Testing Mobile Agent Platforms Over the Air

Oscar Urra [#1], Sergio Ilarri [*2], Eduardo Mena [*3]

[#] *Aragón Institute of Technology*
María de Luna 7, 50018, Zaragoza, Spain
[1] ourra@ita.es

[*] *IIS Department, University of Zaragoza*
María de Luna 1, 50018, Zaragoza, Spain
[2] silarri@unizar.es
[3] emena@unizar.es

***Abstract*— Mobile agents are considered a suitable technology to develop applications for wireless environments with limited communication capabilities. Thus, they offer interesting advantages compared with a traditional client/server approach, derived from their autonomy and capability to move to remote computers. An important benefit is that mobile agents can carry the computation wherever it is needed at that moment. For example, instead of communicating a large amount of data from a computer to a mobile device, a mobile agent can move to that computer to process the data there, saving wireless communications. Similarly, a mobile device can assign a task to a mobile agent which travels to a fixed computer with the appropriate resources, relieving the overload of the mobile device.**

However, most experimental evaluations of the mobile agent technology focus on distributed environments with fixed computers connected through a wired network. Therefore, it is not clear how well prepared the current implementations of mobile agent platforms are to deal with unreliable, wireless communications, and with mobile devices with limited capabilities. In this paper, we evaluate experimentally the behavior of some platforms in such contexts, with the goal of motivating further research.

I. INTRODUCTION

Mobile agents [1] are "programs" that execute in contexts denominated *places* and can autonomously travel from place to place resuming their execution there. Mobile agents are not bound to the computer where they were created; instead, they can move freely between places in different computers.

Mobile agent technology provides many benefits for the development of services for mobile and distributed computing [2], which has encouraged an intensive research work in the area in the last years. Thus, mobile agents offer interesting advantages compared with a traditional client/server approach, thanks to their autonomy, adaptability, and capability to move to remote computers. They can carry the computation wherever it is necessary, without the need of installing specialized servers there (only a generic *mobile agent platform* is needed). For example, instead of communicating a large amount of data from a computer to a mobile device, a mobile agent can move to that computer to process the data locally, minimizing the amount of wireless communications. Similarly, a mobile device can rely on a mobile agent that travels to a fixed computer with the required resources, relieving the overload of the mobile device. This will, in turn, increase the battery life of the mobile device, which is indeed a crucial factor.

In addition, in recent years, mobile devices like cell phones, PDAs (Personal Digital Assistants), and laptops, have become popular and widespread available, due to factors such as their ease of transportation, flexibility, and increasing computing capabilities. However, these devices also have some important limitations, such as intermittent connectivity, low bandwidth, short battery life, and reduced computational power compared to ordinary desktop computers, due to economic and technical factors.

So how do these two worlds cope with each other? On the one hand, the importance of mobile agents for distributed, mobile, and pervasive computing has been highlighted in many works, such as [3], [4], [2]. On the other hand, and despite mobile agent technology has been evaluated in many research works in the context of distributed systems, there are only a few reported experiences on the use of mobile agents in real wireless environments with mobile devices.

In this paper, we test mobile agents in wireless environments and with portable devices. We focus on PDAs because of their popularity and because they achieve a good trade-off between portability and processing capabilities. Our goal is to evaluate whether the mobile agent technology is ready to be used to develop real applications and services in such contexts or, on the contrary, some issues must be addressed before they can be considered a serious alternative. The structure of this paper is as follows. In Section II, we introduce the technological components which are the basis of this work. In Section III, we present some related works. In Section IV, we describe some tests performed with different platforms on different scenarios and compare them. Finally, in Section V, we summarize our conclusions and present some lines of future work.

II. TECHNOLOGICAL CONTEXT

Having mobile agents working on mobile devices involves a set of technological components that will only give the expected functionality when all of them work together. This section describes the most important aspects of these components.

A. Wireless Communications

Mobile devices communicate using radio signals. There are many standards and protocols used for this purpose. We will focus on the two most commonly used today for local-wireless technologies, which will be also considered in our experimental evaluation (see Section IV):

- *Bluetooth* [5] was designed for small devices, such as cell phones and PDAs, and is normally used for the transmission of small amounts of data or to connect to nearby compatible peripherals (e.g., printers, keyboards, or hands-free headsets). It has a limited bandwidth (maximum 3 Mbps), a range of up to 100 meters, and low power consumption.
- *Wi-Fi* [6] networks, based on the IEEE 802.11 standards, allow to expand traditional Ethernet local area networks to places where either cabling is not an option or mobility is desired or needed. Its popularity is growing very fast and almost every laptop computer made in the last years has a Wi-Fi interface which allows it to connect to an also increasing number of public access networks in places such as hotels, airports and restaurants. Compared to bluetooth, it has a higher bandwidth (54 Mbps), a similar range, and a slightly higher power consumption and cost.

Both bluetooth and Wi-Fi are freely available and they are very popular. The main drawback of these short-range technologies when compared to traditional fixed Ethernet networks is that the latency and bandwidth are not only worse, but also variable depending on the distance and obstacles existing between the emitter and the receptor. Moreover, even an established communication could suddenly end if one of the devices moves out of its communication range or enters a shadow area such a tunnel or an underground room (disconnections are frequent).

