



Towards an Asynchronous Dissemination and a Safe Deployment of Lightweight Programs in Mobile Networks

Hervé Roussain, Nicolas Le Sommer, Frédéric Guidec

► To cite this version:

Hervé Roussain, Nicolas Le Sommer, Frédéric Guidec. Towards an Asynchronous Dissemination and a Safe Deployment of Lightweight Programs in Mobile Networks. WoWMoM'05, Jun 2005, Taormina, Italy. pp.481-483. hal-00341750

HAL Id: hal-00341750

<https://hal.science/hal-00341750>

Submitted on 25 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards an Asynchronous Dissemination and a Safe Deployment of Lightweight Programs in Mobile Networks*

Hervé Roussain, Nicolas Le Sommer and Frédéric Guidéc

Valoria Laboratory, University of South Brittany

{Herve.Roussain, Nicolas.Le-Sommer, Frederic.Guidéc}@univ-ubs.fr

Abstract

This paper presents the design and the implementation of a platform allowing owners of mobile devices to exchange software application programs using a peer-to-peer scheme in hybrid networks as well as in pure dynamic networks, such as those composed of volatile communicating devices (e.g. PDA, laptops). This platform provides means to disseminate announcements or discovery requests of application programs in networks, thus permitting owners of mobile devices to announce to their neighbours what applications they can provide, while discovering themselves what applications are available in their neighbourhood.

1 Introduction

Nowadays we observe an increasing proliferation of mobile devices (e.g. personal digital assistants, laptops) equipped with wireless communication interfaces (e.g. IEEE 802.11). The prospect of using these devices to provide people with ubiquitous application services appears as an attractive one. In mobile networks, especially in ad hoc networks, devices can be highly mobile and volatile because the owners of these devices themselves move quite a lot, and because these mobile devices usually have a limited power-budget and their owners tend to activate and deactivate them frequently. Moreover, the wireless interfaces of these devices have a limited transmission range. For the owners of mobile devices it is thus difficult to have a permanent access to services provided by infrastructure networks. In order to provide people with a better access to services, application programs providing all (or a part of) these services are traditionally deployed on the own devices of people. However, in mobile network the client/server deployment scheme –that is commonly use in infrastructure network– is thus hardly applicable in wireless mobile net-

works, since no device are stable and reachable enough to act as servers of application programs in these networks.

In this paper, we present the design and the implementation of the JASON (Java Ad hoc Services on ad hOc Networks) platform with which we investigate the deployment of application programs on wireless ad hoc networks. This platform allows owners of mobile devices to exchange software application programs dynamically in hybrid networks as well as in pure dynamic networks, such as those composed of volatile communicating devices. Seen from this point of view, it provides means to disseminate announcements or discovery requests of application programs in networks, thus permitting owners of mobile devices to announce to their neighbours what applications they can provide, while discovering themselves what applications are available in their neighbourhood. The JASON platform also offers functionalities to download the desired programs from remote devices, and provides a deployment environment supporting dynamic resource access contracting and dynamic application monitoring in order to run the programs thus obtained in a secure manner.

In the remainder of this paper we present briefly the different functionalities of the JASON platform, that is, a mechanism that supports the dissemination, the discovery and the delivery of software application, and a secure environment for the deployment of downloaded applications.

2. Overview of the JASON platform

Our platform allows the owners of mobile devices to exchange application programs, and to run the programs they thus obtain in a secure manner. The general architecture of the JASON platform is organised in three main elements : the graphical user interface, that is shown in figure 1, the context-aware and collaborative deployment manager, and the secure runtime environment.

*This work is supported by the French Conseil Régional de Bretagne under contract B/1042/2002/012/MASC.

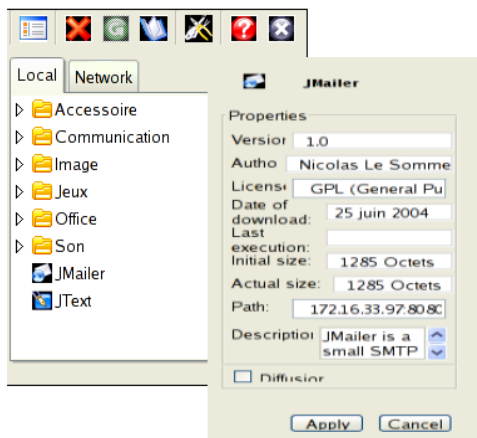


Figure 1. Screenshots of Jason's GUI.

3. Context-aware collaborative software deployment

In JASON, the discovery and the announcement of application programs relies on the dissemination of descriptors. A software package descriptor provides information about its identity, its content, its dependencies regarding other packages, etc. When the owner of a mobile device initiates the deployment of a given application, the deployment manager uses the dependencies described by the descriptor of the application to discover what packages are needed for that application, and decide which packages need to be obtained –if possible– from neighbouring devices.

Descriptors also include information about the resources required by application programs for running. This kind of information is meant to help deploy application programs in ad hoc networks including mobile devices with different hardware and software characteristics. Indeed, all application programs are not necessarily equivalent regarding resource access conditions: some applications can run perfectly with few resources and without any guarantee about the availability of these resources, whereas other programs require important amounts of resources and guarantees regarding their availability. This resource requirement description can typically be used by the deployment manager in order to evaluate whether a program can be deployed on the mobile device, as well as to monitor the program at runtime so as to check that this program behaves as expected.

