

Building a Microcontroller Virtual Lab Using Web-based and Mobile Agents Approaches

Saleh A. Al-Shehri and Mohammed Al-Oqeely

*Department of Computer Engineering, College of Computer & Information Sciences,
King Saud University P.O.Box 51178, Riyadh 11543, Saudi Arabia*

(Received 9 May 2000; accepted for publication 12 May 2001)

Abstract. Among the recent applications of Internet technology is virtual labs which enable users to conduct experiments from a distance. Most of the existing virtual lab applications fall into one of the following categories: lab simulators, telerobotic control, weather readings, database management, real-time video photo transferring and vending machines control. In all of these systems, the user either has a limited interaction with the remote system, or is forced to deal with simulators instead of real devices. In order to make the most benefit of remote experimenting, the user should have sufficient power over actual lab devices and equipment. Furthermore, the student should not worry about restrictions on access times or a failure of one of the lab equipment. In this paper we present the design and implementation of a computer engineering lab that applies both web and mobile agents technologies. The novel proposed mobile agents design offers flexibility and failure tolerance and overcomes some of the pitfalls of many existing virtual labs. Furthermore, the study concludes with a comparison and analysis of the proposed approaches.

1. Introduction

The explosive growth in Internet and web technologies is being driven by the tremendous advantages these technologies offer. Delivering textual information was the initial use of the Internet [1]. Well known services provided by the Internet like Telnet, FTP, Chat, E-mail and Newsgroup were all text based. Nevertheless, the Web revolution has made many of old Internet services, either integrated within HTTP documents or rendered obsolete [2]. Moreover, the use of Internet has gone beyond the expectations of many of us. Remote experimenting and distance learning using what is called *virtual labs* (VLs) is considered among the promising areas of Internet research [1-2].

In almost any university, electrical and computer engineering departments have some engineering laboratories such as robotics, logic, instrumentation and microprocessors lab. Because of the fast growth and advances in the electronics and computer technologies, these equipment and devices become obsolete fairly quickly which makes frequent upgrade of such devices a necessity [2]. Since upgrading must be made in a large portion of equipment and devices in the lab, it may not be always cost effective [2-3]. Another problem is that, usually, the number of students exceeds the number of equipment which makes it necessary to extend lab hours and could also force the instructor to deal with groups of students for each experiment instead of individual students. Furthermore, during the first few classes of each semester, students spend most of their time to familiarize themselves with the new equipment [2]. Simple experiment may take more than one or two weeks to be completed. Considerable time and manpower can be saved if students are granted unrestricted access to labs 'virtually' even before the semester begins.

It is true that nothing can replace the feeling of being in a real lab. However, building a virtual lab that provides appropriate remote experimenting and learning can help to overcome many of the problems mentioned earlier. First, few lab equipment need to be used to conduct experiments. Consequently, upgrading just a small number of equipment will reduce cost [4, 5, 6]. Second, the instructor will not be forced to do group work if he does not desire to. Every user can do his work individually any time. Finally and most importantly, the access can be made location independent; meaning that the technology gives the student the luxury of accessing virtual labs from anywhere in the globe 24 hours, 7 days a week. Such a concept gives the students more opportunities to conduct experiments on some of the cutting-edge labs and equipment [1, 5, 6].

2. Previous Work and Motivation

Remote experimenting using virtual lab systems is a fairly recent technology and a new paradigm [1, 6]. Indeed, virtual lab field is considered one of the active fields for both research and commercial development [2].

2.1 Categories of current systems.

Early attempts to conduct remote experiments on the Internet were reported in [4, 7] in 1994. Most of the work reported since then can be categorized using two features. These are a) accessibility and graphical user interface and b) the type of applications. Fig. 1 shows a simple taxonomy of VLS according to these categorization schemes.

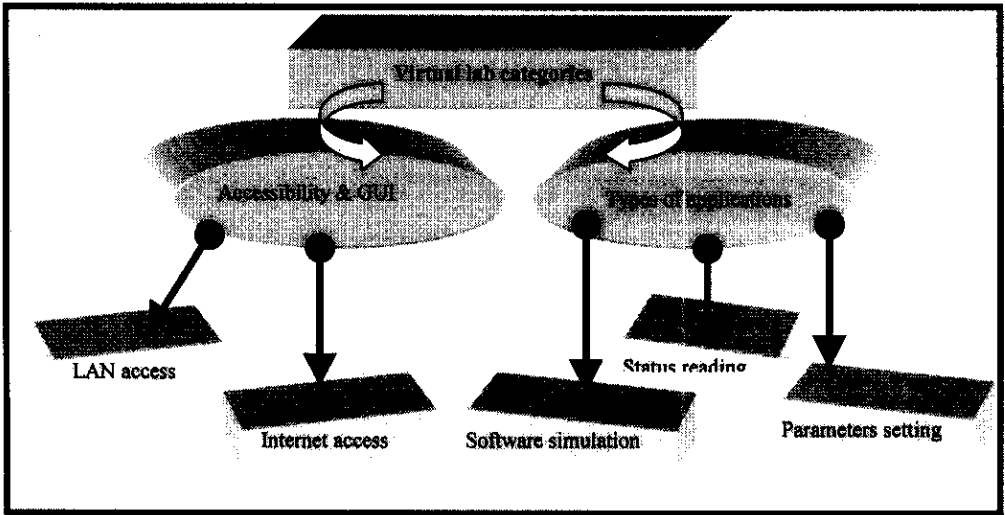


Fig. 1. Simplified taxonomy of virtual labs.

2.1.1 Accessibility and GUI

Previous studies and practical systems implementation show that there are two kinds of virtual lab systems from the standpoint of access to these systems. Virtual labs that can be accessed using local networks using dedicated software and others that can be accessed through the Internet using web browsers as their GUIs [5, 8].

References[1, 3, 4, 7, 9-15] present studies and practical implementations of virtual lab systems where the user can access the virtual labs and conduct the required experiment. Web browsers are the GUI's that are used in these systems.

2.1.2 Types of experimentation

Previous studies and implemented virtual lab systems show that the types of experimentation are mainly limited to setting systems parameters, process and/or software simulations and status reading. In [2], a survey of existing virtual lab systems was done. Of eighty three institutes that provide distance education only 5% have built virtual labs. Moreover, many of these virtual labs operate using software simulators.

