Mobility Support in a Publish/Subscribe Middleware

Mauro Caporuscio
Dipartimento di Informatica
Università degli Studi dell'Aquila
I-67100 L'Aquila, Italy
caporusc@univaq.it

Abstract

This work focuses on the integration of a publish/subscribe middleware service with mobile components and applications. Publish/subscribe middleware is considered a good platform for the integration of loosely-coupled components on a large-scale. However, none of the implementations of publish/subscribe middleware available today is specifically designed to support mobile applications. Such applications are gaining popularity with the introduction of wireless data communication and portable computing devices such as PDAs or 3G cellular phones. Our idea is therefore to study how to design a publish/subscribe middleware capable of serving mobile, wireless applications. This effort consists of two parts: First, we studied the performance of an implementation of a publish/subscribe middleware built on top of a wireless network. Second, we studied the additional service-level requirements posed by mobile, wireless applications over the publish/subscribe middleware. In this paper, we present the results of our performance study, and the design and implementation of an auxiliary service-level support for mobile applications.

1 Introduction

This work is concerned with mobile applications that use a publish/subscribe infrastructure. In particular, this work consists of (1) a case study on the deployment of a publish/subscribe middleware on top of a wireless communication service, where mobility is supported at the network level, and (2) a design and initial implementation of a mobility support service realized within the publish/subscribe middleware.

The increasing size and performance of computer networks is generating a new phenomenon: networks are becoming pervasive and ubiquitous. *Pervasive* means that network connectivity is going to be a basic feature of any computing facility, while *ubiquitous* refers to the ability to use network connectivity independently of the physical location of the user. In this context (usually referred to as a *wide-area network*), applications are characterized by the fact that they are loosely coupled, asynchronous, and heterogeneous. This promotes a class of software system based on the abstract design called *event interaction*, which in turn is supported by an emerging infrastructure called an *event notification service* [5].

Developments in wireless technology are freeing application hosts from a constrained, fixed physical location in the network and enabling the practical realization of the idea of *mobile computing*. In fact, portable computers (such as laptops and PDAs) are growing in popularity while they are shrinking in size. This process of miniaturization, combined with the emergence of high-speed wireless communications, allows users to use portable device with on-demand connections. In this scenario (showed in Figure 1), mobile users can move together with

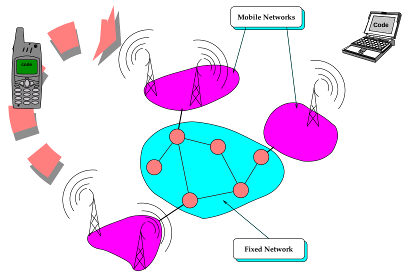

Figure 1: Application hosts can be mobile.

their hosts across different physical locations, while remaining connected to the network through wireless links.

In addition to the mobility of hosts, new techniques based on code migration [6] have been developed to allow applications to move from host to host (see Figure 2). These mobile applications, often referred to as "mobile agents", aim to optimize information retrieval and other similar tasks by moving close to the data stores of interest, where they can execute their queries with low latency and network usage.

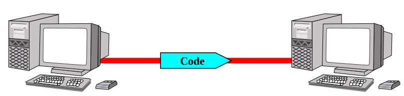

Figure 2: Applications can migrate from host to host.

In this study we are interested in combining the benefits of mobile applications (moving along with their host, or migrating from host to host) with the communication services offered by an advanced publish/subscribe service. In particular, we focus on the SIENA distributed event-notification system [4], hosted over a General Packet Radio Service (GPRS) [1] network. We studied this integration in a situation in which the mobility of clients is transparently managed at the network level.

In integrating publish/subscribe technology with mobile applications, we have two general choices of architecture: In one case we could simply attach the publish/subscribe system on top of a network that offers native support for mobility. In the opposite case, we could have the publish/subscribe system handle mobility without any direct support from the underlying network layers. In this latter case, the publish/subscribe service would implement an additional set of services designed to support mobile applications. These two alternative methods, detailed below, characterize the contribution of this work.

Section 2 describes how we put SIENA middleware on top of the GPRS network and how we combined the concept of *physical mobility* with the event-based architecture. It also explains the experiments we conducted in order to understand SIENA's performance in a mobile environment. Section 3 talks about our research in *logical mobility* exploring the problems that it implies and describes how we allowed SIENA to support the mobility of its clients. This section also presents the algorithms we have been designing to solve the problems explained. Finally in Section 4 we draw some conclusions summarizing our experience and discussing future developments.

