

Mobile Information System, How to Build with Case Study

[doi:10.3991/ijim.v4i4.1357](https://doi.org/10.3991/ijim.v4i4.1357)

R. M. H. Al-Sayyed, S. Manaseer and O. Rababeh
The University of Jordan, Amman, Jordan

Abstract—In a distributed computing environment, comprising mobile computers and wireless networks, users change their locations frequently while still expecting “acceptable” levels of performance. This complexity will produce poor performance with no exact answer why? In such an environment, a new method of designing applications will be required.

It is claimed that understanding network performance is an art rather than a science because in practice there is little underlying useful theory; so, experience with real world examples are used instead to develop rules of thumb [14].

This paper is fundamentally designed to describe types of mobile applications and the major performance factors needed to be considered when designing such kind of applications. Methods of connectivity: always connected, occasionally connected, and occasionally disconnected will be addressed as well. We are proposing a Mobile Information System (MIS) model. This model will address the needs for monitoring wireless environment where the mobile applications running in this area and how it is monitoring the performance factors. Also provided in the model are the features of the MIS, the pros and cons of the model, and a simulation to show the effectiveness of the model.

Index Terms—MIS, QoS, AuthXtion, Connectivity.

I. INTRODUCTION

A mobile application is one that requires devices such as PDAs, laptop computers, or phones where you can use it anywhere you move. It is not necessary to have wireless communication capability for mobile applications. Wireless applications are either thin client (WAP) or thick client (developed by languages such as: Java and C); they require wireless connection. By nature, mobile applications are small, mobile, and have a personal quality. Mobile users have special application needs. Issues such as user interface, display size, data entry problems, the lack of wireless connections, and security must be addressed during the development of such applications. The process of interchanging mobile information is called syncing and it is commonly used with PDAs [1].

Mobile communication has evolved since the 1920s (private mobile and radio). Some mobile-data technologies were introduced in the 1980s (Ericsson’s Mobitex and Motorola’s ARDIS). In the early 1990s, Cellular Digital Packet Data (CDPD) was introduced [9]. The introduction of this technology made wireless data applications common in police departments, emergency management staff, and delivery businesses. The market is expected to develop rapidly as networks mature, the price of wireless

bandwidth drops, and mobile computing platforms become capable of supporting industrial-strength applications [1].

Some researches [3] showed that, in some environments, some applications are user-specific; users for such applications are divided into disjoint location-based sub-communities, each with its own movement, activity, and usage characteristics. Most users exploit the network for web-surfing. The high numbers of users mainly use mobile computers for synchronous communications.

Another important issue is that a single user and application usually cause peak throughput in the mobile network. Also, whilst the overall incoming traffic dominates the outgoing traffic; the reverse situation holds during peak throughput. Peak load can be caused by factors such as the transfer of large volumes of data and increased numbers of users. For interactive application (the goal of this study), software optimization can help in solving the throughput problem. On the user side, latency is a major factor in a user’s satisfaction. Latency for wireless networks is affected by the kind of network structure, the connection medium, and the efficiency of the application itself.

The author of [7] indicated that wireless application developers are not knowledgeable with embedded platforms because they often come from a PC background. As such, they need to be aware of the differences they will find in the new hardware and software environment. Developers who are aware of the differences among the media transmission rates, memory resources, user interface, power requirements, processing core architectures and development environments can plan and execute their software more effectively.

In a study about workflow partitioning in mobile information systems in [2], the authors described a partitioning approach for the execution of Business Process Execution Language (PBEL) onto a network of mobile devices. The produced execution model is “equivalent” to the centralized one with a support to the components that are disconnected and independent workflow engines.

In [4], the authors described the AmbieSense system they developed which was designed to meet people’s information needs by providing ambient, personalized and context-aware application style. In order to address users’ information need, the authors aimed to enable a more seamless interaction between the mobile users, their physical environment and the situation they encounter. They built and tested the content, software, hardware, architecture and the interaction with users. The approach they followed is useful to consider when developing information systems that also scale up.