B. Mobile Devices

Mobile devices could be classified in three basic types:

- *Cell phones*. They are small, light, cheap and with little computation capabilities. Data communications can be carried out through mobile phone networks or via bluetooth. Mobile phone networks have a variable bandwidth depending on the transmission technology (GSM, GPRS, UMTS) and are available almost everywhere, but they have a high economic cost.
- *PDAs or pocket computers*. They are bigger and more expensive than cell phones, but they also have better processing capabilities. There are many architectures (ARM, MIPS, Xscale) and several operating systems that allow the execution of end user applications similar to those available in desktop computers. Communications can be established through bluetooth, and more recently also via Wi-Fi.
- *Laptop computers*. They have capabilities comparable to those of desktop computers. They usually have both Ethernet and Wi-Fi interfaces, but there is also the possibility of using bluetooth for data interchange with small devices.

Currently, the three types of mobile devices mentioned above are starting to mix. Thus, cell phones and PDAs are converging into a single device, called *Smartphone* (PDA with a SIM card). Similarly, laptops and PDAs are mixing into the so-called *Ultra Mobile Personal Computers* (*UMPC*). These devices are still considered very high-end and expensive. Taking into account the market trends, cell phones will not be used in the experimental evaluation of this paper, as they have poor capabilities compared to those of PDAs and will soon be considered obsolete. Thus, we will focus on PDAs and laptops.

C. Mobile Agent Platforms

In order to be able to implement services based on mobile agents, a software platform is required to provide agents a number of basic services, such as the ability to communicate among them and move to other computers. There are many available platforms (e.g., see [7], [8]), some commercial and other of free distribution, which have been used in many works to evaluate the benefits of mobile agents in different scenarios. Most platforms have some common features. First, agents can run in different execution environments (called *contexts*, *containers*, *places*, etc., depending on the mobile agent platform), which may be physically located on the same computer or on different computers connected by a network. Second, many related execution environments can belong to a common group, usually known as *region*, *domain*, or simply *platform*. Finally, agents can communicate among themselves and with other system components, usually regardless of their location.

Many of the existing platforms are implemented in the Java programming language. One of the most important reasons is that Java provides portability, which allows the agents to execute on heterogeneous computers and devices without the need of recompiling or modifying the code. In addition, Java offers a rich set of networking mechanisms (e.g., *Remote Method Invocation -RMI-*) and dynamic loading and caching of classes, among other benefits.

Several mobile agent platforms have been tested experimentally in different research works (e.g., Aglets, Voyager, Tryllian, JADE, SPRINGS, etc.) [7], [8]. However, these platforms have not been tested in mobile devices, where a range of problems can arise (such as a high latency in communications, low bandwidth wireless links, or a limited CPU power). In this paper, we will consider two mobile agent platforms for experimental evaluation (see Section IV):

- *JADE/LEAP*[1]. *Jade* [9] is a very popular agent platform developed by Italia Telecom and a consortium of manufacturers of mobile technologies. It is implemented in Java with RMI as a form of communication and works originally on a fixed platform. It has an additional component, called *LEAP*, which overrides some aspects

[1] `http://jade.tilab.com/`

of Jade to allow its use in mobile devices, with some limitations.

- *SPRINGS* [10]. Developed at the University of Zaragoza (Spain), it is a novel mobile agent platform which stands out for its robustness, scalability and simplicity. It is also implemented in Java and is RMI-based.

We justify the choice of these platforms for experimental evaluation as follows. On the one hand, JADE/LEAP has been designed specifically for mobile devices and it allows a JADE agent to run on a mobile device without modifications. On the other hand, SPRINGS has been found to be executable on a PDA without changes, and so it provides a good example to compare JADE/LEAP with. JADE is the most widely used agent platform at present. SPRINGS has shown a very high scalability in different experimental scenarios [8], [10].

Regarding the two platforms considered in our experimental evaluation: 1) in SPRINGS the execution environments are called *contexts* and the contexts are grouped together in *regions* managed by an *RNS* (*Region Name Server*); 2) in JADE/LEAP, the execution environments are called *containers* and related containers are linked together through a common *main container*.

D. Java Virtual Machines for Mobile Devices

To guarantee portability across heterogeneous devices and computers, Sun Microsystems has created different distributions of the Java platform. From the point of view of this paper, it is interesting to distinguish the following possibilities:

- *J2SE* (*Java 2 Standard Edition*). It is the most widespread edition, used by personal computers and servers running general purpose applications. There are many implementations and *Java Virtual Machines* (*JVMs*) available, usually for free, like the official implementation by Sun Microsystems for PC and Sparc architectures, as well as those offered by companies such as IBM or HP.
- *J2ME* (*Java 2 Micro Edition*). It is designed for small devices and there are two specifications, one intended for more limited equipment such as mobile phones (*Connected Limited Device Configuration -CLDC-*) and the other, more complete, for PDAs (*Connected Device Configuration -CDC-*).

