ISSN: 0128-7680
Pertanika J. Sci. & Technol. **20 (1)**: 63 – 80 (2012) 

# Performance Evaluation of A Mobile Road Traffic Infraction Registration System through Benchmark

**Arasteh-Rad, H.*, Khairulmizam Samsudin, Abdul Rahman Ramli and Mohammad Ali Tavallaie**

*Department of Computer and Communication Systems Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*

**E-mail: HABIB_ARASTEH@yahoo.com*

## ABSTRACT

The rapid development of roads and the increasing number of vehicles have complicated road traffic enforcement in many countries due to limited resources of the traffic police, specifically when traffic infraction registration is done manually. The efficiency of the traffic police can be improved by a computer-based method. This study focused on mobile traffic infraction registration system benchmarking which is used to evaluate the server performance under load. The study attempts to provide a clear guideline for the performance evaluation of mobile road traffic infraction registration system, whereby the traffic police can make decision based on them to migrate from the manual-method toward computer-based method. A closed form of benchmark tool was used for the evaluation of the system performance. The tool was configured to imitate ramp scenarios, and statistics were gathered. The server was monitored at different times and works. Contributing factors include bottleneck, traffic, and response time, which are related with criteria and measurements. The system resource was also monitored for the tests.

**Keywords: Benchmark, road traffic, infraction registration, performance evaluation**

## INTRODUCTION

The telecommunication technology and information revolution are offering solutions for problems in major cities in the world; problems which include congestion, traffic control and E-commerce. The development of computer networking and the use of personal computers have made an important effect to the network applications. However, theses systems have mobility limitations.

In the recent years, the advances of wireless networking, communication, and mobile technology become more popular and create new aspect of services each day. Wireless networks refer to any system of transmitters and receivers that send radio signals over the air, such as a Wi-Fi local networks, cellular networks or satellite networks (Zhu Han & K. J. Ray Liu, 2008). There are several factors and reasons that could lead to the implementation of these services. The most important factor about the wireless services is the anywhere, anytime feature that it offers (Takahashi *et al.,* 2005). Mobile systems and applications could be used to enhance the forms of existing applications (RAD Technical Rep., 2000). Nowadays, mobile equipment has large storage capacity, and a wide array of applications and connectivity options. In other words, mobile devices could create new market since they are even more accessible than PC.

The total number of mobile phone subscribers in the world was estimated to be 2.14 billion in 2005 (Mobiletracker, May 2005). The subscriber count reached 2.6 billion by the end of 2006 (PCWorld, 2006), and this would further boost to 4.5 billion in 2012 (Cellular-News, March 2008).

Received: 23 April 2010
Accepted: 23 August 2010
*Corresponding Author

Around 60% of the world's population have the access to mobile network coverage than it was in 2006. This percentage was expected to increase to 90% by the year 2010 (Turettini, 2006). The rise of the mobile phone technology in developing countries is often cited as an example of the leapfrog effect. However, the cellular network was first designed for voice communication purposes (Qusay, n.d), but further developments within the network (cellular network) have made it possible to transmit data as well. For instance, the GSM technology has the data rate of 9.3 kbps and the current 3-G technology offers a data rate that goes up to 2 Mbps. Data from Informa indicated that by 2010, half of the planet's population would have the access to the Internet through a mobile device (Informa Telecoms & Media, 2008). Surprisingly, road developments have been rapidly occurring and simultaneously increasing the number of vehicles, specifically in developing countries.

For example, the growth rates of the vehicles and roads in Malaysia between 1986 and 2006 were 369% and 147%, respectively (Othman, 2010; Ahmad & Azmi, 2007), and these were 291% and 189% in Iran (Zekavat, n.d). This rapid development introduces a lot of road traffic enforcement issues which are mainly caused by the limited resources of the traffic police. Adding to the problem is that traffic infraction registration is still done manually using a pen and a notebook. Iran is among one of the countries having the highest rates of road violations in the world (Royanian, 2007). A new, convenient and efficient method for infractions registration is deemed to be urgent. In the recent years, some developed countries have initiated and funded the development of electronic ticketing systems. A combination of the E-ticket with wireless or cellular network could potentially increase the efficiency of road traffic rule enforcement. The patrol traffic officers out in the field need a powerful, yet easy-to-use, handheld solution to help them efficiently access traffic databases to collect, transmit and deal with the real-time information. For example, they need to be able to access to the driver offender information to validate the driving license and infraction traffic ticket submission anytime, anywhere. Therefore, there are a lot of officers who can access to the server and make a heavy load on it. Cronkhite states that 'information is the life blood' of police (Cronkhite, 1974). This is because an accurate and rapid flow of police information is essential for effective law enforcement. By using test tools, the server can be tested with various tests including load tests, performance tests, stress tests and ramp tests. Load tests were performed to determine the best estimate of the traffic server needs to support. Consider this as a "real world test" of the server. Performance tests are in fact used to test each part of the server or the Web application to discover how to optimize them for higher traffic. Ramp tests are a set of variations of the stress tests, in which the number of users raise during the test processes, i.e. from a single request to hundreds of requests. Stress tests constitute of the simulated "brute force" attacks that apply excessive loads to the server. For instance, Anwar *et al.* assessed the scalability and performance of a Web application based on the PHP scripts by using stress testing (Anwar & Saleem, 2002). Their experimental work focused on stress testing for two main subjects that contain bandwidth and system loads. Another example would be the work of Santra *et al.* in 2009. The researchers measured memory usage in the web enabled J2EE application under ramp test, while running multithreaded web enabled J2EE application, with and without changing user load.

In this manuscript, a computerized traffic infraction registration system was designed to replace the manual system. The new system has the capability to provide online ticket issuing to mobile officer at the scene. The server's accurate performance was evaluated under performance tests to insure traffic police to migrate from the manual-method towards computer-based method. The rest of this paper is structured as follows. The following section begins with the introduction of the problem statement and next constructs mobile road traffic infraction registration system architecture. Then, Infraction Registration Benchmark Tool (IRBT) that is familiar with the traffic infraction registration system will be elaborated. Meanwhile, experiment methodology is presented in Section 2. Section 3 presents and discusses the results. The last section concludes the present work.

*Problems Statement*

A convenient, reliable, secure and efficient system is in dire need for tickets to be properly registered. In the new system, the patrol officers on duty should have the access to databases to collect, transmit and deal with the infraction information in real time, such as the new computer technologies allow officers to collect data at the scene, transfer the information to the police data centre, provide on-line error checks electronically, and subsequently, issue a traffic ticket in real time. Police should be satisfied with the ability of the new system to approve the migrate. Modelling, analytical system, mathematical simulation and benchmarking are ways to server's performance evaluation. Benchmark was applied in this study to assure traffic police that the new system is always reliable and stable at the expected level even under critical loads. Meanwhile, the hardware and software of server can be assessed by benchmarking under realistic workloads (SPEC, 2009). It has a responsibility for the tuning options best serve requests. Sometimes, a system is designed for a certain level of traffic; when the traffic increases beyond a certain point, response times will also increase to unacceptable time. For instance, some studies have demonstrated that when the response time takes less than 0.1 s, user feels that the system responds instantaneously. Although users are depressed with the waiting time, but they are still focused on the current transaction when the response time becomes less than 1.0 s. Whenever the waiting time gets close to 10s, the likelihood of user distraction increases, and this becomes more than 10 s, the users are most likely to be distracted from the current transaction and lose interest (Dilley, 2002; Abdelazez, 2000). Another consideration is the amount of requests that the server is expected to handle, particularly during the peak load periods. Load and time will also affect on the performance of the server. Therefore, load should be simulated on the servers before putting them online to determine how the server will perform its functions.

Is the server's bandwidth adequate? Is the server prepared for the network traffic that police have prospect? Can the server tolerate the officers' requests traffic? Where are the server's capacity boundaries? These questions can be answered after server performance evaluation has been carried out.