2 Evaluating SIENA in a Wireless Network

In order to evaluate the behavior of SIENA and its demands on the communication resources of the wireless network, we developed a test application (a distributed auction system) that we used in several simulated scenarios. Developing such an application required us to port the SIENA client-side library to the Java™ 2 Micro Edition, a platform specifically targeted at mobile devices such as cell phones and PDAs. The resulting application and library allowed us to run experiments on a simulated PDA, in combination with a highly configurable GPRS network emulator called Seawind [8].

The primary goal of our experiments was to evaluate the impact of deploying SIENA onto the wireless GPRS network. We did this from two different perspectives. The first was to gather data characterizing the performance of three different low-level transport mechanisms (UDP, TCP, and a "keep-alive" TCP that attempts to reuse TCP connections) on the wireless network. The second was to compare these results with baseline data collected on a local-area, wired network.

The results of the experiments gave us an initial indication of whether a seamless integration of wired and wireless communication is feasible for a publish/subscribe communication service [3]). For example, the data shown in Table 1 gives an indication of the circumstances that lead to different notification loss rates. As we would expect, the highest loss rate occurs at an error probability of 10^{-3} under the DROP error-handling mode. We can also see that the keep-alive connector is the most sensitive to increasing error rates and decreasing quality of error-handling service. By comparing these results with the baseline overhead one simple thing we can note is the high overhead of TCP. In fact in the wireless case it is approximately twelve packets per notification, considerably higher than in the local-area, wired case.

3 Mobility Support in SIENA

We studied how to support mobile applications that use the SIENA publish/subscribe system implemented over a wired network. We consider mobile applications that either move along with their host (e.g., because they execute on a laptop or a PDA) or move from host to host using mobile code technology. Regardless of the technology supporting mobility, we assume that the application can detach from one SIENA access point, travel to another network location, and reconnect to another SIENA access point (refer to Figure 3).

To support such applications, we designed and implemented a *mobility service* within the SIENA publish/subscribe system. This service allows applications to receive notifications published while they are traveling to a new destination, and to restore the flow of notifications and their subscriptions when and where they reconnect to the SIENA network at their destination. This introduced some requirements, such as message persistence during the client motion, notification re-routing after the movement, and event downloading from the new client's destination.

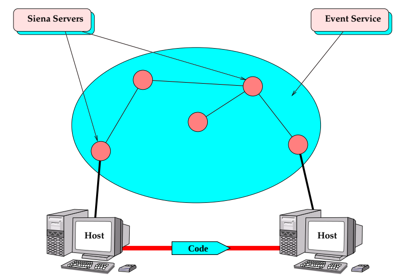

Figure 3: SIENA client moving form host to host.

As a basis for the mobility service, we implemented a persistent storage of notifications. This allows the client to relocate from host to host updating its information maintained by the event service. We extended the set of available operations in SIENA adding new actions specifically oriented to manage the mobility of the clients (see Table 3). The basic operations we developed are **moveOutMaster** and **moveInMaster**. **moveOutMaster** allows a client to declare its intention to move and causes the event service to suspend the delivery of notifications to that client. Of course, all events addressed to this client will be stored by the event service. When the client reaches its new location, it can use the **moveInMaster** operation to reconnect to the event service and retrieve all notification stored while the client was disconnected. Finally, we also added some operations that allow the client to discover other mobility-service-enabled servers available throughout the network [2].

4 Conclusions

In this work we have described our attempt at performance evaluations of a distributed application deployed over a wireless network. The application is characterized by the interaction of multiple clients residing at the periphery of the network, as well as by the need to deploy elements of the application deep into the network.

To our disappointment, we were not able to find tools capable of supporting a full evaluation. We were

error_probability $= 10^{-3}$						
	DELAY_ITERATE		FORWARD		DROP	
	notif.	*IP packets*	*notif.*	*IP packets*	*notif.*	*IP packets*
Keep Alive	79	875	17	350	8	285
TCP	100	1173	64	1205	62	1571
UDP	79	82	73	78	66	97
error_probability $= 10^{-4}$						
	DELAY_ITERATE		FORWARD		DROP	
	notif.	*IP packets*	*notif.*	*IP packets*	*notif.*	*IP packets*
Keep Alive	82	855	72	458	70	443
TCP	100	1153	100	1156	99	1152
UDP	100	106	84	95	76	91