Robotic arms movement and video camera photo and sound transferring are kings of practical implementation in setting systems parameters group [4, 8, 9, 14, 16]. Examples of simulation group implementations can be recognized in the form of logic circuit and nuclear explosion simulations. Weather reading, vending machines and

networks status are forms of status reading group. References [3, 4, 13-14, 16-17] present robotic arm control, vending machines control and electrical power system monitoring systems.

The mobile agents paradigm is a promising technology. It provides a very good environment for building limitless application for different purposes [18]. There are several areas where mobile agent is visible. Among them are search engines, electronic commerce, distributed computing, finance, travel, personal assistance, entertainment, email filters and management of network devices [18-21]. A user can set up certain conditions that the agent will use as a guideline to perform certain tasks autonomously and give the results to the user as a final product [20, 22]. These mobile agents travel through networks and run autonomously on the destination computers which hold programs that offer services to the mobile agent [23]. There are three issues specific to mobile agents that make them a particularly attractive technology namely: mobility, adaptability and collaboration [22, 24]. Using these three features, agents of increasing sophistication and autonomy can be built. The mobile agent technology is pretty much still in the research side and in the moving stage towards mainstream computing. This gives us the motivation to build a mobile agent and try to study it as an alternative for implementing virtual labs. Until the time of writing this paper no published work has investigated the use of mobile agents in virtual labs to the best of our knowledge.

3. The Objective

From the literature review we can see that there is a need for a system that can give the user full access capabilities to remote labs. In fact, reference [2] suggests, for the scope of future study, building a micro-controller virtual lab that will allow the student to write the processor code and pass it to the virtual lab. We have built such a system using the mentioned approaches. Full capability in our study means giving the user the ability of full lab equipment programming such that reasonable experiments can be performed remotely online.

In conclusion, the motivation for conducting this study stems from the fact that a virtual lab system that gives full remote programming capability and at the same time can tolerate failure and offer flexibility, is needed. To build such a system it is important to incorporate a variety of new features including intelligent and real-time human machine interactive interface based on a state of the art technology and recent advances in interactive distributed system and Internet technology such as mobile agents and dynamic HTML. The proposed system will fill in an important gap in the perfect-utilization goal of the engineering department's labs.

4. System Configuration

Figure 2 shows a block diagram our web-based system while Fig. 3 shows our second system based on mobile agents. A microprocessor lab is used as our implementation example in this study. Also, for the purpose of demonstration, we have installed a robotic arm in the virtual lab which can be accessed and controlled by the user. An off-the-shelf micro-controller kit is used. A top view of proposed virtual lab at the CEN department in King Saud University is shown in Fig. 4.

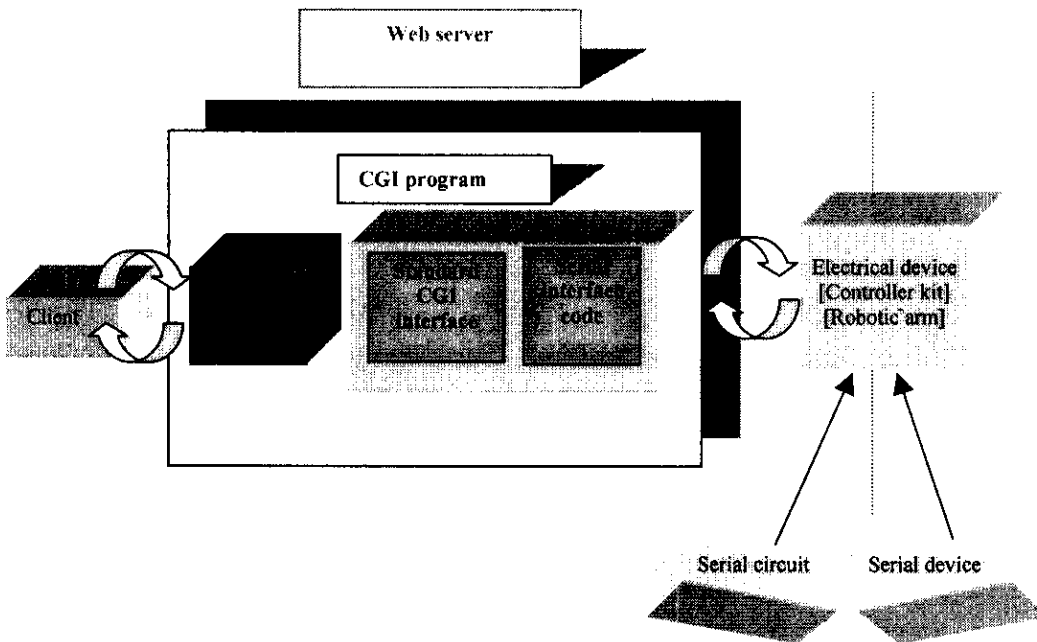


Fig. 2 Web based VL setting.

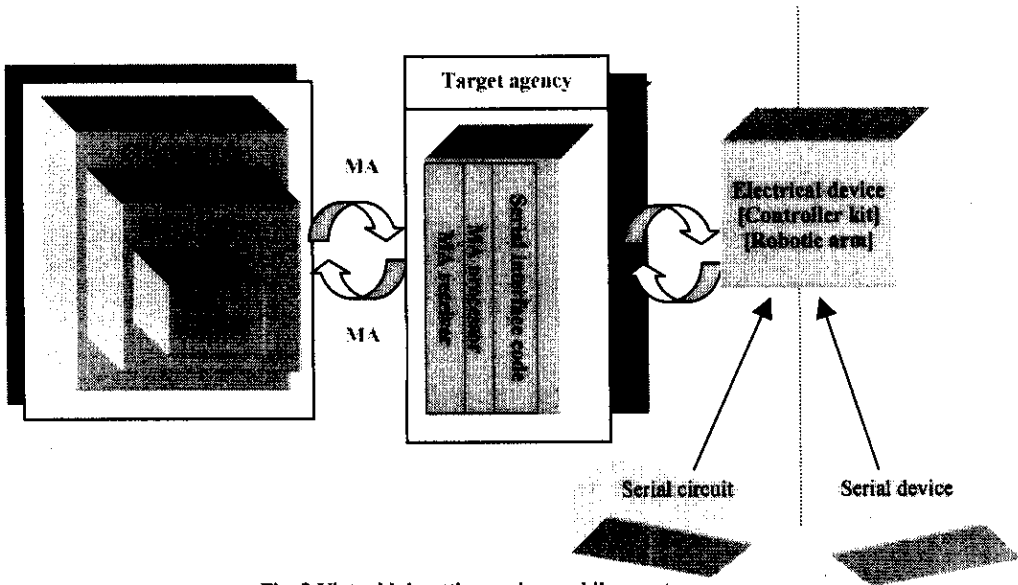


Fig. 3 Virtual lab settings using mobile agents.