*Road Traffic Infraction Registration System Infrastructure*

The mobile road traffic infractions registration system design was based on the Iranian officer's tasks to meet the requirements of traffic policing duties. The system should be able to handle officer's queries in a secure manner. In this manuscript, a request or query means an officer's request for details of offender(s) before issuing ticket or online issued ticket from the scene to the server. The mobile road traffic infraction registration system was constructed on three-tier architecture or structural design. In this system, the front-end is the mobile device, the middleware is the software server running on the desktop workstation that contains the business logic of the system and back-end is referred to the database server. *Fig. 1* shows the mobile road traffic infraction registration system structure and layout. As shown in *Fig. 1*, mobile devices are used to access the police's system. They are the front-end tier (or the clients) in the system architecture. The officer is accessing the server using the GPRS enabled mobile device, and making direct HTTP connections to the IP-based web server without going through the WAP gateway. This is due to the fact that the TCP/IP network principles and concepts over the cellular network are simulating with GPRS. The mobile device would be treated like another IP-network device (3G Americas, 2008). If an officer is communicating with the system through text messages, the police request sent goes through the SMS gateway (Labordère, 2006). The SMS gateway receives and forwards the message to the police's server. If the officer tries to connect to the system through WAP, the requests and replies go via the WAP gateway. The gateway will then convert the requests into the

format that can be understood by the IP-based network device and vice-versa (RAD, Jan 2009). The middleware and back-end contain the business logic and the database servers of the system, respectively. The server would make database accesses to retrieve or manipulate data whenever it is requested by the officers.

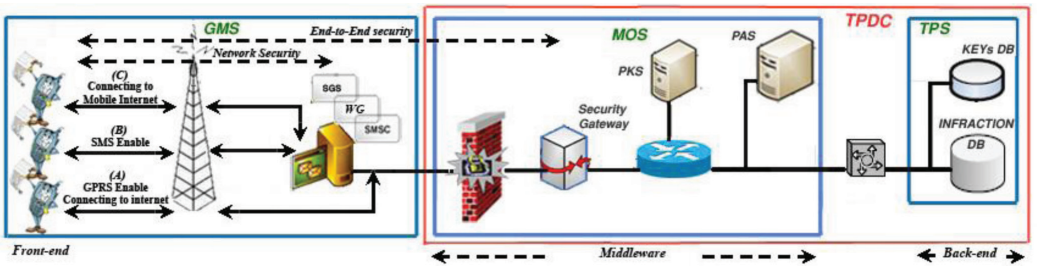


*Fig. 1: The architecture of a secure traffic police mobile communication*

The secure network topology is divided into three subsections, namely, the Traffic Police Subsection (TPS), the Mobile Operator Subsection (MOS) and the General Mobile Subsection (GMS) (see *Fig. 1*). The GMS are facilities operated by a carrier to provide public mobile telecommunication services that include wired and wireless networks. Security Gateway (SG) isolates the TPDC from GMS. The TPS and MOS are located in the TPDC. They are isolated by a VPN gateway to provide a secure real-time data exchange environment (Whale Communications Ltd., Mar 2003). The basic functionality of the TPDC is to handle officers' requests with a secure communication. It is known that mobile devices have limitations on the resources such as power and memory. In the TPDC, however, there are no resource limitations. Therefore, it was decide that most tasks would be handled by TPDC.

In this architecture, when a message (possibly a requesting message) is sent to TPDC by a user, it is first delivered from the Mobile Equipment (ME) to the GMS through the base station system (BSS). The GMS would then forward the message to the application server (PAS) through the SG. On the contrary, when a message (possibly a response message) is sent to a mobile user by the PAS, it is first delivered from the PAS to the GMS through the Security Gateway; the connection between the GMS and the SG is facilitated by a TCP/IP connection over the Internet.

**Databases**

The database schema is shown in *Fig. 2*. The TPDC database requires using of data from other organizations' databases. For instance, citizenship identification data exist in the Personality Registration Organization (P.R.O.) database. Therefore, the TPDC database has been designed as a relational database. In this system, the main databases are *INFRACTION-DB* and *KEYs-DB*. As illustrated in Figure 2, *INFRACTION-DB* contains nine tables which include *Personal, PoliceMan, License, Per_Lic, City, Infcode, Infraction, Vehicle, P_I_V*. The Personal table contains drivers and policeman's personal data. The *PoliceMan* table keeps officers special data such as identity number, mobile number, officer's code, officer's national number, officer's operation section and officer's ID date of expiry. Data pertaining to driving licence are kept in the License table and the city codes stored in the *City* table. Meanwhile, infraction code and type can be retrieved from *Infcode* table and *Infraction* table. The *Vehicle* table contains data on vehicles. The infraction data are stored in the *P_I_V* table and would be the most important table which is frequently used in officer's queries. Officer's transaction keys are stored in *KEYs-DB*. The internal-databases are located in TPS.