Sun Microsystems does not develop JVMs for mobile devices, but they are only made by third parties. In mobile phones, the manufacturers (e.g., Nokia, Motorola, Sony-Ericsson) include the runtime as an indivisible part of its device. This also happened with PDAs in some cases (e.g., with PDAs by Compaq), but now almost none brings any JVM installed, and so it must be acquired separately. JVMs for mobile devices are neither easy to find nor free, in contrast to the standard Sun's JVM. Among the virtual machines for PDA-like mobile devices we can highlight the following:

- *Jeode*. Some years ago, it was distributed with Compaq PDAs. Then, there was the fusion with HP, and the new models did not bring it. The company who originally made it was acquired by another one which no longer continued its development.
- *Cr-EME*[2]. It is a high quality, J2ME-compliant commercial machine, which has some optional components such as RMI or AWT, which are sold separately. Its biggest problem is its distribution, as it is only sold in lots of 40 units at 1000 dollars.
- *IBM Websphere J9*[3]. Another commercial JVM. It is part of the Websphere development suite. It is also a very good option and also has additional components such as RMI and AWT. The price of the runtime (without the development environment) is only six dollars, which makes it very affordable.
- *MySaifu*[4]. It is an open source implementation under development. It is capable of using some features of AWT components and user interfaces. However, it is still at a very early stage regarding networking aspects (e.g., it does not offer support for RMI).

In our experimental evaluation, we have decided to use IBM Websphere J9 because it has a good quality, it is available in several operating systems and architectures, it has a long evaluation period (90 days vs. 30 days for Cr-EME), and it is very affordable.

III. Related Work

Several mobile agent platforms have been developed and evaluated over the years [7], [8]. However, the focus has always been on distributed environments with desktop computers communicating usually through a wired network. For example, in [11], several software models are proposed for mobile computing environments. An implementation of these models using mobile agents (specifically, Aglets) is provided, and solutions for access to databases through the wireless web are implemented. However, the goal is not to evaluate the performance of existing mobile agent platforms on wireless computers/devices, but to assess the benefits of mobile agents in environments with low bandwidth. In the following, we highlight some works related to the use of mobile agent technology in limited devices with wireless communications.

In [12], they present *Okeanos*, a mobile agent system developed for use in mobile environments. They use the Jeode JVM, RMI transport mechanisms, and a HP IPAQ 5400 with Windows PocketPC 2002. Some interesting aspects are discussed, such as the need of dealing with disconnections (a store and forward mechanism is used in Okeanos), the possibility of unlimited waiting times for agents, and problems that may arise due to the use of *DHCP* (*Dynamic Host Configuration Protocol*) in wireless networks. Unfortunately, no experimental evaluation is presented. Moreover, it is claimed that only local communications among agents are supported in Okeanos.

In [13], they propose the *MobiAgent* system, which is based on Voyager[5]. Its main advantage, according the authors, is that

[2] `http://www.nsicom.com`

[3] `http://www-306.ibm.com/software/wireless/weme/features.html?S_CMP=wspace`

[4] `http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html`

[5] `http://www.recursionsw.com/`

the user does not have to install or configure a mobile agent platform on the mobile device. Instead, MobiAgent provides an interface to interact with mobile agents executing in other places. Whereas not executing a local platform in the device removes several difficulties (such as security problems and the need to implement lightweight execution environments for mobile devices), this also imposes serious limitations to the use of mobile agents.

In [14], they present an overview of multi-agent platforms created for limited devices, such as *MobiAgent*, *kSACI*, *MAE*, *AgentLight*, *MicroFIPA-OS*, and *LEAP*. They distinguish between: 1) *portal platforms* (e.g., MobiAgent), which do not execute mobile agents on the mobile devices but just allow the interaction with mobile agents running on remote hosts; 2) *embedded platforms*, which are executed totally on the mobile device; and 3) *surrogate platforms* (e.g., kSACI), which are executed only partially on the mobile device. However, neither they focus on mobile agents nor perform an experimental evaluation.

Finally, in [15], they develop a mobile agent architecture to execute on limited devices based on the *CLDC* of *J2ME*. The authors of this work claim that, in that case, running an entire agent on a mobile device is not an option, and so they split an agent in two parts (code and data), such that only the data reside on the mobile device. Unfortunately, no experimental evaluation is provided.

IV. Experimental Evaluation

In the tests presented in this section, we will evaluate the behavior of the mobile agent platforms JADE/LEAP (version 3.5) and SPRINGS (version 1.0) when they run on mobile devices and wireless communications are used. To achieve this, a comparison is carried out by running some test programs with the same functionality on both platforms, in fixed and mobile computers. Times are measured between key points of the execution in order to compare the performance of the platforms in the different scenarios. In the rest of this section, we first describe our experimental environment, and then we present and perform some tests in three different scenarios.