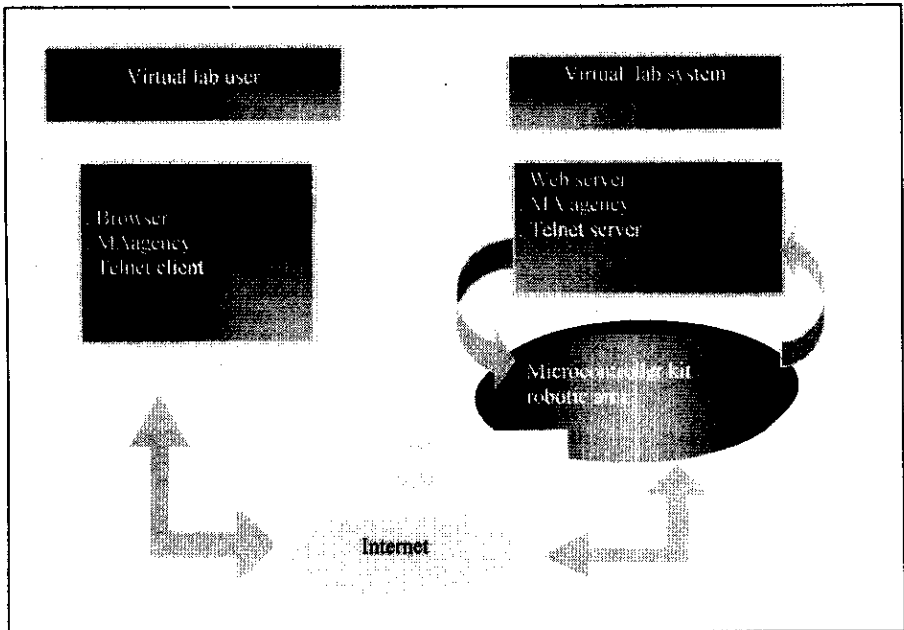


Fig. 4 CCIS virtual lab settings.

In our work, we have used an MS Windows 98 machine as our lab host machine. This machine supports all of the required services (HTTP, Telnet, Mobile agent system, etc.). The user will be able to write the code of this kit and pass it to the remote kit using any of our methods through the Internet then run the code and receive the result.

5. The Virtual Lab System Construction

5.1 Pure HTTP approach.

5.1.1 Pros and cons.

The advantages of using web technology in such system can be observed from the fact that the location of the user machine is transparent. This is also true for the server machine [5, 6, 8, 11, 25]. The system provides unlimited access time to the user. It shall be available for 24 hours [5, 6, 11, 25]. Variant documents formats are supported [5, 6, 26]. WWW supports text, graphics, audio, video, etc. The good graphical user interface (GUI) is an important advantage of WWW [4, 6, 25-26]. Since browsers are used in WWW, the system is platform independent.

Nevertheless, the use of HTTP for virtual labs has some disadvantages. The general weaknesses can be summarized as follows:

- 1- Not suitable for long server processing time.
- 2- Security weakness [6, 27].
- 3- Limited group collaboration [28].
- 4- Does not support perfect real-time application because of its stateless nature. The HTTP protocol was designed with a different goal in mind. It was designed originally to retrieve static documents in specific format [1, 11, 29, 30].

5.1.2 Construction of web-based VL system

The proposed web-based implementation offers user authentication using a user ID and password if needed. The administrator is responsible for adding all users with their passwords. Fig. 5 shows general diagram of a web site design whereas Fig. 6 shows the simple access control block diagram.

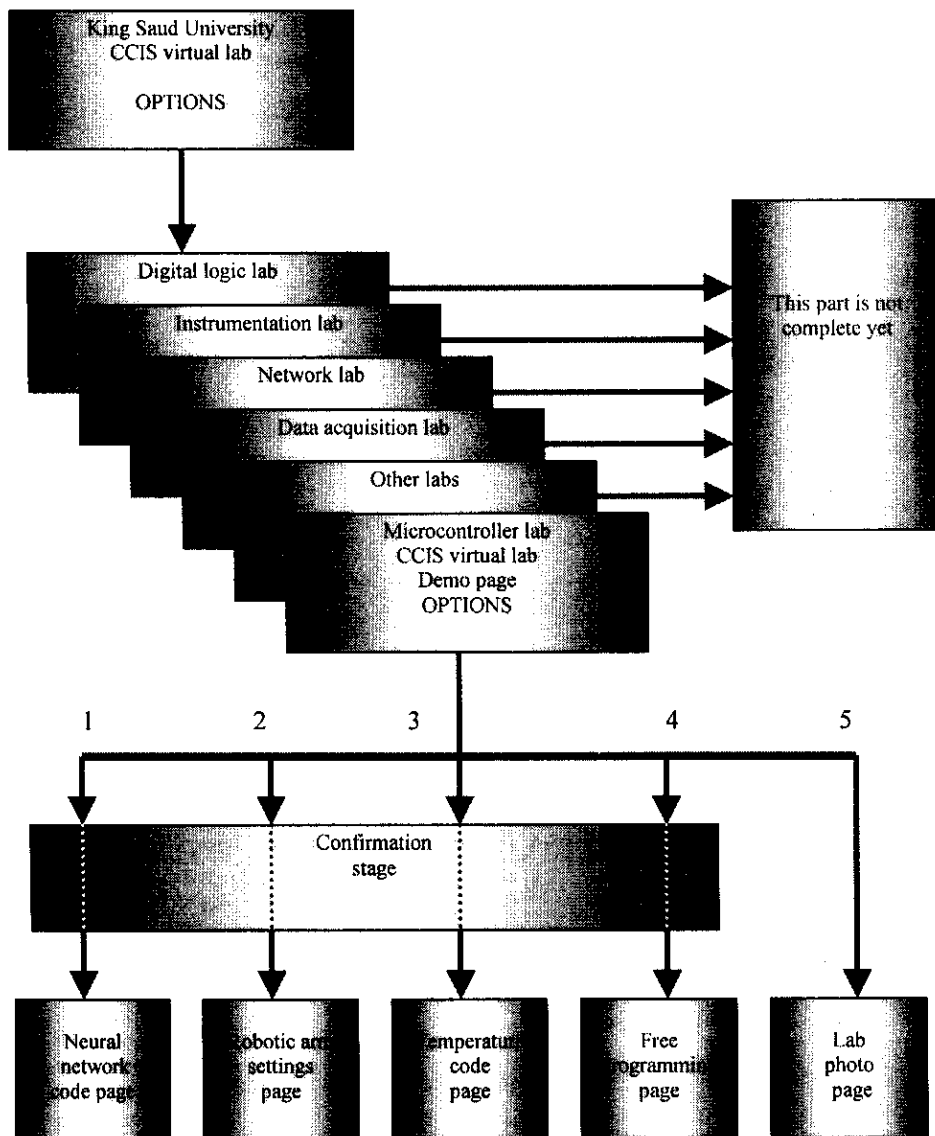


Fig. 5 General diagram of the CCIS virtual lab web site.