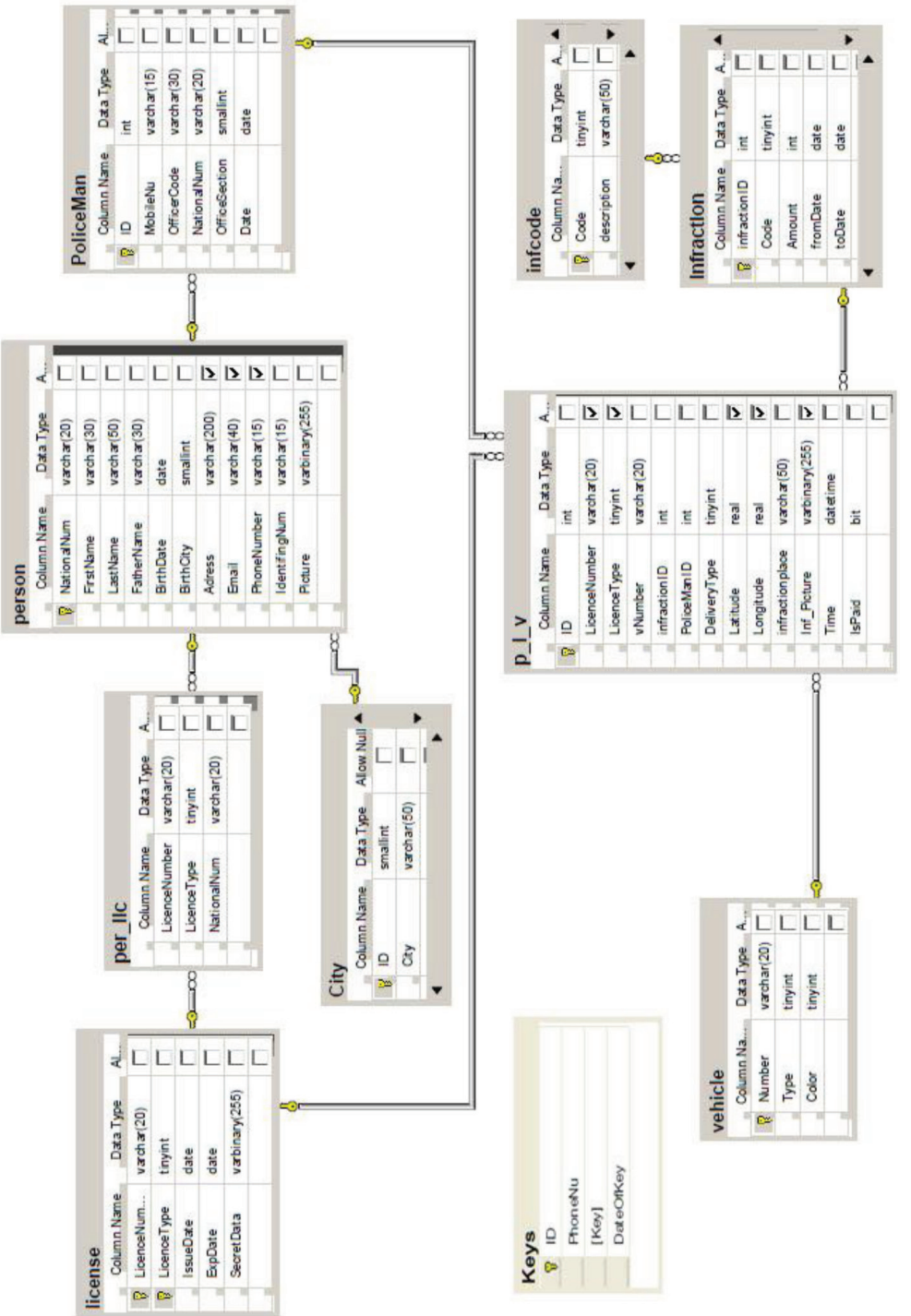


*Fig. 2: Traffic police database ER-diagram*

### Application Server

The Security Gateway and the Police Keys Server are responsible for confirming an officer's authorization, authentication and request decryption. The application server is responsible for relaying officers' request to the database. Officer's requests (queries) can be classified in two groups, namely, officer's inquiries and traffic ticket issuing. Officer's inquiries are classified in four sub-groups, which include penalty amount history, and driving license validation, to check driving license, and vehicle confirmation.

### Security Gateway

The Security Gateway (SG), which is a gateway with security functionalities, resides in TPDC. It provides an end-to-end secure communication between the Traffic Police Data-Centre (TPDC) and its mobile officer (see *Fig. 1*). Security Gateway (SG) isolated MOS and GMS. It is composed of User Authentication Agent (UAA) module and Access Point (AP) module. The AP module provides the necessary interface to the police mobile clients. It communicates with the General Mobile Subsection over TCP/IP and takes the responsibility of receiving/sending messages from/to GMS. The UAA module takes the responsibility of the authentication of the mobile clients.

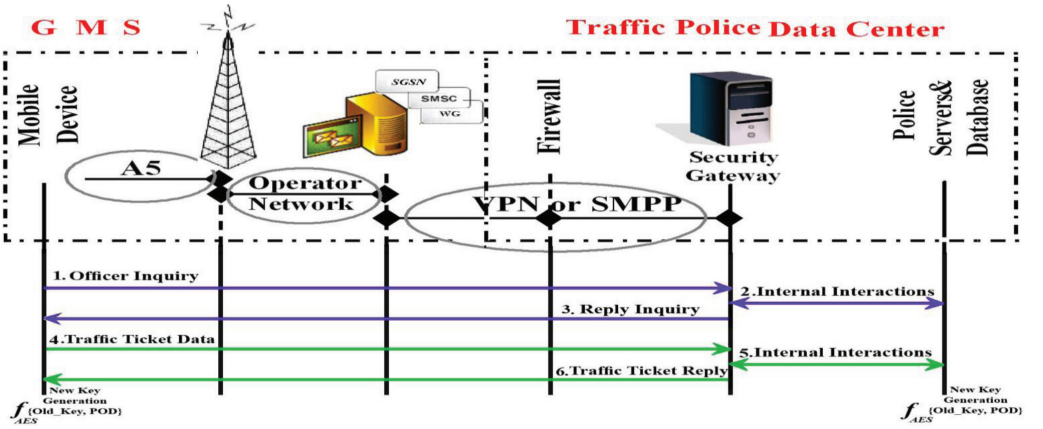


*Fig. 3: Data transaction flow*

### Police Keys Server

The PKS is a trusted party that validates the identities of each entity. Its main functionalities include: (1) generation of new keys, (2) cryptographic operation, and (3) maintaining a database containing valid keys, and Revocation Lists (RLs). The Police Keys Server fetches the key from the *KEYs-DB* pertaining to each request. It provides confidentiality for messages transmission.

Encryption and officer validation Symmetric encryption algorithms have been chosen to provide the necessary cryptanalytical strength and low resources consumption (Potlapally *et al.,* 2003; Gupta *et al.,* 2002; Gebotys. 2004). However, key transmissions on the network have major risk (Mobile Payment Forum, 2003). The use of key distribution techniques, such as divided key methods and submission of a key to trusted neighbour, have also been employed. In this research work, a novel protocol for security improvement has been proposed (Rad *et al.,* 2009). To reduce the risk, the novel method was introduced to enforce the changing of exclusive keys between two parties periodically. This protocol would consider that offender's driving license is an exclusive

data between the Traffic Police Data-Centre (TPDC) and the police officer. The TPDC stores the information pertaining to driving license in the database, and the police officer may obtain the driving license of the offender physically at the infraction venue. *Fig. 3* illustrates the data transaction flow for issuing ticket after an inquiry. The first key is issued to the police officer by the TPDC. The officer will use this key to encrypt the first message (i.e. offender driving license) required to verify the driving license and calculate a hash of the officer ID,

$$\text{Req\_Verify\_License} = \{H(ID_{\text{officer}}), f_{\text{AES}}\ (\text{Driving License})\}$$

First, the TPDC checks $H(ID_{\text{officer}})$ and officer's mobile number, which is used as the fingerprint of the current officer's ID. If the check fails, the protocol will abort the request. Otherwise, it will fetch the key from the PKS server according to the mobile phone number and the TPDC will decrypt the transmitted data and authenticate the police officer's ID. Then, the TPDC will assign a new and temporary ID for this officer session (Daily_session_ID). A response will be sent to the officer to confirm the offender's driving license. This confirmation message consists of license confirmation, Daily_session_ID, date and time session availability. The officer will use the first key for decryption. Once the driving license has been verified, the police officer will issue the traffic ticket, encrypt traffic ticket and Daily_session_ID and send to TPDC. At this stage, the TPDC server and the police officer's mobile phone will generate a new key using the first key and any predetermined data from the offender information (POD):

$$\text{New\_Key} = f_{\text{AES}}\ \{\ \text{Old\_Key, POD}\}$$

This measure provides a high level security transaction, since the key is changing periodically with each new offender. AES is the chosen symmetric algorithm even when considering the mobile resource considerations (Potlapally *et al.,* 2003).

**Mobile Client**

PDAs or Web-enabled mobile devices can use mini Web browsers to access mobile Web applications via wireless Internet connection (Siau *et al.,* 2003). They are preferred for the mobile road traffic infraction registration system. Mobile Web applications are hosted on application servers and can consume Web services on the server-side. They can invoke Web services, integrate responses from these Web services, and then return the consolidated results as Web pages to mobile devices.

*Infraction Registration Benchmark Tool (IRBT)*

The testing tools offer a wide range of functionality needed for performance testing, such as load generation and measurements. Many tools have been developed today for generating workloads on the server. For instance, some testing tools can be found from Banga and Druschel (1997), SPEC (2009), Tschalar (2001) and ServerWatch (2009). They are all based on making repeated requests as quickly as possible at a predetermined rate. For instance, TPC-W is a benchmark from the Transaction Processing Council designed to evaluate e-commerce systems (TPC, 2009). It implements an online bookstore and has three workload mixes that differ in the relative frequency of each of the transaction types. Elnikety *et al.* performed some experiments by using TPC-W and RUBiS (Elnikety *et al.,* 2007) and explored a range of the problem space by varying the size of the database. They focused on TPC-W and scale the database with a small database (0.7 GB), a medium database (1.8 GB), and a large database (2.9 GB). In addition, they varied the memory as 256 MB, 512 MB, and 1024 MB. In another experiment, they used RUBiS Benchmark. In this

attempt, the RUBiS database has a constant load of 2.2 GB. Two workloads were mixed in this experiment containing browsing mix that is read-only and a bidding mix having 15% updates. The work of Deng *et al.* can be seen as another example of the use of the performance tools based on TPC-W Benchmark, in 2004. Their work defined 14 Web interactions in the TPC-W specification. Six of those are read only, while the other 8 update the database. The researchers also defined the database in 8 tables. The test case is based on the sequence of pages to be visited, apart from the input values to be provided to pages containing forms.