When the owner of a mobile device initiates the deployment of an application on its device, the deployment manager examines the dependencies mentioned in the descriptor of this application. Based on this information the deploy-

ment manager can decide if the application can be installed immediately (which is possible if all the required packages are already available in the local repository), in which case the installation request is passed directly to the runtime environment of the platform. However, If some packages are missing in the repository, the deployment manager initiates a background process that aims at collecting these packages from neighbouring devices. Since all devices are highly mobile and volatile in the kind of network we consider, the collecting process of missing packages can sometimes be a lengthy one. In order to account for the constraints presented by the network, a number of strategies can be used by the deployment manager. For example, each deployment manager can be configured so as to announce periodically to its neighbours the packages it owns locally. This is done by sending in the network the descriptors of these packages, rather than the packages themselves. When a device receives a such announcement, its deployment manager examines the descriptors contained in this announcement. For each descriptor that matches a desired package, the deployment manager can then send a request to the device that made the announcement, asking that the package mentioned in this descriptor be sent in the network.

Alternatively –or complementarily– to the abovementioned process, a deployment manager that misses some packages can periodically send a request for these packages in the network, thus inviting its neighbours to announce – or send directly– packages that match the request. In order to take ad hoc networks characteristics into account, the JASON platform uses the peer-to-peer, asynchronous document dissemination scheme described in [2].

By combining these two basic scenarios, and by adjusting for example the period for sending announcements and requests in the network; a number of deployment strategies can be devised and implemented. In the current implementation of the JASON platform, the owner of a device is responsible for setting parameters (such as transmission periods) in order to define how the deployment manager should behave. In the future we plan to define heuristics so that each device adjusts automatically to its environment. For example a device may announce spontaneously the packages it owns whenever other devices appear in its neighbourhood.

4. A secure deployment of programs

4.1. Contract-based approach of resource utilisation

In order to be deployed on the JASON platform, the downloaded application programs must contractualise their resource access conditions with the platform. Programs that neglect to do that are either simply rejected or run using a

predefined set of resources. This contractualisation process aims at checking whether the resource access requirements exhibited by a program do not violate the resource access policy enforced by the platform, and also aims at working out if these requirements can be satisfied, and thus to decide if the program can be allowed to run. To perform this process, downloaded programs can be designed to define dynamically a contract object using the resource requirement information included in their descriptor, and can be developed to submit this contract to the platform. The platform makes it possible to submit several contracts before subscribing a specific one. In JASON, a contract can pertain on qualitative access conditions (i.e. resource access permissions) as well as on quantitative access conditions (i.e. resource quotas).

In the JASON platform, contracts are evaluated by a broker implementing an admission control based on a resource reservation scheme. This admission control provides hosted programs with a certain level of quality of service regarding the availability of the resources that are necessary for their execution. A contract is declared acceptable by the broker if all of its clauses can be satisfied. If the contract cannot be satisfied, this resource broker builds a report specifying which clauses are rejected. This report is expected to help the deployed program in the definition of its next contract submissions. When one of the contracts submitted is marked as acceptable, the program can initiate a contract subscription process with the platform. Since the platform may be carrying out several negotiations concurrently with many candidate programs, the status of resources may change between the time a submitted contract is declared acceptable by the resource broker, and the time this contract is subscribed. Consequently, whenever an application program subscribes a contract, the terms of this contract are examined again by the resource broker, if only to check that they are still valid. If so, then the resources for which the candidate program requires availability guarantees are reserved by the resource broker for this program. In JASON, contracts can be renegotiated as and when needed, by negotiating and subscribing amendments. Amendments specify the modifications that must be applied on a contract (addition, deletion, modification of clauses). The principles underlying the negotiation and the subscription of amendments are similar to those of contracts. The contractualisation process implemented in JASON is more detailed in [3].

4.2. Monitoring of applications programs

Since programs obtained from mobile devices can be potentially untrusted, every program running on the JASON platform is pro-actively monitored in order to check whether it uses resources in accordance with the contract it subscribed with the platform.

The JASON platform provides a specific implementation

of a resource monitor for each basic resource type (e.g. CPU, memory, file system, network). Each resource monitor can be configured dynamically. The role of a resource monitor is to supervise the utilisation of the resource –or collection of resources– considered in the directives it received, and to ensure that this utilisation remains in accordance with those directives.

When a resource monitor detects a contract violation, it applies a pre-defined sanction. In the current implementation of the platform, several kinds of sanctions are actually applicable to faulty programs. These sanctions range from a simple warning addressed to a faulty program (using an event based model), up to the immediate termination of this program. This monitoring principle is plentifully detailed in [1].

5. Conclusion and future work

In this article, we have presented a platform called JASON which provides owners of mobile devices with means to discover application programs that are offered to them spontaneously by the other members of their network. It also provides them with a secure runtime environment to host application programs they download. The first experiments of spontaneous announcement, discovery and hosting of application programs we have made prove to be relevant and promising. It is our conviction that the development of JASON should be carried on. In this viewpoint, we plan to improve the discovery functionalities implemented in the JASON platform. Moreover, we intend to consider not only announcement and discovery of application programs but more generally of all kinds of documents (e.g. textual documents, images).

References

- [1] F. Guidéc and N. Le Sommer. Towards resource consumption accounting and control in Java: a practical experience. In *16th European Conference for Object-Oriented Programming (ECOOP 2002)*, Malaga, Espagne, jun 2002. Workshop on Resource Management for Safe Language.
- [2] F. Guidéc and H. Roussain. Asynchronous Document Dissemination in Dynamic Ad Hoc Networks. In *Second International Symposium on Parallel and Distributed Processing and Applications (ISPA'04)*, Hong-Kong, China, December 2004.
- [3] N. Le Sommer and F. Guidéc. A Contract-Based Approach of Resource-Constrained Software Deployment. In *First International IFIP/ACM Working Conference on Component Deployment (CD 2002)*, June 2002.