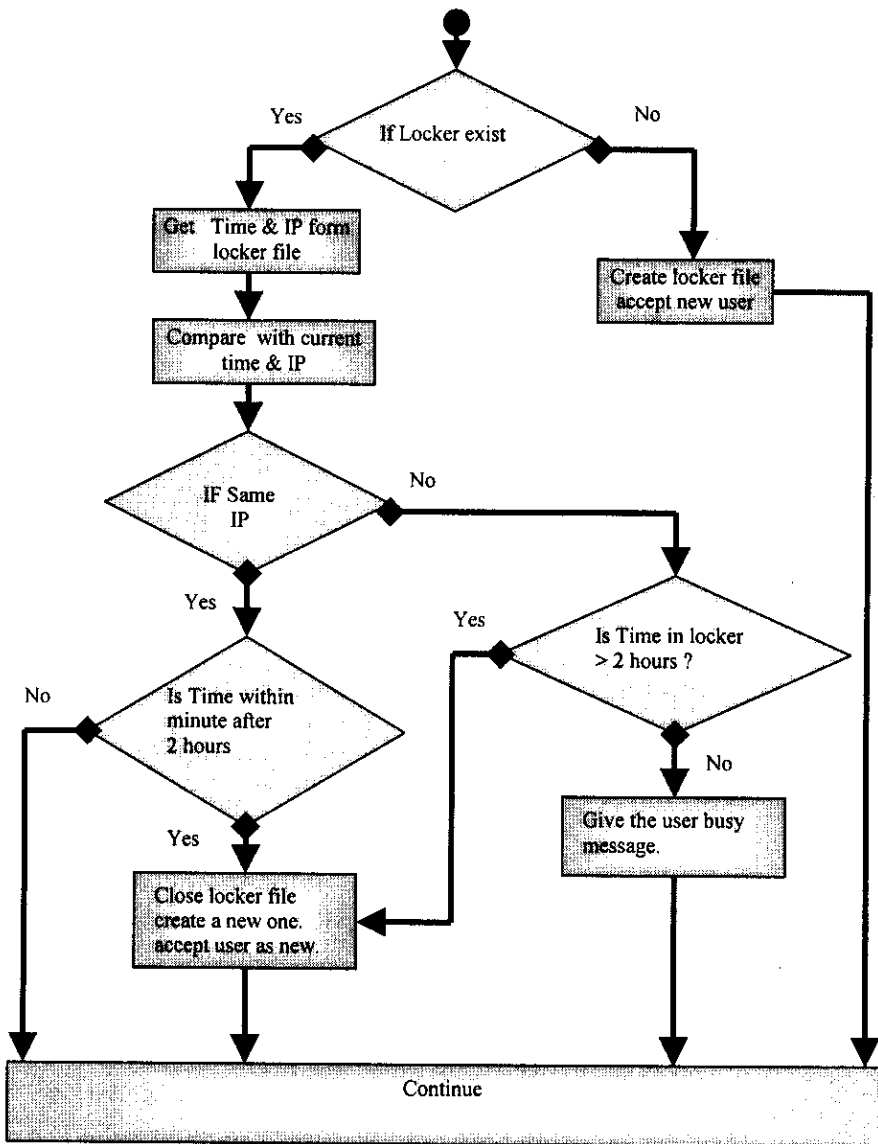


Fig. 6 Access control block diagram.

Common Gateway Interface (CGI) programming scheme was used to build this system. CGI is a standard for interfacing external application with web servers. The CGI programs are executed in real-time to generate dynamic information as illustrated in Fig. 7.

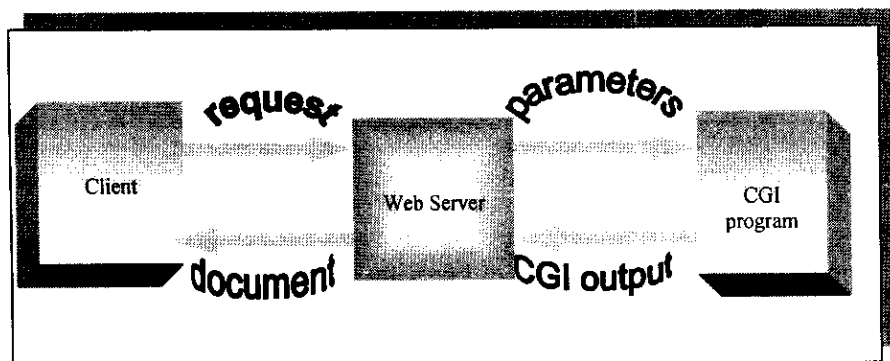


Fig. 7 Flow of data using CGI protocol.

Details of the system have been omitted in this paper because of space limitations, but can be obtained from [31]

5.2 Mobile agent approach.

5.2.1 advantage of mobile agents over the web.

There are some limitations in using the web approach for constructing a virtual lab. In addition to the general comments given earlier in section 5.1, some of the following points are specific to the use of HTTP in our own implemented virtual lab:

1. Using GET or POST methods to submit the user kit's code is not convenient.
2. If the kit code executes for a long time, the connection between the user's browser and the server will be disconnected after pre-set time. Usually, the web server is set to accept a connection interval within certain value. This is to ensure that the server will not be overloaded 'heavy users.'
3. If the server is down for any reason, the user will have to wait and continue to check form time to time. This may force him to waste a lot of time.
4. The access to the lab must be done while the user is online. It would be much more convenient to let the user specify the time(s) that the experiment will take place even if he is not present.
5. If the server fails for any reason, all of the steps in the session should be done all over again.