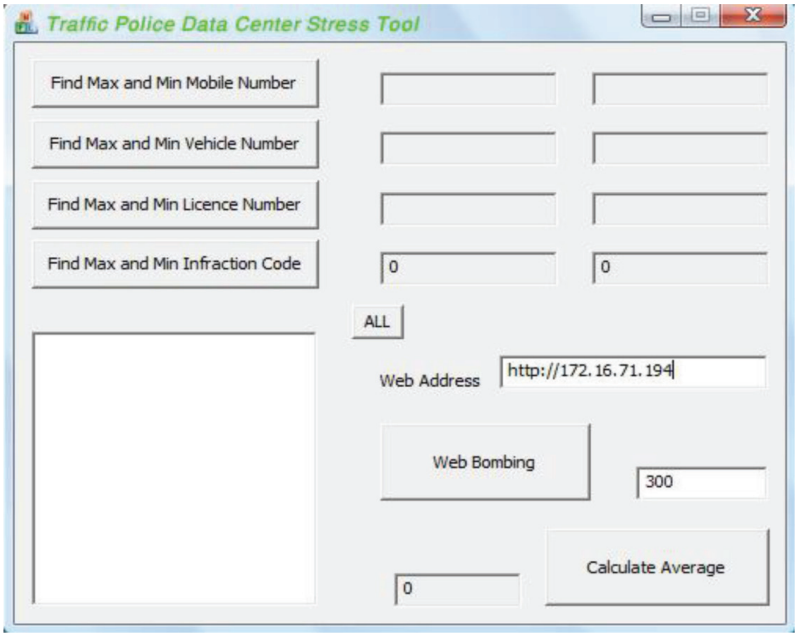


*Fig. 4: A screenshot of Infraction Registration Benchmark Tool (IRBT)*

However, the test tools only send a new request after the server has replied to its previous request and there is also no freedom of action to change the load-generator model. In order to obtain an in-depth information about the behaviour of the server, the authors needed to use a specific tool to emulate the behaviour of the clients. For the purpose of the experiments, a tool that is familiar with the traffic infraction registration system. A closed form benchmark tool was used for the evaluation of the system performance. Infraction Registration Benchmark Tool (IRBT) is a powerful HTTP-client/server test application that was designed to exactly determine the critical performance issues in the Traffic Police Data Centre that might prevent an optimal traffic ticket registration for officers (Arasteh-Rad *et al.,* 2009). The IRBT was designed and implemented to simulate the real-world traffic ticket generation. *Fig. 4* illustrates a screenshot from the IRBT.

The tool generates independent officer's requests. Each random request includes police's phone number, vehicle number, driving license number, delivery type, infraction place and infraction code (see *Fig. 5*). It continuously submits requests to the server, waits for a period of time after the server has sent a reply to the request (processing time), and then submits a new request. The proposed benchmark tool can simultaneously generate from a few to several hundred requests at the same time. Each emulated traffic ticket issued is called a virtual traffic ticket, which is the key of the load-testing concept in the current study.

| Request Name | Request Type | Officer Code | Police Phone Num | Delivery Type | Infraction Place | Vehicle Number | Driving License Num | Infraction Code | Date Time |
|---|---|---|---|---|---|---|---|---|---|

*Fig. 5: Traffic ticket format*

The actual flow of control for the IRBT data generation is illustrated in *Fig. 6*. In the first step of the workflow, the Infraction Registration Benchmark Tool performs to estimate from the data in the INFRACTION-DB. The benchmark tool can find data fluctuating boundaries in the database related to the police's request fields. After finding the data range, the Infraction Registration Benchmark Tool randomly generates the request fields. This is followed by the total requests (traffic tickets) issued per term, the number of concurrent requests, and average request size and URL. The total request issues per term refer to the total number of requests made by the Infraction Registration Benchmark Tool during the test. 'Concurrent requests' in the tests also refer to the case, where two or more simultaneous requests for traffic ticket registration, which could be different, are sent to the server. This corresponds to the case where multiple officers send traffic tickets at the same time. The server URL is specified manually. The interval between the requests generation can be set optional such as manually or with a probability distribution function. Finally, when the total number of generated requests is more than the total number of requests, proceed will be finished. The number of hits and time to the last byte are important metrics for the measurement. The numbers of hits are the total number of requests made by Infraction Registration Benchmark Tool during the test run. TTLB (Time to Last Byte) is the total time, in milliseconds, that last byte of the last request is responded by the server and the client connection will be closed.

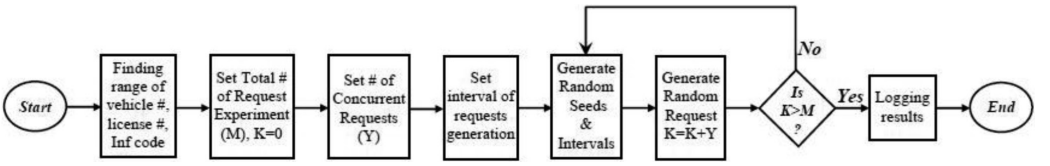


*Fig. 6: Infraction Registration Benchmark Tool (IRBT) flowchart*

## EXPERIMENTAL METHODOLOGY

Hardware, network and external load are basically elements that influence on the performance. These elements are given by information such as server hardware configuration, network bandwidth, database size, average size of the files served and concurrent arrival requests. The effect amounts of them depend upon the system bottlenecks. In the mobile traffic police infraction registration system, the database size and the concurrent arrival requests variations are notably important elements. For example in Iran, more than forty two million traffic tickets were issued in 2007 (Royanian, 2007). Therefore, the police database size increases every year, and the massive access to the system is also expected at the same time. Moreover, the database records should be accessible for police officer's queries. Here, the system was assumed to have enough network bandwidth.

In the following sections, the main parameters in the IRBT used in the simulation test in the benchmark, configuration parameters in the IRBT benchmark, test-bed in the proposed servers and monitoring the server performance behaviours are discussed.

*Performance Metrics of the Experiment*

The IRBT supports various types of tests contain, load tests, stress tests, and ramp tests. The number of simultaneous requests on one URL raises during the ramp tests processes, i.e. from a single request to hundreds of requests. Ramp tests can be considered, such as a set of variations of the stress tests. Stress tests simulated excessive load to server. In this study, the focus was given on the ramp tests, while stress test, such a part of ramp test, was also highlighted.

In the experiment carried out in the current study, the system performance evaluation was based on the several factors, such as CPU load, physical memory usage, limitations of the number of users, response time and the number of users accessing to server at a given time. Any one of these factors may degrade the performance of the server. If any resource experiences a situation resembling a bottleneck, it can be a restriction for the system. For example, if CPU utilization is optimized at 100% during normal load hours, then there will be no capacity to handle a peak load.

*Test Environment*

It is important to highlight that the same hardware was used for all the experiments. Each one contains Intel® Pentium TM Dual Core 2.40 GHz 1 MB Advanced Transfer Cache, 2048 MB DDR2 RAM, and a 320 GB Serial ATA hard drive (7200 rpm). They are connected via multiple 24 port gigabit switches and run a stock Linux Ubuntu 8.10. The TPDC was implemented based on Apache version 2.2.11 as the proposed Web-server, the PHP v.5.2.8 module to provide server side scripting for generating dynamic content and the MySQL v.5.1 database server with a default configuration. The Load-Generator was developed based on Java. Meanwhile, the database server and the application server run on a computer.