A. Experimental Environment

Our environment for evaluation is composed of the following elements:

1) A *fixed desktop computer*, which is an Intel Pentium 4 at 2.0 GHz with 768 MB of RAM, running Ubuntu Linux with kernel 2.6.20. The network interface is a standard 100 Mbps Fast Ethernet. The Sun JVM 1.5 is used.
2) A *mobile computer*. Depending on the experiment, the mobile computer used may be:
    - A *laptop computer*, which is an Intel Centrino with two 1.66 GHz cores and 2 GB of RAM, running Ubuntu Linux with kernel 2.6.22. The fixed network interface is a standard 100 Mbps Fast Ethernet, and the wireless interface is a standard 802.11a/b/g Wi-Fi integrated card. The Sun JVM 1.5 is used.
    - A *PDA*. We have performed tests with two different models:
        - A *Fujitsu-Siemens Loox 720*, with a X-Scale processor at 520 MHz, 128 MB of RAM, running Windows Mobile 2003. Its connectivity options include an integrated bluetooth adapter (class 2, 10-meter range), a USB 1.1 cable, and Wi-Fi.
        - An *HP iPAQ 1940*, with a Samsung S3C processor at 266 MHz, 64 MB of RAM, running Windows Mobile 2003, USB 1.1, and bluetooth (class 2, 10-meter range), but without Wi-Fi support.

    Unless specified otherwise, the model used by default in the experiments is the Fujitsu-Siemens Loox 720, which has superior features and also offers Wi-Fi communications. The JVM used on the PDA, in any case, is the IBM Websphere J9 version 11 for Windows Mobile.

In this environment with a fixed desktop computer and a mobile computer (laptop or PDA), we consider different connectivity options:

- *Wi-Fi connections* (PDA and laptop computer). A Wi-Fi router (802.11a/b/g) routes Wi-Fi communications and is located eight meters away of the mobile computer, with light obstacles in the middle (two walls, which reduce the signal intensity to 65%-75%). The maximum bit rate is 54 Mbps. The router's fixed interface connects to the LAN (Local Area Network) where the fixed computer is.
- *Bluetooth connections* (PDA). The PDAs have a bluetooth interface and therefore we can also evaluate this type of network connection.
- *USB wired connection* (PDA). This is the only wired connection available on PDAs, as they lack an Ethernet network interface. Thus, we can use a USB cable to connect a PDA to a fixed computer with a network connection, enabling Internet access on the PDA. Although PDAs have been designed to work (usually) with a wireless connection, we will evaluate this situation as a baseline for comparison.
- *Ethernet wired connection* (fixed desktop computer and laptop computer). The fixed desktop computer has this type of network connection. We sometimes also use this type of connection on the laptop computer, in order to compare the performance with other connection possibilities (e.g., Wi-Fi).

A summary of the infrastructure used for the tests is shown in Figure 1. Notice that this simple infrastructure is enough for our purposes, as our main goal is to see how the mobile computer (PDA or laptop) performs (the performance of mobile agents in fixed computers connected through wired networks has been studied in other research works). Our intention is to evaluate SPRINGS and JADE/LEAP. For SPRINGS, the *RNS* (*Region Name Server*) executes on the fixed computer. For JADE/LEAP, the *main container* and the *AMS* (*Agent Management System*) run on the fixed computer.

Each experiment is repeated five times and average measures are reported. On a few occasions, some outliers may occur due to a temporary wireless network instability; we have decided not to remove these outliers so as to allow the obtained results reflect the existing unpredictability.

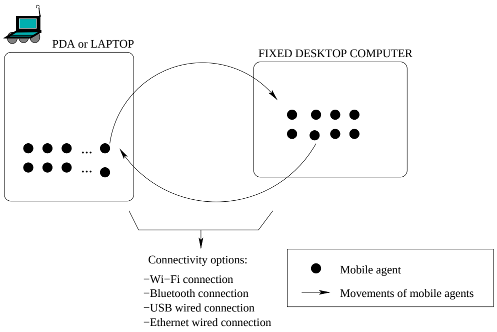

Fig. 1. Environment for experimental evaluation

B. *Scenario 1: 20 Agents Moving*

The goal of this test is to evaluate the performance and reliability of agent mobility with wireless communications and portable computers and devices. For this, 20 agents are created (one after another) on the mobile computer. Then, each agent performs 5 hops between the mobile computer and the fixed computer. We choose these parameters (20 agents and 5 hops) to evaluate a scenario of intermediate complexity (the test is neither challenging nor trivial). In this scenario, we perform several experiments. We will report delays for the different agents to show the variability of the results.

1) Time to Create the Agents on a Laptop, with Wired and Wi-Fi Communications: In Figures 2 and 3, we show the average agent creation time for each of the 20 agents created in the test, with JADE/LEAP and SPRINGS, respectively, using a laptop. We can see that the average creation times are small and similar for both JADE/LEAP and SPRINGS. The delays are, in general, slightly higher when the laptop uses Wi-Fi communications (instead of wired communications) to communicate with other elements in the agent platform (as part of the creation of the agent). However, the scale of the delays shown in the figure is in both cases very small (a few dozens of milliseconds). We can also observe that the variability of the results increases when the communication is via Wi-Fi (different context factors affect the quality of the wireless link). The high delays for the first agents could be due to internal initializations of the JVM.