6. If more than one virtual lab has the same set of equipment and setup then switching from one lab to another must be done manually by the user.
7. The HTTP is a stateless protocol where the server and the client do not remember (without extra overhead) previous interactions.

These limitations and drawbacks motivate the search for alternatives. In our study, the mobile agent paradigm has been chosen as an alternative. Fig. 8 shows the mobile agent solution.

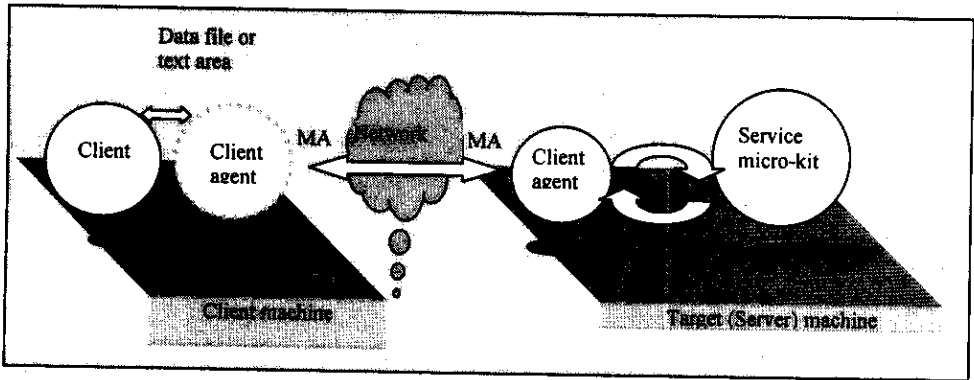


Fig. 8 Mobile agents paradigm.

5.2.2 Pros and cons.

Mobile agents technology offers some nice features. First, few network resources are used by the mobile agent since it carries most of its required resource with it [23, 24, 32, 33]. Second, the data is transferred with the mobile agent code to the destination which reduce the network traffic [23, 24, 32, 33]. Third, the mobile agent interacts with the surrounding autonomously and asynchronously which saves a lot of user time [23, 32, 34]. Moreover, mobile agents can react to environment changes dynamically which makes the mobile agents system a robust and fault tolerant one [18, 32]. Finally, to make the mobile agents run in heterogeneous environment it must be implemented as a platform-independent system [20, 32].

On the other hand, since a mobile agent system is an open system it makes the host subject to security problems [35]. Further, dedicated mobile agent environments in the form of mobile agents places have to exist in every host that the mobile agent will visit [20, 24, 36]. These are the two main disadvantages of mobile agents system.

5.2.3 Mobile agent construction

There are several mobile agent platforms. We have chosen the Jumping Beans [18] platform as our base to build the virtual lab mobile agent for several reasons:

- 1- It adopts the client/server architecture which provides excellent management and security.
- 2- Its security is forced by using the network administrator monitoring. Also, the development language is Java which is interpreter language. This improves system's security.
- 3- One of the key features of Jumping Beans is the central management capability [18].
- 4- Jumping Beans can be integrated with existing systems. Developers need not redesign their application.
- 5- Store-and-Forward is supported by Jumping Beans. The mobile agents will be stored until the target machine is available. Duplication or lost mobile agents are guaranteed to not happen with Jumping Beans. If unexpected network failure occurs or the host machine crashes, the mobile application will not be lost [18].
- 6- Persistence concept is implemented by Jumping Beans. If power loss or other factor happened that interrupted the execution, the state of the system is not lost. It can be re recovered using persistence file support.
- 7- Embedded application in networked devices and similar application are achievable using Jumping Beans since it is 130KB in size [18].

Jumping Beans is a Java framework, or API, which can be embedded in an application to add mobility to it. By embedding the Jumping Beans, the application can jump from computer to computer while they are executing, even if the application has never been installed on the target machine. Each host computer which wishes to participate in the mobility must have the Jumping Beans "daemon" software running on it. This daemon listens to the TCP-IP port to receive mobile application. In addition it performs many other tasks, including [18]:

- Receive incoming mobile applications.
- Enforce security while mobile applications are executing.
- Dispatch, deactivate, and reactivate mobile application.
- Provide Jumping Beans services to mobile applications and to the resident host software.
- Crash recovery when needed.

The basic architecture of Jumping Beans is shown in Fig. 9.

Since the mobile agent is an application that can travel from one machine to another and is executed there the problem of long server processing time, which is a major disadvantage in web approaches, does not exist here. Also, if crashes happen at any time the Jumping Beans system helps in recovering the mobile application from its

last state using a persistence system. If more than one identical virtual labs are provided then the mobile application can be easily designed to interact with any one of them either transparent or not to the user. Finally, the user does not have to be online during the lab session to perform his experiment. Instead, he can enter his microcontroller kit code in the nice GUI of the mobile application and specify the time he wants his mobile application to travel to the destination. Also, he can specify the time he wants to get the result. If he is not online, the mobile application will stay in the system until the user gets back online.

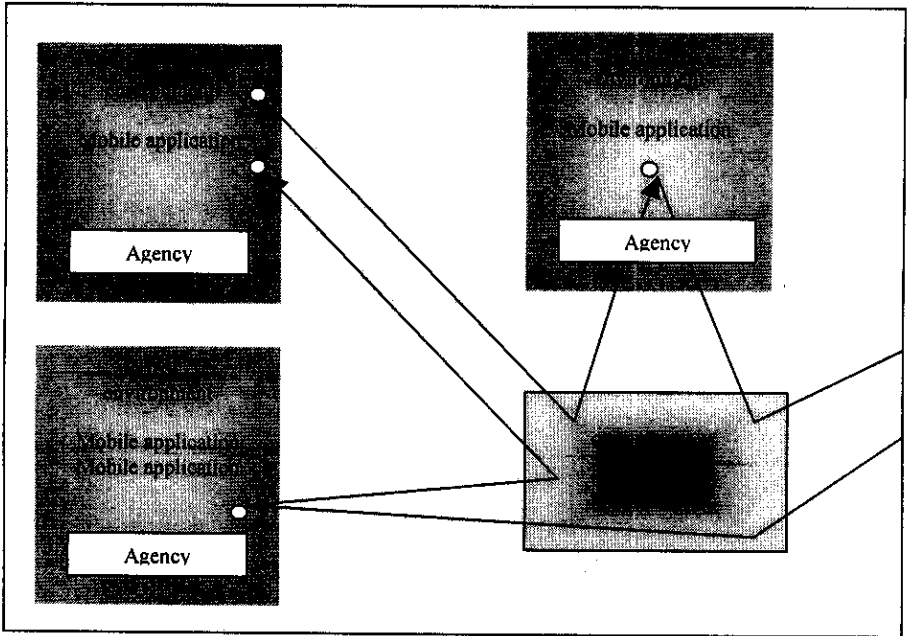


Fig. 9 Basic architecture of the jumping beans system.