*Test IRBT Criteria*

The IRBT with the parameters under the previously mentioned criteria was applied in the current test to help us evaluate the performance of the server under workload. The total requests issued per term, the number of concurrent requests, intervals and average request size were configuration parameters that are fixed in the tests. The system performance was evaluated by increasing the concurrent requests on six steps according to the sizes of the database. In each step, the database size increased about one thousand twenty hundred sixty (1260) Mega-bytes (i.e. approximately equivalent to ten million traffic tickets). In the experiment, the authors started with 10 concurrent requests and gradually increased the number of the concurrent requests to 180. The experiment for each concurrent step was repeated three times. The time difference between the time when a request was sent and when a successful respond was completely received specified the response time. With respect to the testbed of the current study, the average service time, average CPU load, and average physical memory usage were calculated as the sample mean of the observations.

## DISCUSSION AND RESULTS

The scenario ramp test was executed to determine the initial maximum load that the system resource could sustain. In the experiments, this was started with 10 concurrent requests and the number of the concurrent requests was gradually increased to 180 in six steps, according to the sizes of the database. The experiment for each concurrent step was repeated three times. The mean response time, the mean CPU utilization and the usage of RAM were measured. *Fig. 7* and *10* illustrate the relationship between the growth of the database size, service rate and service time in various concurrent arrival requests. Meanwhile, *Fig. 8* and *9* demonstrate the correspondence between the

concurrent arrival requests, service times and service rates in various database sizes. These figures explain that the concurrent requests increase the service rate for each database size, from 10 to 60. As shown in *Fig. 11*, a considerable amount of CPU utilization of the server is also increased. These figures also reveal that the service rate can considerably be reduced for each database size from 100 to 200, specifically when the database size is already large. This is due to the concurrent control cost such as context switch and logs. At this stage, CPU is utilized at 100%, and it has more no capacity for handling a peak load.

*Fig. 11* shows that CPU utilization increases almost linearly with the number of requests. This figure also reveals CPU can be the bottleneck resource, especially the peak throughput point. It jumps from 57% CPU utilization for 20 concurrent users to 100% with 90 concurrent users in the peak of database size. With more requests, the situation reverses and the CPU on the server becomes the bottleneck. *Fig. 12* shows that memory utilization also increases correspondingly. *Fig. 13* illustrates network utilization when the number of concurrent requests is set to the maximum. This particular figure also illustrates the network was used less than 2.5%.

Let consider that when the concurrent arrival is set to 20 in the peak of database size as a stress point. The results of this point can be observed as a stress test. The server could handle each request in 48.19277 milli second. Therefore, the mean service rate is approximately 20.75 per second. As shown in Figures 14 and 15, the CPU was used at around 64.14% and the usage of RAM was low, i.e. around 38.57%. *Fig. 16* depicts open and abort connection statistics in this point. Disk usage on the server is reported in *Fig. 17*. At the beginning of the experiment, when the first requests started to arrive at the server, there are a lot of disk activities and server processes. This explains the initial disk activity and the sharp rise in memory use, until the point in time at which, most of the working sets (indices and frequently used tables) are in memory. After that, there is little disk activity and the memory usage remains stable at around 39 %. Hence, it could be concluded that the resources were used moderately most of the time and could not lead to a limitation with 20 concurrent users. The service rate is also more than the arrival rate, and therefore, the average queue size is bounded and the system responds instantaneously.

## CONCLUSIONS

The performance evaluation of the mobile traffic infraction registration system is crucial, particularly due to the fact that there are a lot of traffic tickets issued in the recent years and the significance of police telecommunication. It is concluded that the most results were derived from the proposed methods and procedures through the application of the IRBT benchmark on mobile traffic infraction registration server. The results achieved are very favourable, and the current empirical study has led to the following conclusions.

It is clear that the environmental constraints can be placed on the server by benchmark tool before the application is deployed for assuring that the mobile road traffic infraction registration system is always reliable and stable at the expected level, even under critical loads.

The server is tested using the ramp test. However, the study considered that the concurrent arrival was set to 20 in a peak of the database size as a stress point. The scenario ramp test was executed to determine the initial maximum load that the system resource could sustain. The results gathered from the ramp test showed that the concurrent requests would increase the service rate for each database size from 10 to 60. It also increased a considerable amount of CPU utilization of the server. Meanwhile, the service rate could considerably be reduced for each database size from 100 to 200, especially when the database size was already large. This is due to concurrent control cost, such as context switch and logs. At this stage, CPU is utilized at 100%, and it has no more capacity for handling a peak load. Therefore, the increase in response time can lead to a

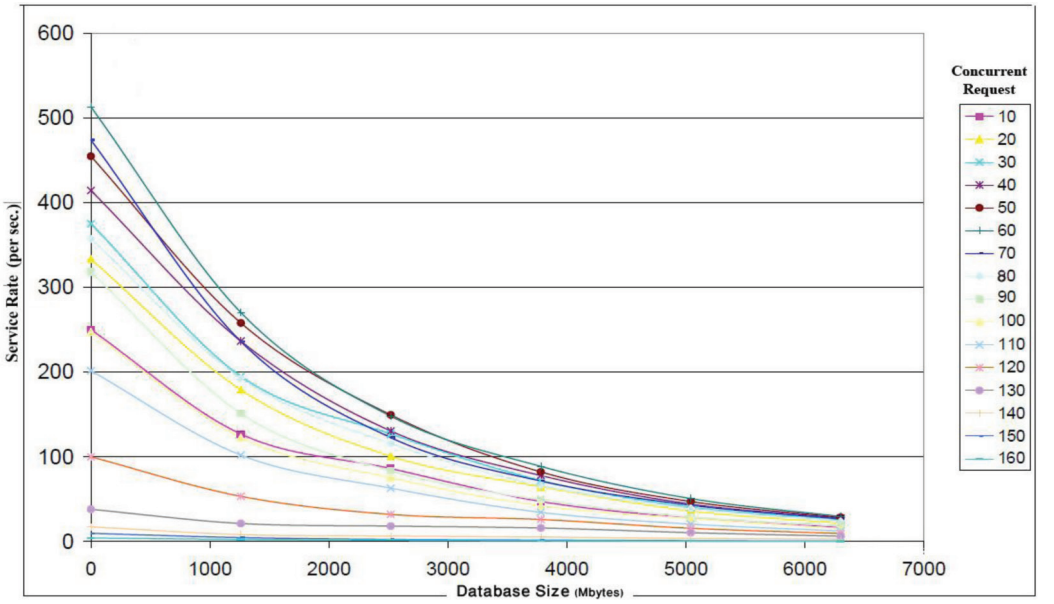


*Fig. 7: The experiment results for the correspondence between growth of database size and service rate in various concurrent requests*