The authors of [10] described a rule-based event-driven modular framework for building self-adaptive applications in mobile environment. They employed a “mobile event engine” for all applications to detect changes. In order to evaluate the potential of the approach, they provided analytical characterization and detail simulation. In order to show the feasibility and effectiveness of their methods, they provided a modular prototype implementation and performance results.

The rest of this paper is organized as follows. Section II covers the concepts of developing a mobile/wireless application, the considerations that must be taken to develop the applications; the connectivity methods, and a brief introduction to synchronization techniques. Section III covers the common features of the MIS, the hardware architecture, the software architecture, and the user view of the system. Section IV covers the design and implementation of the Case Study system along with the performance parameters used. Section V covers the pros and cons of the various implementation methods. Section VI covers analysis to WLAN MIS environment and at the end we draw our conclusion.

II. DEVELOPPING MOBILE/WIRELESS APPLICATIONS

Mobile application developers face many challenges during the development process. Limitations such as bandwidth, nontraditional computing devices, lack of standards, security issues, and transactional integrity [8] affect the design. Mobile developers must also be familiar with the target device's specifications.

The following steps should be considered when developing a mobile application [5]:

1. Understand the user. The *mobile user* helps in specifying the real application. The *interaction designer* creates the typical user (called persona).
2. Build a mobile database with content: The information design is developed.
3. Design a logical application: simulator, screens are built, and while attached to real content, the business logic is coded. The major components of the application are Wireless field application, Back office administrative application and Design a Web PC start page if needed.
4. Use a device: try the application on a device as described in the persona.
5. Check Quality of Service (QoS) guarantees: QoS consideration is very important when designing applications for low bandwidth and congested mobile environment. Although QoS can be provided at system level, operating system and application levels, providing QoS at application level only is highly important especially when considering the mobile environment where QoS parameters are hardware independent and operating systems are being improved. While expanding the bandwidth can be a clear solution, its cost is not trivial [11], and this implies the ability to control and manage the resources.

Designers should consider the connectivity models when designing an application. The connectivity models are [6]:

- **Always connected:** All changes by mobile users are stored immediately in the enterprise.

- **Occasionally connected:** the application connects to the network to access the data on occasions where data is required. Synchronization is needed periodically and databases are compact for small devices.
- **Occasionally disconnected:** the application can function with and without connection. Another name for this type of connection is called *always available*. The availability of the server is desirable but not essential.

III. THE MIS FEATURES

The MIS is a combination of hardware and software model for an interactive wireless application/tool. The proposed model is for use in a university wireless environment to help users access needed resources. The emerging mobile computing technologies in communications and hardware have posed many challenges. Users are requesting access to data anywhere and any time from their homes, offices, and while roaming outdoors even though their portable devices have diverse user interfaces and incompatible software. In [12], the authors have proposed the Vivian solution (using Java and CORBA), to overcome the problems of weak equipment, limited input and output, heterogeneous platforms, security guarantees, administration routines, and availability on the fly.

Information management tools of such systems can be specific (e.g. stock control) or meant to interact with other systems in an unpredictable manner [13]. In this context, mobility is of high importance where user can move from one context to another, from one physical location to another, or from one application to another. Information management can be developed to supply real-time database queries if location awareness is considered. In addition, mobility raises additional security problems although mobile computers have added a lot of flexibility to distributed systems, however, distributed databases have traditionally not been very flexible because of the need to synchronize all transactions at all sites [15].

A. MIS environment

As shown in Figure 1, the following constitute the basic hardware structure of the model:

1. Three wireless labs on a university campus or similar environment
2. Each lab has an access point that is *cabled* to a switch that is, in turn, cabled to a LAN.
3. The server of the LAN is connected to another server installed in another location, with access to it through the Internet connection

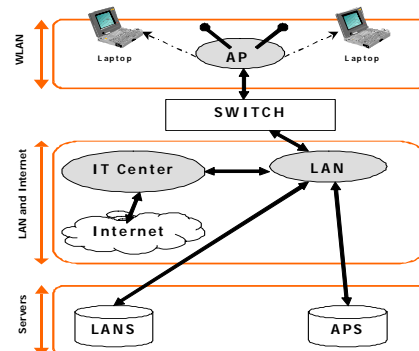


Figure 1. Basic Hardware components

B. MIS Software Architecture

Software wise, the architecture of the system is depicted in Figure 2. It is mainly composed of the following sub-systems:

- Authentication and authorization system: to protect students' data from unauthorized access such as others students or external users
- Students' attendance tracking systems: for both students and instructors. The instructor records the class attendance of students and the students can query their individual attendance records.
- Homework posting and tracking system: this will generate a load on the network for testing purposes; students will hand in their homework assignments to their teachers through this system and get feedback comments about their submitted assignments.
- Logging (registry) system: the main goal for this system is to track all the input and output traffic through the access point to and from the LAN for studying purposes.

In addition, students should be able to access the Internet and all the other available applications such as the Registration system and the library system. Note that the active users can range from 1 to 25 each of whom is running an application. The applications that run can be all the same or a mixture of different applications.

AuthXtion represents the application that is used to validate users' authentication and authorization. Note that X is used in the name to substitute the parts *entica* and *oriza* for both Authentication and Authorization.

Figure 3 depicts the behavior of the homework posting application (our case study) and its relation to the user. The Data that is sent to the system includes Student ID, Date, Time, Course ID, and Assignment number. The query results are: Student ID, Course ID, Assignment number, and instructor's comments.

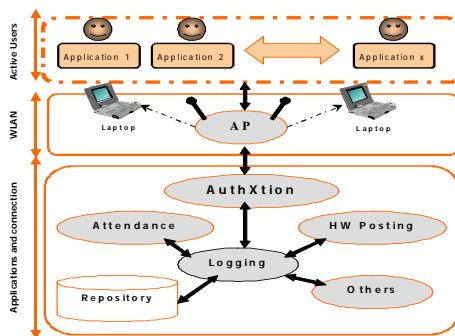


Figure 2. Basic Software components

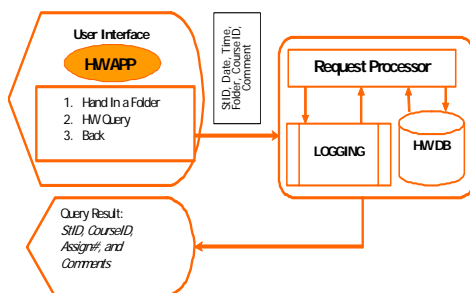


Figure 3. HW Posting System Architecture

IV. MIS SYSTEM DESIGN

A. Requirements Analysis Phase

1. The application should validate user' identity for privacy of data
2. The application should display a menu to the student to select from with the options: (a) Hand In a file, (b) HW Query, (3) Exit
3. The fields of interest are: StID, Date, Time, CourseID, Teacher, FileName, Path, Grade, Comments
4. The log information (internally) should be like: HWStartTime, HWEndTime, SizeInBytes
5. The application should allow the user to query about the grading of the homework.

B. MIS Design, Assumptions, and Preparation for other Phases

The system allows users to login securely, select and perform the needed operation such as posting or querying about homework grading. The following assumptions were made for purpose of study:

- a. The application keeps track of only one homework assignment per student.
- b. The system considers the student id the same as the UID

C. Addressing the Performance Factors:

The following factors affect the overall performance of mobile applications and will be studied as follows:

D. Response time

This work measures the time needed for both "hand in" and "query". Measuring this factor is done by recording the time (in microseconds) of the hand in process until the homework is delivered and an acknowledgement is recorded.

E. Throughput

The throughput between two points (point 1 & point 2) is the amount of data sent from point 1 to point 2 (acknowledged) during a certain time.

F. Privacy and Security

To minimize the effect on overall performance, a simple security mechanism is included in this work. Security is evaluated by comparing the overall time to complete an operation with and without it.

V. PROS AND CONS OF MIS IMPLEMENTATION

WSTTCP software [16] and the QCHECK software [17] are used for measuring the TCP and UDP performance between two systems; we use them here to evaluate the MIS model. For instance, due to the small size of mobile screens, designing an interactive interface for users is not an easy job. If the processing of a command is timely, that will affect the overall application performance. It is not always possible to optimize algorithms for both performance and memory at the same time. Communication costs for a mobile device are more than those of its wired counterpart.