Our MA has two versions. The first version gives the user a text area in the GUI at the start of the MA to enter the kit's code in it. In the second version, the user can use a file, which holds code. There are several fields that the MA GUI has namely: the input file, the target machine name, the serial port settings, the time out settings and the speed of transmitting the code to the kit. Figure 10 shows the block diagram of our mobile application while figures 11 and 12 show the its GUI.

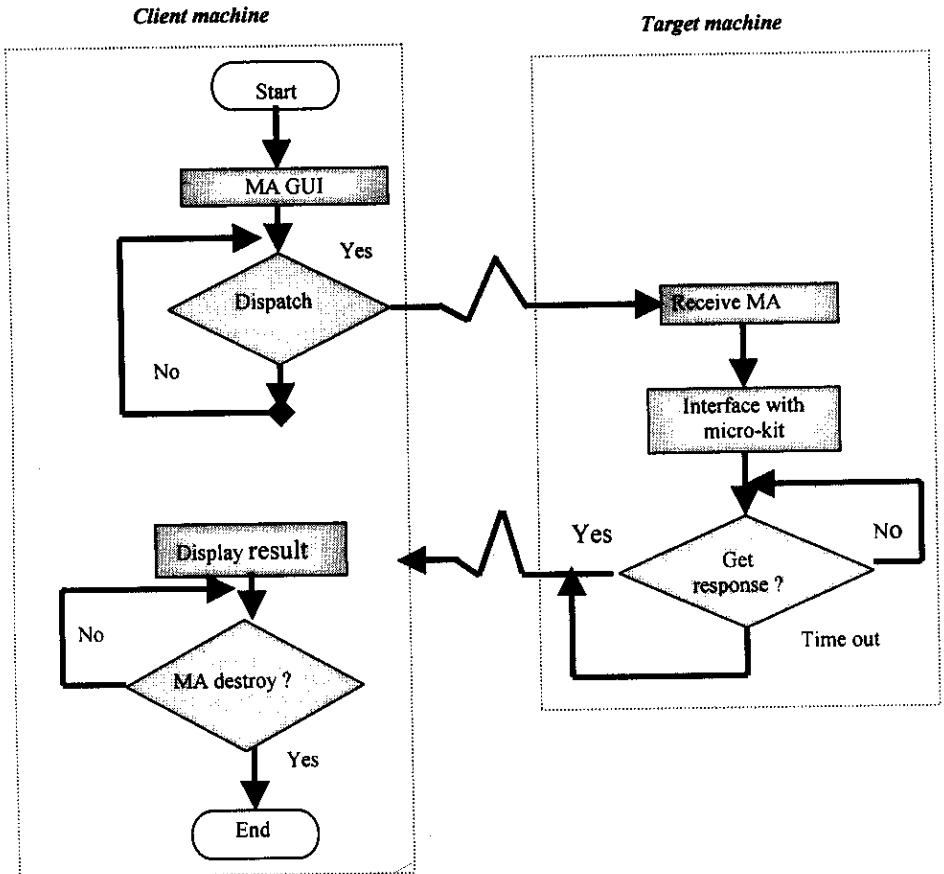


Fig. 10 Block diagram of the mobile application flow.

CCIS Virtual Lab GUI : واجهة التطبيق للمستخدم

Parameters of serial port on target machine:

Port Name: **COM1**

Baud Rate: **9600** Parity: **None**

Data Bits: **8** Stop Bits: **1**

Flow Control In: **None** Flow Control Out: **None**

What is the Target machine to interact with? **RECEIVING**

Where should the mobile application go after getting the result? **SENDING**

Enter the Microcontroller Kit Code here :

```
ram
new
10 print "Hello "
20 print "CCIS members"
run
```

Enter the delay between lines : **1000**

Enter the serial port timeout : **10000**

If the serial port times out, what machine to go? **SENDING**

Dispatch

Fig.11 Mobile agent GUI for entering microcontroller kit code.

In the server machine, the administrator can track the mobile agents and get status information about them by pressing the GetInfo button in the Jumping Beans administration console. It has a variety of useful administration functions [18].

A good feature of using MA can be seen from the fact that you can connect several kits to several target machines then dispatch the MA to the machines using the itinerary list. If one is busy or is not on-line then the MA will travel to another machine. There could be other departments or other universities in the region having the same equipment. This MA can travel to those departments. This can be transparent to the user. With a very simple code modification as itinerary list can be built to handle the above scenario.