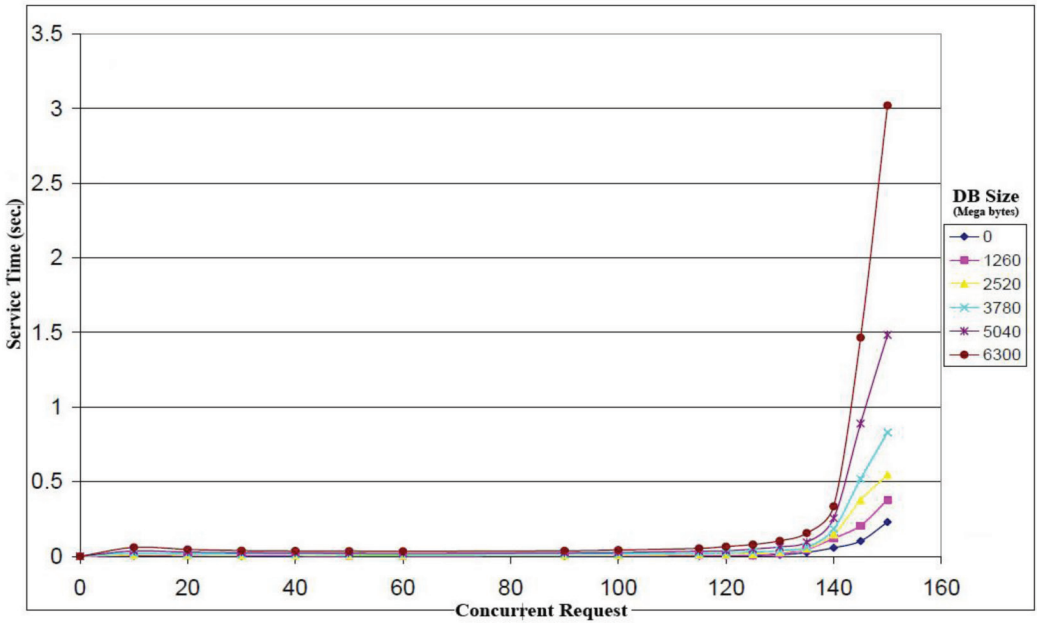


*Fig. 8: The experiment results for the correspondence between concurrent arrival requests and response time in various database sizes*

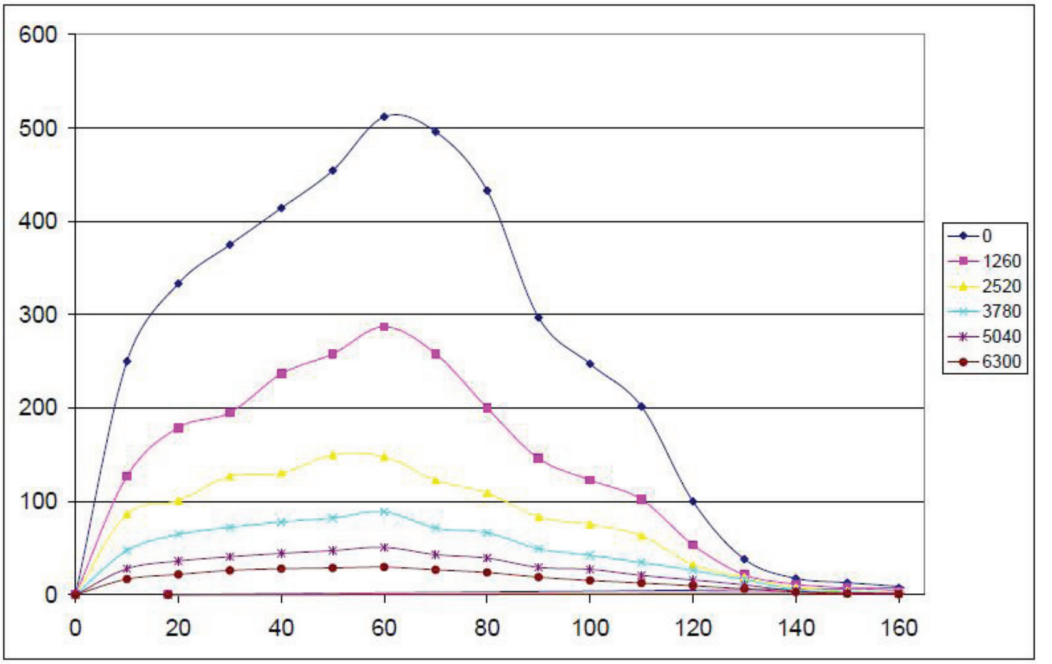


*Fig. 9: The experiment results for the relationship between growth of concurrent arrival requests and service rate in various database sizes*

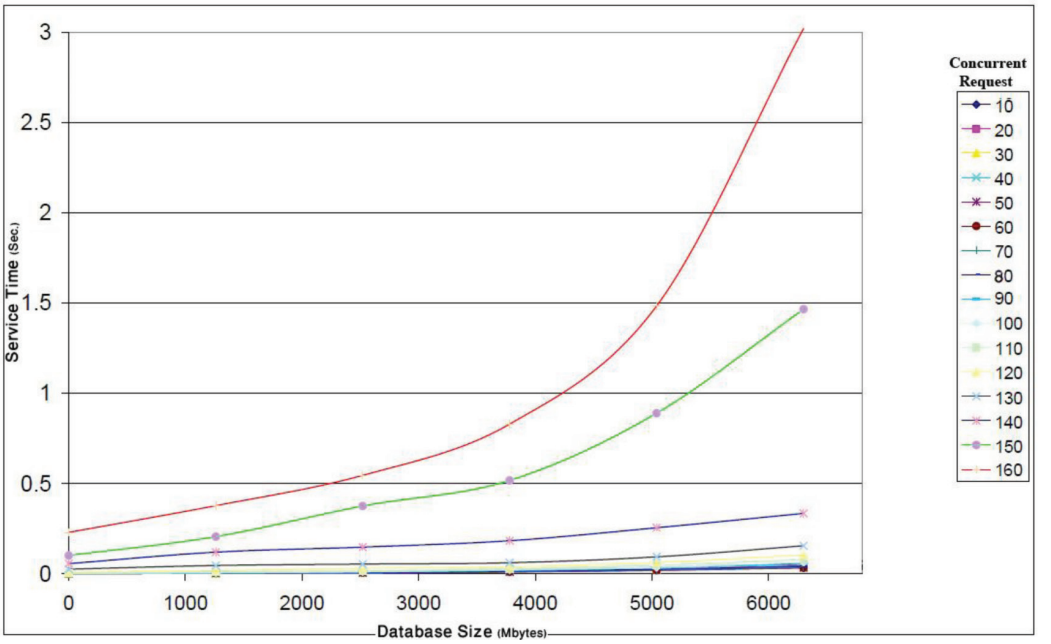


*Fig. 10: The experiment results for the relationship between growth of database size and service time in various concurrent arrival requests*