2) Time to Complete an Agent's Itinerary Using a Laptop, with Wired and Wi-Fi Communications: In Figures 4 and 5, we present the total time needed by each agent to complete its entire itinerary, with JADE/LEAP and SPRINGS, respectively. In this case, we can observe that the performance of JADE/LEAP is slightly better than SPRINGS in this scenario. Thus, a JADE/LEAP agent needs an average of 1.4 seconds to complete the itinerary (five hops) in the Wi-Fi scenario, whereas a SPRINGS agent needs about 2.5 seconds. The reason could be that SPRINGS has not been specially adapted to scenarios with Wi-Fi communications. Regarding the results obtained for JADE/LEAP, we can observe that the performance results obtained with Wi-Fi are better than those obtained with wired communications. This may seem surprising, but the large Wi-Fi bandwidth is probably no bottleneck in this experiment.

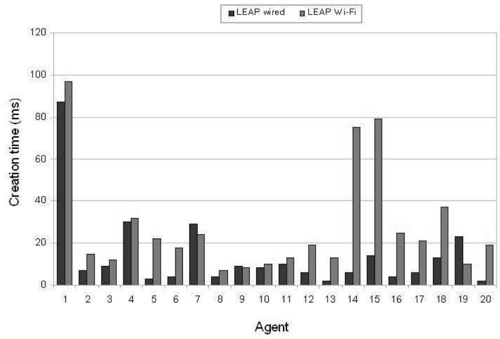

Fig. 2. Time to create the JADE/LEAP agents on a laptop, with wired and Wi-Fi communications

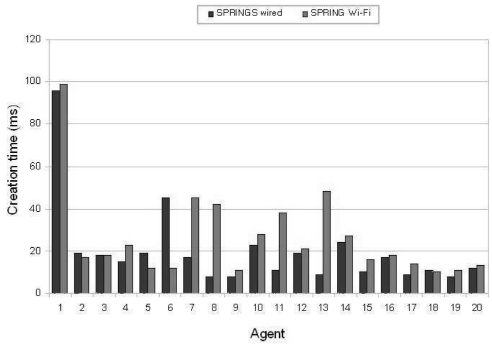

Fig. 3. Time to create the SPRINGS agents on a laptop, with wired and Wi-Fi communications

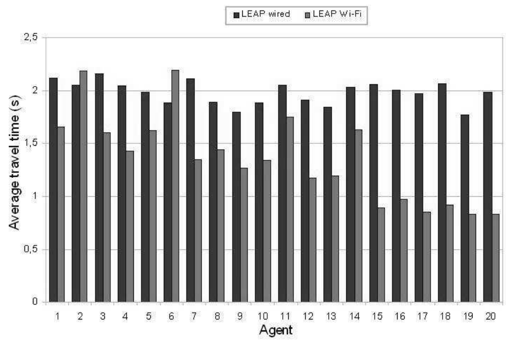

Fig. 4. Average itinerary times for JADE/LEAP agents, using a laptop, with wired and Wi-Fi communications

3) Time to Create the SPRINGS Agents and Complete an Agent's Itinerary Using a PDA Connected via USB, for two different PDA models: Now, we repeat the previous experiments using a PDA. In this experiment, we assume a wired communication, by connecting the PDA via USB 1.1

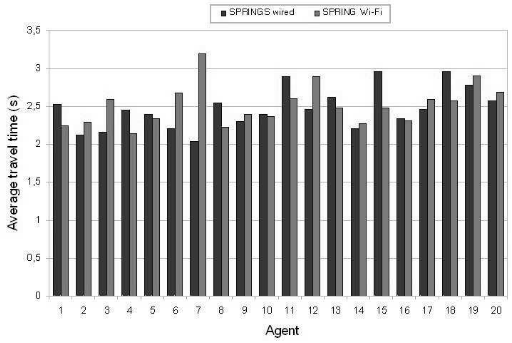

Fig. 5. Average itinerary times for SPRINGS agents, using a laptop, with wired and Wi-Fi communications

(up to 12 Mbps) to a fixed computer with a wired connection to the Internet. The results are shown in Figures 6 and 7. First, we would like to emphasize that we were unable to realize the experiment with JADE/LEAP agents running on the PDA, due to a bug in JADE/LEAP arising when a mobile agent moves out of the PDA (the code is simple and it runs without any problem on a laptop, as we have seen before), which we expect to be fixed in next releases. Second, we can notice a high increase in the experienced delays when a PDA is used instead of a laptop. For example, the average agent creation time with the SPRINGS agents on a laptop (see Figure 3) is about 30 ms, and with the agents on a PDA it reaches about 750 milliseconds. Similarly, in the experiment with a laptop and Wi-Fi (see Figure 5) the average itinerary time is around 2.5 seconds with SPRINGS, whereas if the execution environment is hosted on a PDA (with a wired connection) the average itinerary time is about 5 seconds. The high delays for the first agents could be due to internal initializations of the JVM. Finally, the Fujitsu-Siemens Loox 720 obtains a slightly better performance than the HP IPAQ 1940. This, as well as the fact that the HP iPAQ 1940 does not support Wi-Fi, justifies the choice of the Fujitsu-Siemens Loox 720 in our experiments.