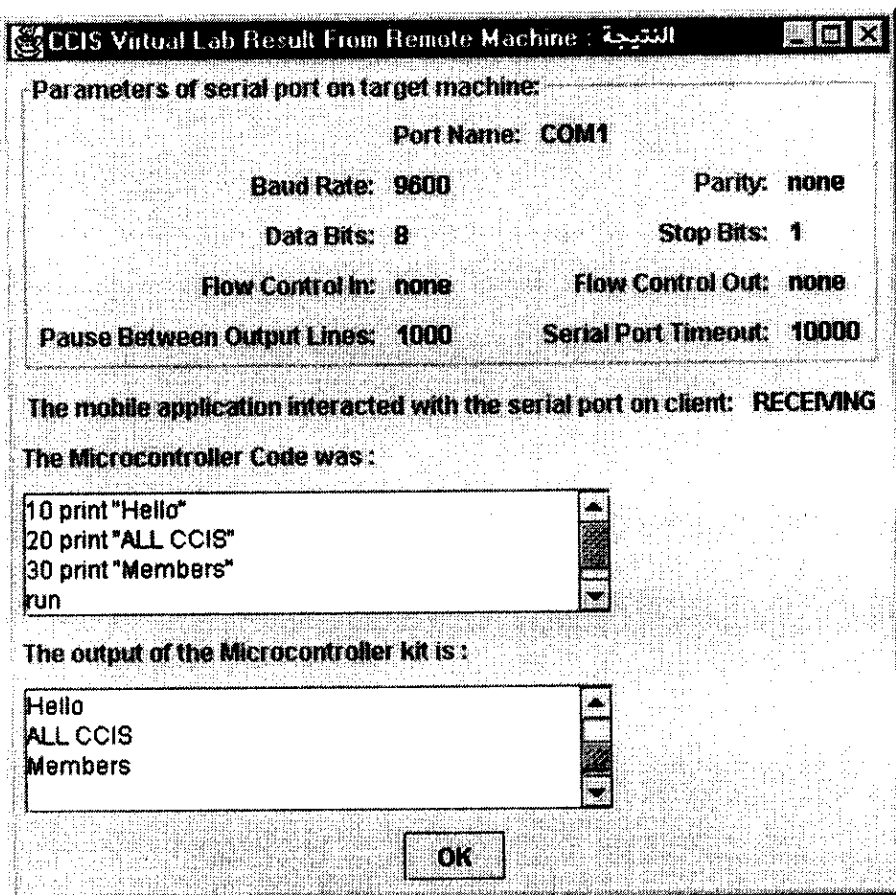


Fig. 12 Result of microcontroller kit experiment.

5.3 Telnet

Telnet is an Internet service that connects a client machine with remote machines. [37]. By using Telnet, or similar remote shell access services, one can establish a connection through the Internet and remotely login into any computer that supports Telnet communication. The main advantages of Telnet are its ease of use, error-free connection, convenience for poor line quality and remarkable stability [37- 38]. The key disadvantage of Telnet is the fact the Telnet does not support graphics. A Telnet client and server have been installed in both the client and server machine. Any available

communication software located in our server can be used to communicate directly with the serial port of the server machine.

6. Results Summary and Alternatives Comparison

A micro-controller virtual lab was built using two approaches. In the web-based design, the user enters his microcontroller kit's code into this GUI through his browser and then he hits a submit button. The result comes back to the user as an HTML file. In the other approach, using Java Foundation Classes (JFC) it was possible to build a very comprehensive GUI. The user input should be entered into these GUIs in the client machine and then the user gives the dispatch command. The mobile agent travels to the target machine and interacts with the virtual lab equipment then brings the results back to the user machine. Table 6.1 summaries the results, observations and caparison between the three approaches used in KSU computer engineering virtual lab.

Table 6.1 Summary and comparison

Approaches items	HTTP approach	Mobile agent approach	Telnet
Tool	1. CGIC library 2. PCLAC library 3. Ximax web server 4. IE browser 5. Borland C compiler 6. Quick cam	1. Java 1.1.8 2. Jumping beans 3. Quick cam	1. Telnet client 2. Telnet server 3. Hyper terminal
User Interface	Very friendly	Friendly	Text only
Effort in Development	Moderate	High	Very low
Platform-dependency	Low	Low	Moderate
Online necessity	Yes	No	Yes
System failure tolerance	No	Yes	No
Applicability to long time processing	No	Yes	Yes

The study has concluded that the two main advantages of mobile application over the HTTP approach are its ability to tolerate with the long processing time in the server side in case that the microcontroller kit's code needs such long processing time. The other advantage is the ability of the mobile application to stay on the system for arbitrary time in case of server not available or the user specifies future time for the mobile application travel to the virtual lab. Other less important advantages can be observed from table 6.1.

Future extensions

From all of previous discussion, we have come up with a good framework for building virtual labs using different approaches. This framework will enable us to upgrade our existing system to a more advanced one in the near future. A tentative scenario for our upgrade could be summarized as follows:

- A user will launch a MA and enter his kit's code then dispatch it.
- When he gets the result, he could re-do the experiment again if the result is not satisfactory or otherwise hit a special dispatch button which will make the MA travel to the lab grader.
- The lab grader will see the lab's result, enter the mark in the dedicated field and then dispatch the MA.
- The MA will travel to the course instructor. He will see the lab mark, add it to the course mark and come up with the final grade for that student. Then he will dispatch the MA to the university's registrar.
- The MA will travel to the registrar system and place the student's grade into the appropriate system

Furthermore, it is evident that adding other dynamic web tools like Active X and Java applets is always advantageous. For instance, using the Java programming language, it is possible to build a customized connection-oriented applet for dealing with a specific lab instrument. This direction is also a good future extension.

7. Conclusions

A prototype of a virtual lab for digital systems and micro-controller experimenting is presented. The proposed design exploit web and mobile agents technology. In the first of two proposed paradigms which implement a full control micro-controller lab, a standard HTTP dynamic CGI approach is adopted. In the second, the same full functionality is achieved using Jumping beans mobile agents. The mobile agents approach which was used for the first time in such an application, opens up a new domain to explore. For example, having the whole experiment package travel to various lab stations (possibly all over the Internet) could revolutionize the way we view remote experimenting. Moreover comparative analysis between the proposed approaches has been conducted, as well, to show the merits and pitfalls of each option.