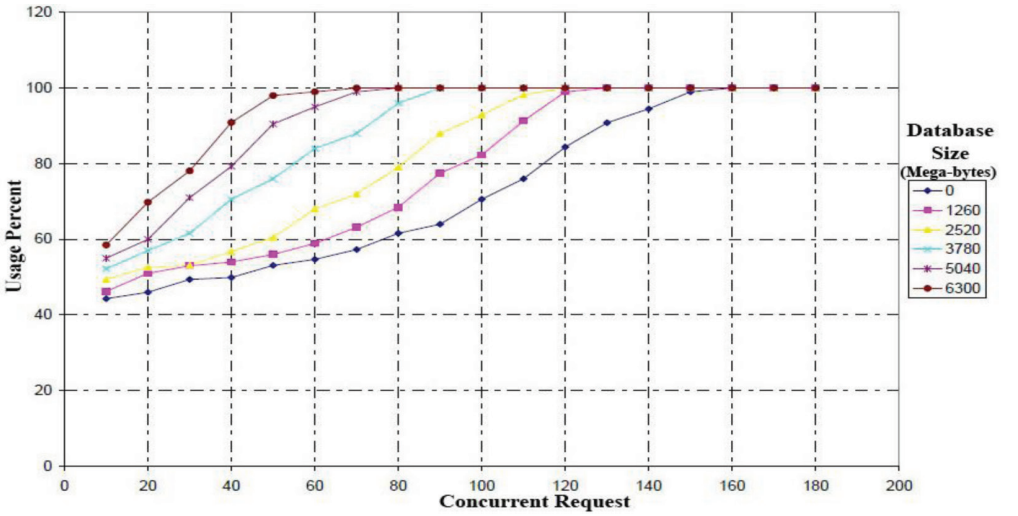


*Fig. 11: CPU utilization in the ramp test*

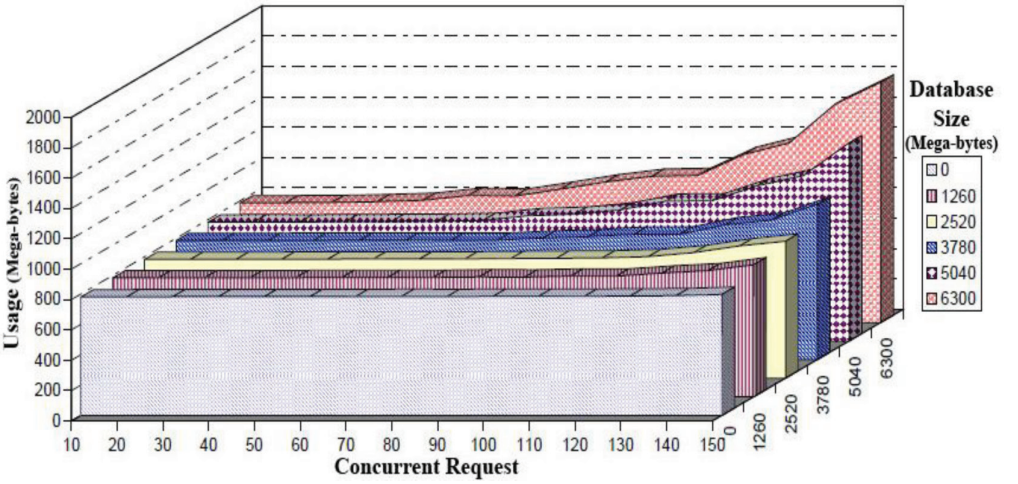


*Fig. 12: Physical memory utilization in the ramp test*

point where optimization of the system is unavoidable. The metrics can be further improved by upgrading hardware to professional servers. In the first step, the amount of RAM on the server could be increased if the attempt was to achieve high performance under workloads.

This manuscript has also shown CPU can be the bottleneck resource, specifically the peak throughput point. In the stress point, the server could handle each request in 48.19277 milli seconds, so the response time is 20.75 per second. The CPU was used at around 64% and the usage of RAM was low (i.e. around 38%). Therefore, it could be concluded that the resources were moderately used most of the time and the server could approximately handle triple of the maximum average arrival rate in the real time. Finally, the results provide a clear guideline for performance evaluation of mobile road traffic infraction registration system.