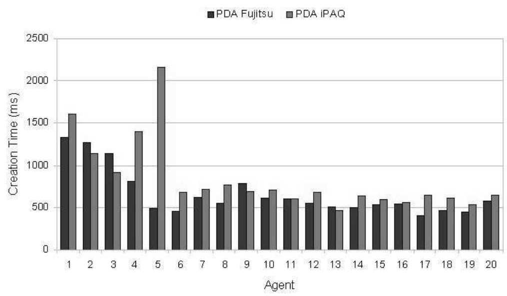

Fig. 6. Time to create the SPRINGS agents on a PDA, connected with USB

4) Time to Create the SPRINGS Agents and Complete an Agent's Itinerary Using a PDA Connected via Wi-Fi: We now carry out the same experiment but this time using a Wi-Fi connection. In Figure 8, we show the average time needed to create each SPRINGS agent on the PDA. On average, about 0.6 seconds are needed, which is greater than in case a laptop is used (see Figure 3) but smaller than if the PDA is connected

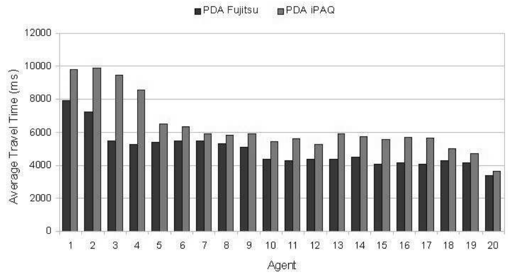

Fig. 7. Average itinerary times for SPRINGS agents, using a PDA, connected with USB

via USB (as shown in Figure 6). In Figure 9, we show the average time needed by each agent to complete its itinerary. On average, about 8 seconds are needed, which is much higher than the time needed when we replace the PDA by a laptop (see Figure 5) and slightly higher than the time needed with a PDA with USB connection (see Figure 7).

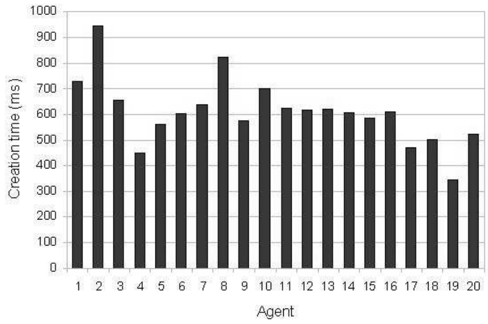

Fig. 8. Time to create the SPRINGS agents on a PDA, with Wi-Fi

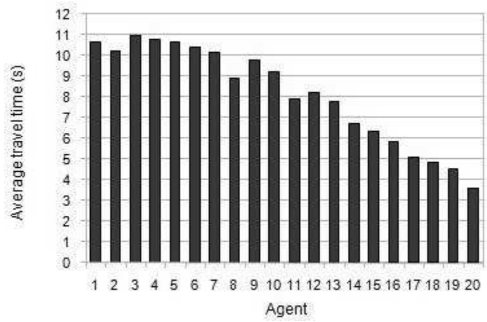

Fig. 9. Average itinerary times for SPRINGS agents, using a PDA, with Wi-Fi

5) Time to Create the SPRINGS Agents and Complete an Agent's Itinerary Using a PDA Connected via Bluetooth: We also performed the previous experiment using a Bluetooth connection instead of Wi-Fi. Unfortunately, when evaluating the Bluetooth connection with a mobile agent platform, we found that it was not very reliable. Thus, in some occasions, the connection is dropped shortly after it has been established and cannot be recovered during the test. The average agent creation time with this experiment is 0.83 seconds, and the average time needed to finish an itinerary is 9 seconds. We omit the figures for space reasons.

C. Scenario 2: One Mobile Agent and One Static Agent Calling the Mobile Agent

We now extend the experiments presented in the previous section to include also communications among the agents. In this case, only two agents are created on the mobile computer. One of these agents remains fixed on the mobile computer and sends synchronous messages to the other, one message every five seconds. The reception of the message triggers the movement of the receiving agent to the other computer. In each test, we send 20 messages, in total, to the receiving agent. In this scenario, we perform several experiments.

1) Time to Communicate with the SPRINGS or JADE/LEAP Mobile Agent, Using a Laptop, with Wired and Wi-Fi Communications: In Figures 10 and 11, we show, with SPRINGS and JADE/LEAP, respectively, and for each of the 20 communications in a test, the average time interval between the moment a message is sent and the moment when the receiving agent begins to move, which is given by the communication delay. On average, Wi-Fi communications are slightly slower, although in certain communications the opposite may be true (e.g., in our experiment with SPRINGS, for communication number 2), due to the unpredictability of communications. The variability of Wi-Fi communications is also higher than that of wired communications. We can also observe that SPRINGS takes a slightly smaller amount of time to communicate in a wired network (82 s vs. 86 s), but it takes more time in a Wi-Fi network (146 s vs. 93 s).