To evaluate the model, the basic application factors should be explained in more details. As part of designing a mobile application, we will look at the factors: display size, restrictions on input facilities, computing power, energy supply, limited bandwidth, and network access. Because these factors are related to each other, considering all of them at the same time is problematic. For instance, due to the small size of mobile screens, designing an interactive interface for users is not an easy job. If the processing of a command is timely, that will affect the overall application performance. It is not always possible to optimize algorithms for both performance and memory at the same time. Communication costs for a mobile device are more than those of a wired one, so we choose to use synchronization and replication when possible to avoid on-line connection since it is costly. In wired communication, the user doesn't care much about the bandwidth or power whereas these are major considerations for mobile applications.

A. MIS Evaluation

In the design of MIS, the logging application is a bottleneck of the proposed model; failure of the logging application indicates the failure to the whole system. The same applies to the **authXtion** application. Even though the logging subsystem is the bottleneck of the system, the whole system shouldn't allow unauthorized users to access the system.

If the environment does not accept any failure, another recovery server can be employed to take over the authXtion turn. By installing more servers, traffic generated to guarantee consistency might degrade the whole performance. If the authXtion server is having a second backup server and the authXtion is crashed for any reason, recovering from the crash might lead to unacceptable performance especially when the second server starts upgrading the authXtion for consistency.

On the other hand, if the logging application (server) goes down, each application will record the logging in its repository until a successful commit is done. Committing is done automatically if the process succeeds. If for some reasons the process does not succeed, the previous logging information will be recorded in the failures log (separately) and the operation itself will be rolled back. After a successful commit or rollback, the operation-auditing log should be removed (cleared) making room for other operations. All these operations will be recorded on the proper server not the laptop itself to give faster access to laptop users and for better auditing process. This means that the user can access the required application without waiting until the logging application server is up again. All logging information is kept on the various servers until the logging server is up at which time all the logging information is transparently transferred to the logging server. After the logging server acknowledges the other servers' auditing information is recorded successfully, these servers can then remove the auditing information properly.

The assumed connectivity method is always connected with some of the advantages of always available (when the logging server is down). Any application maintenance will not affect the user. We separated the interface from both the data and logic layers making the maintenance simpler on the server side where the application resides. So, any application maintenance will not affect the user.

As we employed minimal number of menus, processing of the selected commands is simple enough for users to work with the application and make the system user-friendly. Also, since the type of devices used is only laptops, many standards can be employed to simplify user's life. The mobility of the user does not need special care in implementing the application since students are moving within a limited area inside one campus. Our MIS model can be integrated easily with other applications. The new application will be installed on a proper server and easily be called by adding another entry for it in the model's menu.

VI. ANALYSIS TO WLAN MIS ENVIRONMENT

A. The environment:

Figure 4 shows a summary of the hardware equipment and the WLAN configuration. The access point (AP) is installed on the second floor inside a room with all the other equipments. In some tests, the laptop was in the first floor.

B. Throughput Results

Figure 5 shows the throughput when Wired Equivalence Privacy (WEP) is disabled and Figure 6 shows the throughput when WEP is enabled. WEP is with 40-bit encryption key.

It is not difficult to determine (by looking at the two figures) that the overall UDP throughput is higher in both cases even though it is started less. Notice also the expected results that with WEP, throughput becomes less due to the security overhead. The results were recorded 5 times and the average was taken to be realistic.

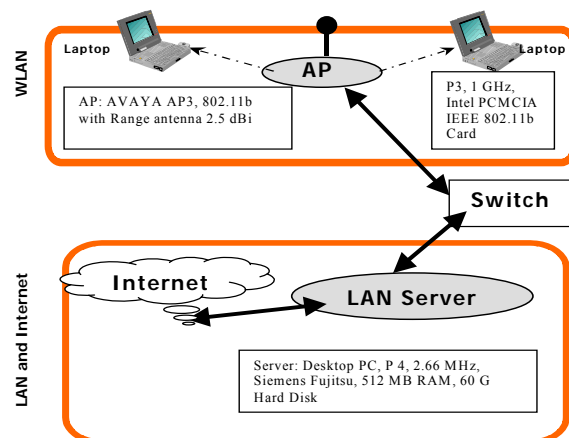


Figure 4. WLAN Configuration.