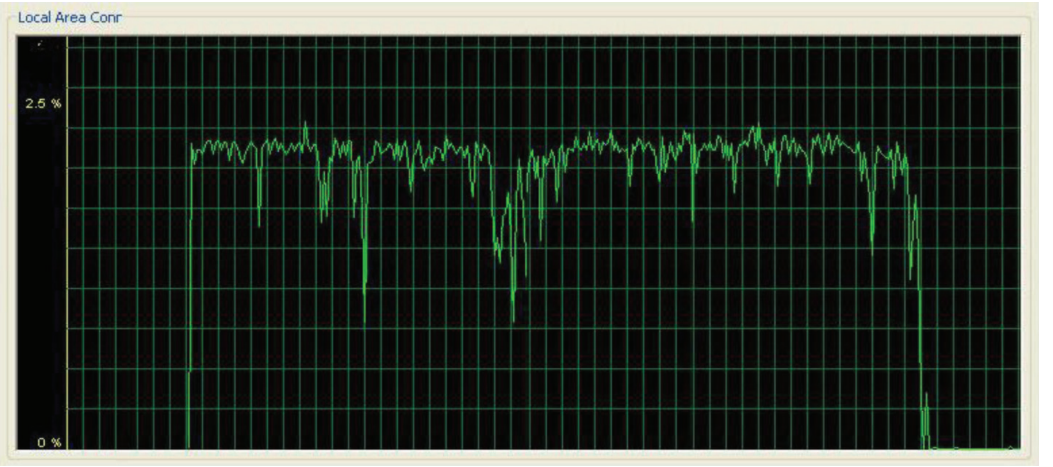


*Fig. 13: Network utilization*

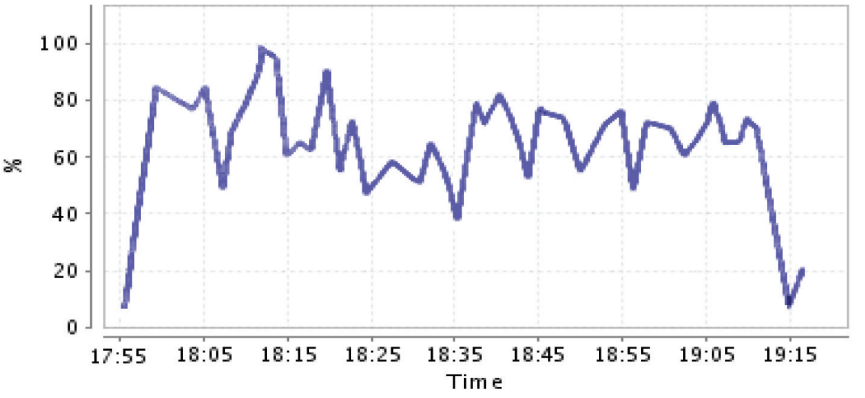


*Fig. 14: CPU utilization in stress test*

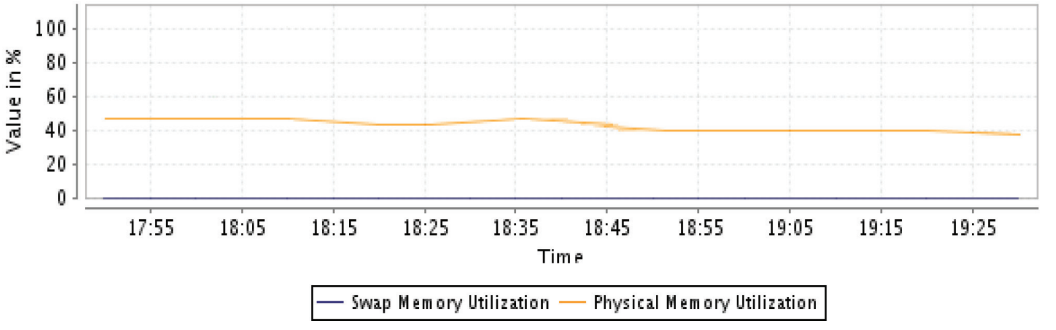


*Fig. 15: Physical memory utilization in the stress test*

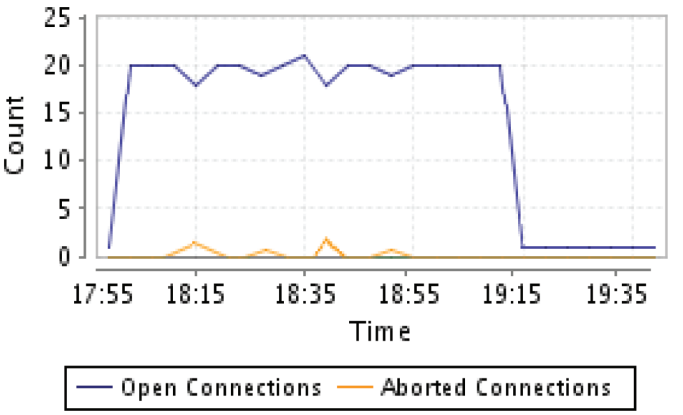


*Fig. 16: Connection statistics in the stress test*

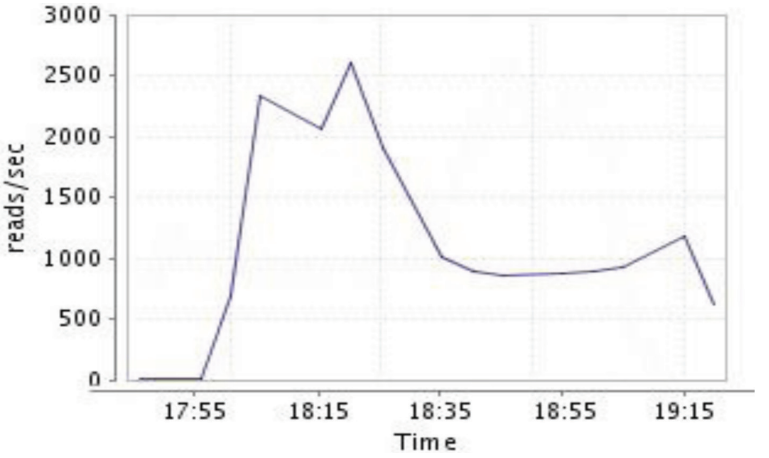


*Fig. 17: Disk I/O statistics in the stress test*

## FUTURE RESEARCH

The data flow over the public network and this implies that the TPDC is exposed to an unprotected environment. Therefore, a continuation of the work, or an additional research should be performed on the security evaluation of the mobile road traffic infraction registration system. Security is another major issue that it is associated with the mobile road traffic infraction registration system. The police data are almost critical or confidential. Hence, to enhance and justify the security level, a future possible research can identify the security evaluation of the mobile road traffic infraction registration system.

## REFERENCES

3G Americas (n.d). GPRS: General packet radio services. Retrieved on Febuary, 2008.

Abdelazez, S. M. (2000). *Web server evaluation by using simulation model*. (Unpublished Doctoral dissertation), Egypt, 2000.

Ahmad, S. K., & Azmi, A. E. (2007). Road evolution in Malaysia: from footpaths to superhighways. Roads Branch, Public Works Department, Malaysia.

Anwar, K. &, Saleem, A. (2004). Web Application Stress Test and Data Analysis. Retrieved from http://www.webstar.co.uk/Stress.pdf

Arasteh-Rad, H., Samsudin, K., Ramli, A. R., Tehrani, M. B., & Tavalai, M. A. (2009, April). *Securing mobile communication system for traffic infraction registration*. Paper presented at International Conference on Future Computer and Communication, Kuala Lumpur, Malaysia.

Banga, G., & Druschel, P. (1997). *Measuring the capacity of a web server*. Paper presented at Proceedings of the USENIX Annual Technical Conference, Monterey, CA.

Cellular-News. (2008, March). Global mobile phone subscribers to reach 4.5 billion by 2012. Retrieved from http://www.cellular-news.com/story/29824.php

Cronkhite, C. L. (1974). Automation and Law Enforcement. *Charles C Thomas*.

Deng, Y., Frankl, P., & Wang, J. (2004). Testing web database applications. *ACM SIGSOFT Software Engineering Notes, 29*, 1–10.

Dilley, J. (n.d). Web server workload characterization, Hewlett-Packard Laboratories, 2002. Retrieved from http://citeseer.ist.psu.edu

Elnikety, S., Dropsho, S., & Zwaenepoel, W. (2007). Tashkent+: memory-aware load balancing and update filtering in replicated databases. *SIGOPS Oper. Syst. Rev., 41*(3), 399-412.

Gebotys, C. H. (2004, 8-10 Sept. 2004). *Low energy security optimization in embedded cryptographic systems*. Paper presented at the Hardware/Software Codesign and System Synthesis, 2004. CODES + ISSS 2004. International Conference on.

Gupta, V., Gupta, S., Chang, S., & Stebila, D. (2002). *Performance analysis of elliptic curve cryptography for SSL*. Paper presented at the Proceedings of the 1st ACM workshop on Wireless security, Atlanta, GA, USA.

Informa Telecoms, & Media (n.d). World Cellular Data Metrics. Retrieved November, 2008 from http://shop.informatm.com/marlin/30000001001/MARKT-EFFORT/marketingid/20001131760

Iran Daily (2007, January). Reducing Road Accidents. Retrieved from http://irandaily.com/1385/2777/html/focus.htm

Labordere, A. H. (2006). *SMS and MMS Interworking in Mobile Net*. Artech House Publishers.

Mobile Payment Forum. (2003). *White Paper: Risks and threats analysis and security best practices*.

MobileTracker. (2005, May). The total number of mobile phone subscribers. Retrieved from http://www.mobiletracker.net/archives/2005/05/18/mobile-subcribers-worldwide

Othman, M. R. (2006, November). Highway network development plan for Malaysia. *The World Road Association-PIARC*. Retrieved from http://www.piarc.org/en/

PCWorld. (2006). Mobile subscribers to reach 2.6b this year. Retrieved from http://www.pcworld.com/article/127820/mobile-subscribers-to-reach-26b-this-year.html/

Potlapally, N. R., Ravi, S., Raghunathan, A., & Jha, N. K. (2003, 25-27 Aug. 2003). *Analyzing the energy consumption of security protocols*. Paper presented at the Low Power Electronics and Design, 2003. ISLPED '03. Proceedings of the 2003 International Symposium on.

Qusay, M. (n.d). WAP for Java Developers: Develop WAP Applications with Servlets and JavaServer Pages. Retrieved June, 2007, from http://developers.sun.com/techtopics/mobility/enterprise/articles/wap/intro/

RAD. (2000). Applications for WAP. Tech. Rep., 2000. Retrived from http://www2.rad.com/networks/2000/wap/apps.htm.

RAD. (n.d). What is WAP?. Retrieved on January, 2009 from http://www.pulsewan.com/data101/wap-basics.htm.

Rad, H. A., Samsudin, K., Ramli, A. R., & Tehrani, M. B. (2009). *The Design and Implementation of a Stress Tool for Traffic Infraction Registration System*. Paper presented at the Proceedings of the 2009 Second International Conference on Environmental and Computer Science.

Royanian M. (2007, November 16). 45m tickets for reckless drivers. *Iran Daily*.

Santra, A., Krishna, M., & Das, A. (2009, Sept). Measurement of memory usage in j2ee applications. *Journal of Scientific and Industrial Research, 68*, 786-788.

ServerWatch (n.d). Server compare. Retrieved on 2009 from http://www.serverwatch.com/stypes/compare/

Siau, K., Shen, Z., & Varshney, U. (2003). Communications and mobile services. *International Journal of Mobile Communications 1*, 3–14.

Standard Performance Evaluation Corporation (n.d). SPECweb2009. Retrieved on 2009 from http://www.spec.org/web2009/

Standard Performance Evaluation Corporation, SPEC (n.d). Retrieved on 2009 from http://performance.netlib.org

Takahashi, H., Suganuma, T., & Shiratori, N. (2005, 20-22 July 2005). *AMUSE: an agent-based middleware for context-aware ubiquitous services.* Paper presented at the Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on.

Transaction Processing Performance Council, TPC (n.d). TPC-Benchmark W. Retrieved on January, 2009 from http://www.tpc.org/tpcw/default.asp

Tschalar, R. (2001, May). HTTPClient Version 0.3-3. Retrieved from http://www.innovation.ch/java/HTTPClient

Turettini, E. (2006). Up to 90 percent of globe to have mobile coverage. Retrieved from http://www.textually.org/textually/archives/2006/10/013841.htm

Whale Communications Ltd (n.d). The e-Gap remote access appliance secures SSL VPN technology for providing remote access from anywhere. Tech. Rep. Retrieved on March, 2003.

Zekavat, S. R. (n.d). Road safety - government policy. Regional Experiences and Lessons in Financial Highway Infrastructure and Improving Road Safety, Bangkok, May 2006. Retrieved from http://www.unescap.org/ttdw/roadsafety/StatusPapers2006/IslamicRepublicofIran-statuspaper.pdf

Zhu Han, & K. J. Ray  Liu (2008). Resource Allocation for Wireless Networks, Basics, Techniques, and Applications. *Cambridge University Press* vol 2.