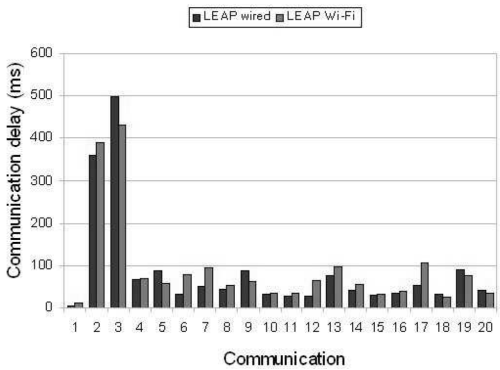

Fig. 10. Communication delays for JADE/LEAP, using a laptop, with wired and Wi-Fi communications

2) Time to Communicate with the SPRINGS Mobile Agent, Using a PDA, with a USB Connection: In Figure 12, we evaluate the communication delays for SPRINGS when the mobile computer is a PDA, connected through the fixed network through a USB cable. Around 1 second (on average) is needed to communicate with an agent.

3) Time to Communicate with the SPRINGS Mobile Agent, Using a PDA, with a Wi-Fi Connection: In Figure 13, we evaluate the communication delays for SPRINGS when the mobile computer is a PDA, but this time using Wi-Fi. In this case, we can see how the communication delays are similar as in the previous case where the communication is through the USB connection (an average of 1.3 seconds vs. 1 second per communication) although more variable with Wi-Fi.

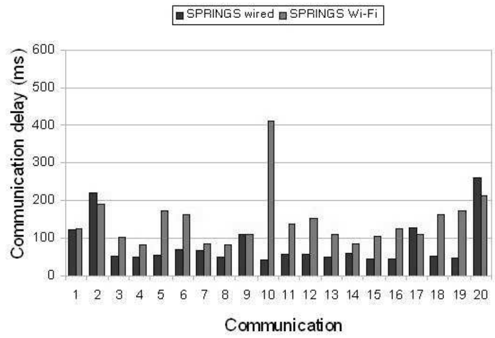

Fig. 11. Communication delays for SPRINGS, using a laptop, with wired and Wi-Fi communications

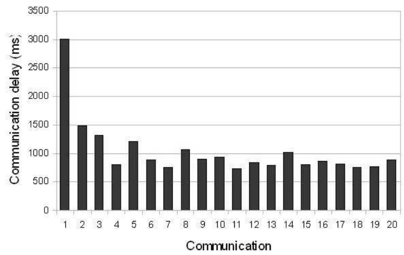

Fig. 12. Communication delays for SPRINGS, using a PDA with USB

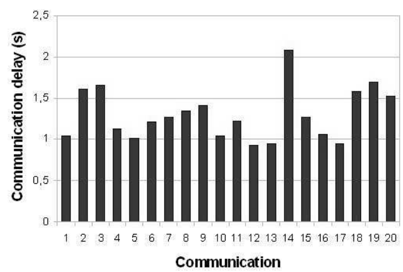

Fig. 13. Communication delays for SPRINGS, using a PDA with Wi-Fi

D. Scenario 3: Testing RMI vs. Sockets

Motivated by the fact that JADE/LEAP (which is implemented using sockets) achieves sometimes a slightly better performance than SPRINGS (implemented using RMI) with portable computers and Wi-Fi communications (e.g., compare the Figures 4 and 5), we have decided to perform a final experiment, where we compare the use of RMI with the use of sockets. In this experiment, the mobile computer is a PDA and it needs to communicate with an object located on the fixed desktop computer. We perform 20 communications in each test. In Figure 14, we show the average time needed by each RMI communication, for different connectivity alternatives:

PDA communicating with bluetooth, via Wi-Fi, and using a USB cable. We omit in the figure the time needed by the first RMI communication, which is significantly higher (probably, due to the way RMI is internally implemented): an average of 535 ms with USB, 2052 ms with bluetooth, and 353 ms with Wi-Fi. Without considering the first communication, the average delay is about 314 ms with USB, 144 ms with bluetooth, and 165 ms with Wi-Fi. Using a USB cable or Wi-Fi yields similar results. Leaving aside the RMI first communication, Bluetooth performs slightly better. In Figure 15, we perform the same experiment using sockets. In all the cases, the average communication time is smaller than when using RMI (118 ms with USB, 67 ms with bluetooth, and 47 ms with Wi-Fi).