References

- [1] Munoz, C., Rayhan, S., Lels, T. and Patel, N. "UCF Virtual Lab-Software Requirement Specification ." http://feeds.engr.ucf.edu/VILAB/srs_v2.html
- [2] Anandapuram, S. "Remote Laboratories for Distance Education", *Master Thesis*, The College of Engineering, Florida Atlantic University, Florida, 1999.
- [3] Itschner, R., Pommerell, C. and Rutishauser, M. " Glass: Remote Monitoring of Embedded Systems in Power Engineering ", *IEEE Internet computing*, Vol. 2, (1998), 46-52.
- [4] Taylor, K. and Dalton, B. " Issues in Internet Telerobotics ", FSR'97 International Conference on Field and Service Robotics, 1997.
- [5] Electrical and Computer Engineering-Carnegie Mellon University. " The Virtual Lab: Engineering the Future." <http://www.ece.cmu.edu/~stancil/virtual-lab/virtual-lab.html>, fall 1995.
- [6] Greenwood, D. " Web-based Learning: Building an Online Course." *Master Thesis*, Information Networking Institute, Pennsylvania, 1998.
- [7] Rindrer, M. "Remote Experimenting." Interactive Computer Aided Learning (ICL 98), Austria, 1998.
- [8] Aktan, B., Bohus, C., Crowl, L., and Shor, M. " Distance Learning Applied to Control Engineering Laboratories." *IEEE Transaction on education*, Vol. 39, 1996.
- [9] Prihavec, B. and Solina, F. "User Interface for Video Observation Over the Internet." *Journal of Network and Computer Application*, 21 (1998), 219-237.
- [10] Karweit, M. "A Virtual Engineering/Science Laboratory Course." <http://www.jhu.edu/~virtlab/virtlab.html>
- [11] Munoz, C., Rayhan, S., Iles, T. and Zalewski, J. " Creating a Virtual Lab for Real-Time Education on the Web." *Proceeding of the 2nd IEEE Real Time Systems Education Symposium*, Montreal, Canada, (1997), 80-85.
- [12] Bergsma, G. " Building Web-Enabled Mobile Devices." *Circuit Cellar-The Computer Application Journal*, (1998), 41-47.
- [13] Concur System Technology " Concur System Technology Introduces Their Cyber Acquisition System", <http://www.concursys.com/press%20releases/prCAS.htm>, July 1998.
- [14] Goldberg, K., Santarromana, J., Bekey, G., Gentner, S., Morris, R., Sutter, C. and Wiegley J. " About the Tele-Garden", <http://telegarden.aec.html/intro.html>.
- [15] The University of Edinburgh, "The ECOSSE Control Hyper Course, Virtual Control Laboratory Manual." <http://www.chemeng.ed.ac.uk/ecosse/control/course/map/intro.html>
- [16] Steinberg, J. "Java Goes to the Extreme." <http://www.javaworld.com/javaworld/topicalindex/jw-ti-security.html>, 1998
- [17] Gergsma, G. " Building Web-Enabled Mobile Devices." *Circuit Cellar-The Computer Application Journal*, 1998.
- [18] Ad Astra Engineering "Jumping Beans White Paper. " <http://www.JumpingBeans.com>, 1999.
- [19] Cockayne, W. and Zyda, M. " Mobile Agents" Prentice Hall , 1997.
- [20] Versteeg, S. "Languages for Mobile Agents." *Master Thesis*, Department Of Computer Science and Software Engineering, The University of Melbourne, Australia, 1997, <http://www.cs.mu.oz.au/~scv/thesis.html>.
- [21] Do, O., March, E., Rich, J. and Wolff, T. " Intelligent Agents & The Internet: Effects On Electronic Commerce and Marketing", <http://nolb.coba.unr.edu/odie/paper.html>.
- [22] Wang, H., and Wang, C. "Intelligent Agents in the Nuclear Industry ." *IEEE Computer*, (Nov. 1997), 28-34.
- [23] General Magic " Mobile Agents White Paper." <http://www.genmag.com>, 1997-1998.
- [24] Joshi, N. and Ramesh, V. "On Mobile Agent Architectures." *Fourth International World Wide Web Conference*, Massachusetts, 1995.
- [25] Agranat, I. " Engineering Web Technologies for Embedded Applications." *IEEE Internet Ccomputing*, Vol. 2, (1998), 40-45.
- [26] Restino, A. " Health Communication Resources on the World Wide Web." Master's of Art in Helath Communication, Emerson Clollage, Massachusetts, 1996.
- [27] Tanenbaum, A. *Computer Networks*. 3rd ed., New York: Prentice Hall, 1996.

- [28] Dwyer, D., Barbieri, K. and Doerr, H. " Creating a Virtual Classroom for Interactive Education on the Web." *The 3rd International WWW Conference*, Germany, 1995.
- [29] Ibrahim, B and Franklin, S. " Advanced Education Uses of the World Wide Web." *Computer Networks and ISDN Systems*, Volume 27, 1995.
- [30] Parnes, P., Mattsson, M., Synnes, K. and Schefstrm, D." The Web Presentation Framework." *The Sixth International World Wide Web Conference*, California, 1997.
- [31] Alshehry, S. "Design and Implementation of Virtual Labs," *MS Thesis in Progress*, Computer Engineering Department, King Saud University, 2000.
- [32] Lange, D. and Oshima, M. " Programming and Deploying Java Mobile Agents With Aglets." Addison-Wesley, 1998.
- [33] Falchuk, B. and Karmouch, A." Visual Modeling for Agent-Based Application." *IEEE Computer*, Volume 31, (1998), 31-38.
- [34] Franklin, S. and Graesser, A. " Is It an Agent or Just a Program?: A Taxonomy for Autonomous Agents." *Proceedings of the Third International Workshop on Agent Theories, Architecture, and Languages*, Springer-Verlang, 1996.
- [35] Brown, L. " Mobile Code Security ", AUUG 96 and Asia Pacific World Wide Web, *2nd Joint Conference Proceedings*, Australia, 1996.
- [36] Lingnau, A., Drobnik, O and Doemel, P. " An HTTP-based Infrastructure for Mobile Agents." *Fourth International World Wide Web Conference*, Massachusetts, 1995.
- [37] An academic computing document "Remote Login with Telnet." <http://www.ecnet.net/users/miac/doc/unix/ecom/remte/telet.html>, Nov. 1995.
- [38] Khare, R. " Telnet : The Mother of All (Application) Protocols." *IEEE Internet Computing*, 2 (1998), 88-9.

بناء معمل افتراضي للتحكم الرقمي عن طريق التقنية النسيجية وتقنية العميل المتنقل

صالح علي الشهري و محمد العقيلي

قسم هندسة الحاسب، كلية علوم الحاسب والعلوم

جامعة الملك سعود، ص ب: ٥١١٧٨ ، الرياض ١١٥٤٣

(قدّم للنشر في ٠٩/٠٥/٢٠٠٠م؛ وقبل للنشر في ١٢/٠٥/٢٠٠١م)

ملخص البحث. المعامل الافتراضية تعتبر من التطبيقات الحديثة لتقنية الإنترنت وهي تتيح للمستخدم القدرة على إجراء التجارب عن بعد عبر الشبكة. وهناك عدد كبير من الأبحاث والأنظمة المطورة في هذا المجال وهي في الغالب تدرج في المجالات التالية: محاكيات المعامل، التحكم في الأنظمة الروبوتية عن بعد، قراءة وإرسال القراءات أو نقلها عبر الوسائط المتعددة .

في هذا البحث نعرض تصميم وبناء النموذج لمعامل هندسة الحاسب الآلي (أنظمة التحكم الرقمية) بإعتماد التقنية النسيجية وتقنية العملاء المتنقلين .

التصميم الأخير المقترح يتميز بتفادي بعض العوائق التي واجهت الأنظمة السابقة ويوفر درجة من المرونة وتحمل الأعطال. وفي نهاية هذا البحث تم عرض مقارنة تحليلية بين النظامين المقترحين.