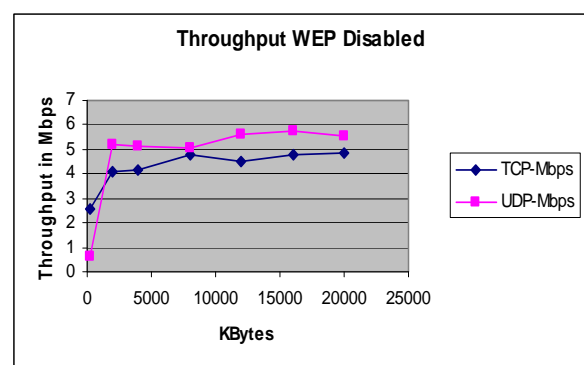


Figure 5. TCP vs. UDP Throughput – WEP Disabled (WS)

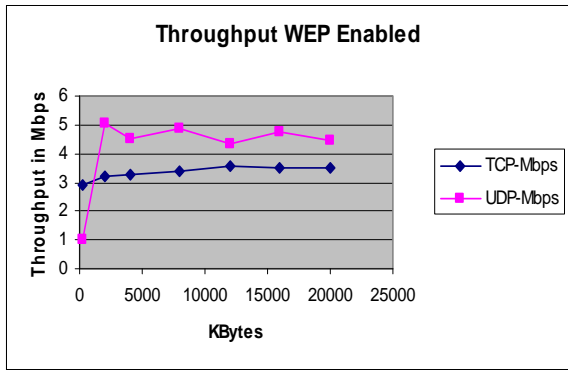


Figure 6. TCP vs. UDP Throughput – WEP Enabled (WS)

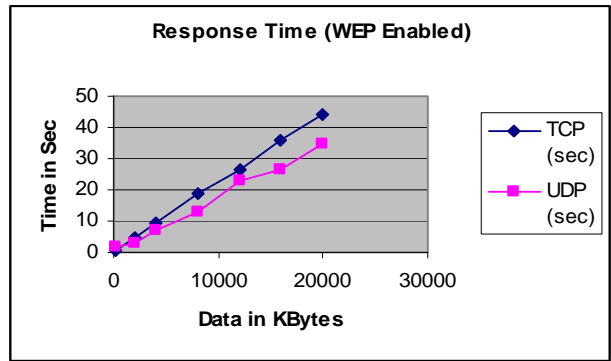


Figure 8. TCP vs. UDP Response Time (WS) WEP Enabled.

One important thing to add here is that the experiment while WEP is enabled was performed inside the same room where as while the WEP is disabled test was performed on a laptop in the first floor while the access point, the server, and the antenna are inside a room in the second floor. The reason for this is to take the best case while WEP is enabled (inside the same room) and a bad case while the WEP is disabled (in another floor); this situation will give results with no courtesy to the environment and we expect such results are practical to consider.

C. Response Time Results

Figures 7 and 8 show the response time of the system. As can be seen, the UDP has lower response time in both cases. This result holds for the QCHECK software when the WEP is enabled and reversed when the WEP is disabled. Figures 9 and 10 indicate that the difference in all cases between the UDP and TCP is small.

Response time is measured by selecting the average response time value is a set of 10 trials (with Min, Avg, and Max) to be practical; i.e. the average of 10 times is recorded.

D. Streaming Results

QoS and user satisfaction are of high importance to users of the multimedia, Voice over IP, and video broadcast applications. No need to acknowledge and data sent for faster delivery. In our system, the UDP protocol will be used for this purpose. Lost data in such applications is a measure of acceptance for the whole system; the less lost data, the better the system. QCHWCK software is used to measure this factor. Figure 11 and 12 shows the streaming results.