Table 1: SIENA Behavior in the Wireless GPRS Network.

moveOutMaster()
moveOutMaster(boolean *QoS*)

moveInMaster()
moveInMaster(PacketReceiver *pr*)
moveInMaster(string *uri*)
moveInMaster(string *uri*, boolean *QoS*)
moveInMaster(PacketReceiver *pr*, string *uri*)
moveInMaster(PacketReceiver *pr*, string *uri*, boolean *QoS*)

addMobileDispatcherFinder(Notifiable *rec*)
removeMobileDispatcherFinder(Notifiable *rec*)

Table 2: Interface SIENA Mobility Support.

limited to the narrow evaluation of a single client interacting across the network with a single server. Nevertheless, our experience should not be taken as a criticism of Seawind, the tool that we decided to use for our evaluation. In fact, we found Seawind to be a reasonable and useful tool for its purpose.

Clearly, a need exists for a different kind of tool for wireless-network performance evaluation. Before embarking on the development of such a tool ourselves, we first plan to study the capabilities of NS-2 and its GPRS module, which hold some promise for modeling and evaluating services deployed deeply into a wireless network. We might in fact be able to extend them to also allow modeling and evaluation of client interaction over the network.

As we explained Section 3, our intent was to deploy mobility support in the SIENA publish/subscribe middleware. The API extension we presented in Table 3 provides specific services for mobile clients. However we do not consider it to be a definitive solution, but rather as a basis for future work. In fact the new operations allow us to perform additional case studies. One important aspect that we would like to study is the level of reliability provided by the new services. In particular, we would like to quantify the probability of losing or duplicating messages, or of changing their order [9]. As another further development, we would like to perform additional tests that we believe are very important due the probabilistic nature of the errors affecting the system.

We would also like to experiment with mobile agents using the SIENA mobility support in order to study the impact of the publish/subscribe architecture in this context. We would like to combine the mobility support we deployed in SIENA with *host mobility*, for example using an ad-hoc network [7]. In fact, since an ad-hoc network is completely composed by mobile hosts, its topology (and therefore the relations among the masters that composed it) changes quickly over time. In a situation like this, we imagine a SIENA server running in every mobile host and using the mobile features to manage its relation to the other components (usually referred to by the term *context management* [10]). It would be useful to validate our solution in this scenario and possibly study alternative solutions.

5 Acknowledgements

The author would like to thank Antonio Carzaniga, Paola Inverardi, and Alexander L. Wolf for their contributions to the work described in this paper.

This work was supported in part by a Research Exchange grant from the USENIX Association, and by the Defense Advanced Research Projects Agency, Air Force Research Laboratory, Space and Naval Warfare System Center, and Army Research Office under agreement numbers F30602-01-1-0503, F30602-00-2-0608, N66001-00-1-8945, and DAAD19-01-1-0484. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory, Space and Naval Warfare System Center, Army Research Office, or the U.S. Government.

References

[1] G. Brasche and B. Walke. Concept, services and protocols for the new GSM Phase 2+ General Packet Radio Service. Technical report, IEEE Communications Magazine, 1997.

[2] M. Caporuscio. CO.M.E.T.A. - mobility support in the SIENA publish/subscribe middleware. Master's thesis, Università degli Studi dell'Aquila - Dipartimento di Informatica, L'Aquila - Italy, March 2002.

[3] M. Caporuscio, A. Carzaniga, and A. L. Wolf. An Experience in Evaluating Publish/Subscribe Services in a Wireless Network. Submitted for Publication.

[4] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service. In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 219–227, Portland, OR, July 2000.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and Evaluation of a Wide-Area Event Notification Service. *ACM Transactions on Computer Systems*, 19(3):332–383, Aug. 2001.

[6] E. Durocher and I. Filotti. Native Code Migration over a Heterogeneous Network - An Efficient Approach to Process Migration.

[7] Internet Engineering Task Force. Mobile Ad-Hoc Networks (MANET) WG Charter. www.ietf.org/html.charters/manet-charter.html.

[8] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, and K. Raatikainen. *Seawind v3.0 User Manual.* University of Helsinki, Finland, September 2001.

[9] L. Lamport. Time, clocks, and the ordering of events in a distributed system, 1978.

[10] A. L. Murphy, G.-C. Roman, and G. P. Picco. Coordination and Mobility. In A. Omicini and F. Zambonelli and M. Klusch and R. Tolksdorf, editor, *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 254–273. Springer, 2000.