISBN: 0-321-46342-0

[3] V. Goyal, "Pro Java ME MMAPI: Mobile Media API for Java Micro Edition", Apress, 2006, ISBN: 1-59059-639-0

[4] Jen-Chao Tai, Shung-Tsang Tseng, Ching-Po Lin, Kai-Tai Song, "Real-Time Image Tracking for automatic traffic monitoring and enforcement applications", Image and Vision Computing, vol. 22, Issue 6, pp. 481-503, June 2004

[5] Zhigang Zhu, Guangyou Xu, Bo Yang, Dingji Shi, Xueyin Lin. "VISATRAM: a real-time vision system for automatic traffic monitoring", Image and Vision Computing, vol. 18, Issue 10, pp. 781-794, July 2000

ABOUT THE AUTHOR

Velislava Spasova, 4th year BCs student in Computer Systems and Technologies, Technical University Sofia, branch Plovdiv, E-mail: vgs@tu-plovdiv.bg

Assoc. Prof. Grisha Spasov, PhD, Head of Department of Computer Systems and Technologies, Technical University of Sofia, branch Plovdiv, Phone: +359 (32) 659 724, E-mail: gvs@tu-plovdiv.bg