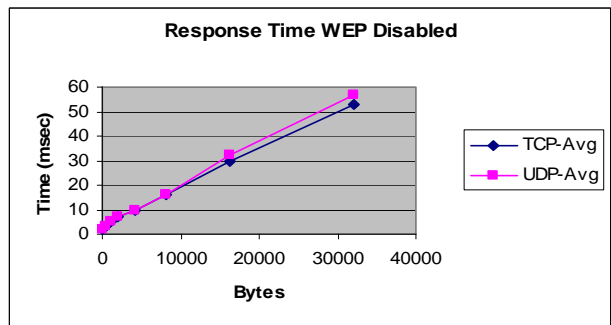


Figure 9. TCP vs. UDP Response Time (QC) WEP Disabled.

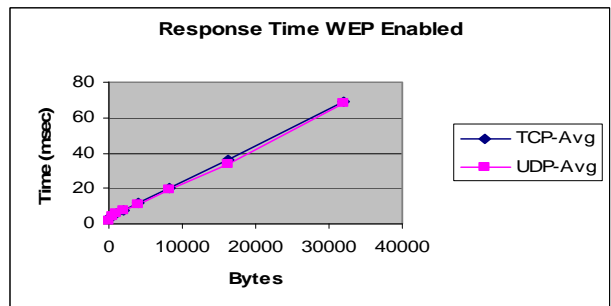


Figure 10. TCP vs. UDP Response Time (QC) WEP Enabled.

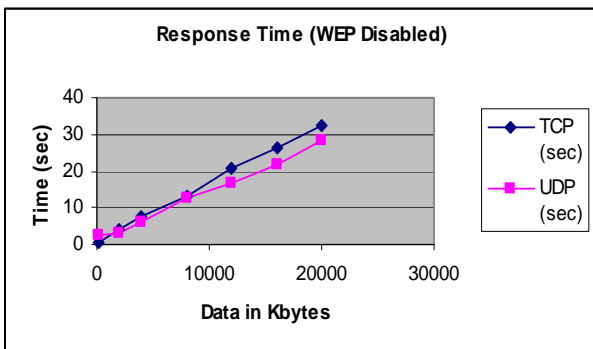


Figure 7. TCP vs. UDP Response Time (WS) WEP Disabled.

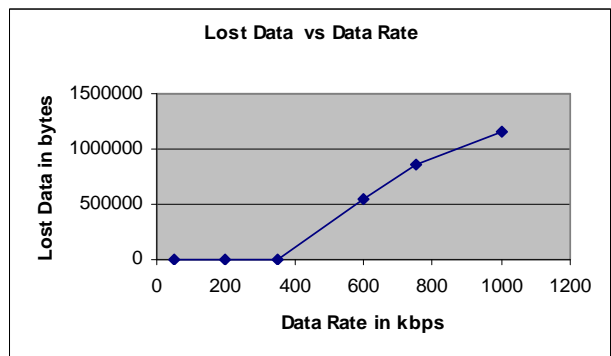


Figure 11. Lost Data vs. Data Rate.

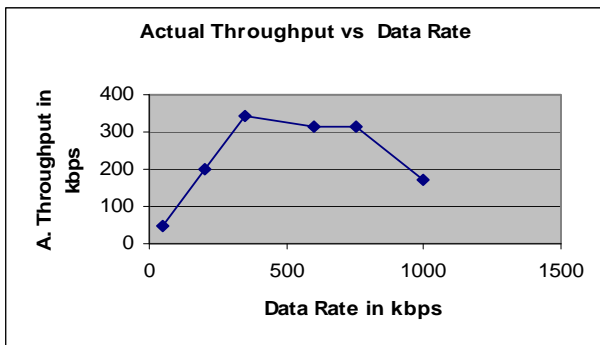


Figure 12. Throughput vs. Data Rate.

By looking at Figures 11 and 12, we notice that the data loss rate increases as the sent data rate increases. This result however, doesn't hold when the data rate sent is low where the chance for losing data becomes zero. Figure 12 indicates that the actual throughput increases to a point and starts decreasing again and a clear reduction is shown with large data sizes.