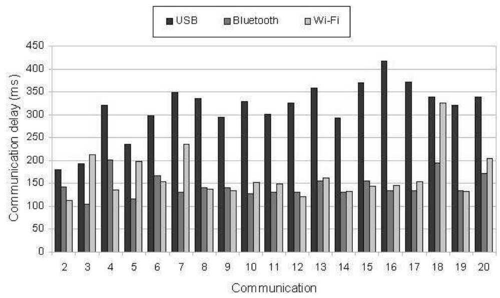

Fig. 14. Communication delays using a PDA and RMI

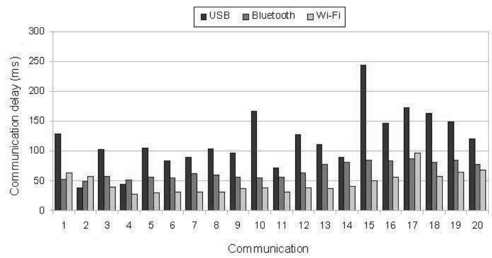

Fig. 15. Communication delays using a PDA and sockets

V. Conclusions and Future Work

In this paper, we have evaluated several mobile agent platforms in wireless environments, in a variety of conditions and scenarios. In the following, we summarize our conclusions:

- There is a bug in the current implementation of mobile agents in JADE/LEAP (version 3.5), as they do not work on PDAs. This problem does not appear if the mobile computer is a laptop or if the agents do not move.
- Wi-Fi has a good bandwidth. Therefore, there is only a slight decrease in performance when using Wi-Fi instead of wired communications. Nevertheless, the delays of Wi-Fi communications are subject to more variability.
- The performance of a mobile agent platform depends on the device used. Thus, the observed delays are significantly higher when a PDA is used, due to its limited capabilities rather than the type of connection used.
- Bluetooth (at least, class two) has not behaved reliably in our experiments with mobile agents. Failed communications can be retried but high delays may arise anyway if a connection is dropped.
- Regarding agent mobility, JADE/LEAP is slightly better than SPRINGS when there are wireless communications. However, agent communications are faster in SPRINGS.
- With PDAs, sockets are more efficient than RMI.

As future work, we plan to extend our experiments to evaluate other conditions (e.g., cause interferences or bandwidth problems, move around with the devices, etc.). We will also analyze how to adapt SPRINGS to perform better in these environments. Two possible adaptations could be considered: 1) use sockets instead of RMI; and 2) reduce the capabilities of the places (execution environments for mobile agents) hosted on mobile devices, to make them more light-weight.

Acknowledgements

Work supported by CICYT project TIN2007-68091-C02-02.

References

[1] D. Milojicic, F. Douglis, and R. Wheeler, *Mobility: processes, computers, and agents*. New York, NY: ACM Press, 1999.

[2] C. Spyrou, G. Samaras, E. Pitoura, and P. Evripidou, "Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies," *Mobile Networks and Applications*, vol. 9, no. 5, pp. 517–528, 2004.

[3] I. Satoh, "Mobile agents for ambient intelligence," in *First Int. Workshop Massively Multi-Agent Systems I, MMAS'04*. London, UK: LNCS vol. 3446, Springer-Verlag, 2005, pp. 187–201.

[4] A. Selamat and H. Selamat, "Analysis on the performance of mobile agents for query retrieval," *Information Sciences*, vol. 172, no. 3-4, pp. 281–307, 2005.

[5] R. Morrow, *Bluetooth: Operation and Use*. McGraw-Hill Professional, 2002.

[6] F. Ohrtman and K. Roeder, *Wi-Fi Handbook : Building 802.11b Wireless Networks*. McGraw-Hill Professional, 2003.

[7] K. Ludwig, A. Josef, W. E. Edgar, S. Wolfgang, and G. Franz, "Using mobile agents in real world: A survey and evaluation of agent platforms," in *Second Intl. Workshop on Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems, at the 5th Intl. Conf. on Autonomous Agents (AA'01)*. ACM Press, May 2001.

[8] R. Trillo, S. Ilarri, and E. Mena, "Comparison and performance evaluation of mobile agent platforms," in *The Third Intl. Conf. on Autonomic and Autonomous Systems (ICAS'07)*. IEEE Computer Society, June 2007.

[9] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley, March 2007.

[10] S. Ilarri, R. Trillo, and E. Mena, "SPRINGS: A scalable platform for highly mobile agents in distributed computing environments," in *4th Intl. WoWMoM 2006 Workshop on Mobile Distributed Computing (MDC'06)*. IEEE Computer Society, June 2006, pp. 633–637.

[11] C. Spyrou, G. Samaras, P. Evripidou, and E. Pitoura, "Wireless computational models: Mobile agents to the rescue," in *10th Intl. Workshop on Database & Expert Systems Applications (DEXA'99)*. Washington, DC, USA: IEEE Computer Society, 1999, p. 127.

[12] M. Friedrich, K. Terfloth, G. Nusser, and W. Küchlin, "Mobile agents: A construction kit for mobile device applications," in *Intl. Conf. on Internet Computing (IC'04)*, H. R. Arabnia and O. Droegehorn, Eds. CSREA Press, 2004, pp. 186–194.

[13] L. Vasiu and Q. Mahmoud, "Mobile agents in wireless devices," *IEEE Computer*, vol. 37, no. 2, pp. 104–105, 2004.

[14] C. Carabelea and O.Boissier, "Multi-agent platforms for smart devices: Dream or reality?" in *Smart Objects Conf. (SOC'03)*, May 2003, pp. 126–129.

[15] A. Wang, C. Srensen, and E. Indal, "A mobile agent architecture for heterogeneous devices," in *Intl. Conf. on Wireless and Optical Communications*, 2003, pp. 14–16.