VII. CONCLUSION

Special consideration should be given to developing wireless applications including offloading some of the processing to network, code compaction due to memory limitations, dynamic memory allocation tracking, recursive functions consideration, screen size, power consumption, code partitioning for better performance, APIs reduction, and the ability to manage. MIS model addresses many performance factors needed to be considered when developing mobile applications and have the limitation of the environment it is used in. Generalizing the MIS to other environment and using other devices shouldn't be a headache if all the new environmental parameters were taken into considerations.

REFERENCES

- [1] Bryan Morgan. "Introduction to the Wireless World Part 1 & 2". <http://www.InformIT.com>, Feb 2, 2001.
- [2] Luciano Baresi, Andrea Maurino, Stefano Modafferi, "Workflow partitioning in mobile information systems", Proceedings of IFIP TC8 Working Conference on Mobile Information Systems (MOBIS 2004)

- [3] Diane Tang and Mary Baker. "Analysis of a Local-Area Wireless Network". Copyright ACM 2000.
- [4] Ayşe Göker, Hans Myrhaug, "Evaluation of a mobile information system in context", Information Processing and Management: an International Journal, v.44 n.1, 2008, p.39-65. [doi:10.1016/j.ipm.2007.03.011](https://doi.org/10.1016/j.ipm.2007.03.011)
- [5] Mark Beaulieu, "Concepts for Working with Wireless Application", <http://www.InformIT.com>, Feb 01, 2002.
- [6] Eric Giguère, iAnywhere Solutions, "Designing Mobile Applications for the Enterprise", a Sybase company, 2000, <http://www.perfectxml.com/Conf/Wrox/Files/eric1text.pdf>
- [7] Justin Helmig, "Programming Considerations for Developing Next-Generation Wireless Embedded Applications", Texas Instrument, SQP002 – January 2002 (White Paper)
- [8] HP Company, "Wireless and Mobile Computing", the Total-e-Mobile Business Solution, March 2001.
- [9] Fred Goldstein, "Wireless Technologies for Mobile Applications", TIAX LLC, Massachusetts, EntNet@SuperComm, 2001.
- [10] Shioh-yang Wu, Chun-Shun Chang, Shih-Hsun Ho, Hung-Shun Chao, "Rule-based intelligent adaptation in mobile information systems", Expert Systems with Applications: An International Journal, v.34 n.2, 2008, p.1078-1092. [doi:10.1016/j.eswa.2006.12.014](https://doi.org/10.1016/j.eswa.2006.12.014)
- [11] Ghani, Rod, (2001), "3G: 2B or not 2B?", www.starcore-dsp.com/docs/pdf/wsdpaper_oct00.pdf
- [12] Bourmakin, Eugene; Michael Samarin, Lora Samarina, Eero Vitie, Eugene Gryazin, Johannes Lehtinen, "Indoor Guide A Java and CORBA based mobile application", Group for Robotics and Virtual Reality, Department of Computer Science, Helsinki University of Technology, www.nrc.nokia.com/Vivian/Public/Misc/repBoumarkin0601.pdf
- [13] Krogstie, John, "Requirements Engineering for Mobile Information Systems", 2001, <http://www.ifi.uib.no/conf/refsq2001/papers/p6.pdf>
- [14] Andrew S. Tanenbaum, *Computer Networks*, Fourth Edition, Prentice Hall PTR, 2003, p 557-573
- [15] Joanne Holiday, Divyakant Agrawal, and Amr El Abbadi, "Disconnection Modes for Mobile Database", July 2002, *Wireless Networks*, V 8 issue 4, 2002.
- [16] WSTTCP software (Copyright © Sungiin Chun, 1996)
- [17] QCHECK Console Version 1.3 software (Copyright © NetIQ Corporation, 1995-2000).

AUTHORS

R. M. H. Al-Sayyed, S. Manaseer and O. Rababeh are with the University of Jordan, Amman, Jordan.

Manuscript submitted on June 22nd, 2010. Published as resubmitted by the authors on June 29th, 2010.