An Agent-Based Architecture for Service Discovery and Negotiation in Wireless Networks

Erich Bircher and Torsten Braun
University of Bern, Neubrückstrasse 10, 3012 Bern, Switzerland
Email: braun@iam.unibe.ch

Abstract

This paper proposes a market-place based architecture, where users can detect wireless network services, negotiate with the identified service providers about price and service features, select the best service, and finally configure their devices according to the selected service. The architecture automates these different steps by the use of agents that represent the involved entities such as user, service provider and marketplace operator. The architecture has been implemented using FIPA-OS. Performance measurements show that procedures such as service discovery and service negotiation can be performed far below one second despite the significant overhead introduced by the FIPA-OS platform.

1 Introduction

Wireless local area networks (WLANs) became recently attractive for telecommunications operators to offer public wireless communication services. Despite the relatively high speeds in WLANs network resources are still scarce and are usually not offered for free or by flat rates. Such resources must be reserved and charged dependent on the resource usage. This requires some kind of a contract (also called service level agreement, SLA) between customer and service provider. On the other hand, the customer is interested to compare service offerings and select services with the best performance / price ratio. We propose to implement an environment based on agents and marketplaces where agents representing wireless service customers can detect and meet agents representing wireless network service providers, negotiate with them about the offered services, and reserve the resources for the agreed price. Marketplaces are well suited for realizing rendezvous places where services providers and customers can meet each other. Agents allow flexible negotiations among the involved entities and flexible configuration of end system software. After discussing related work in this area in Section 2, we will present our system architecture and implementation design in Section 3. This architecture is based on marketplaces that are used by service providers to register their wireless services in a certain geographical region and by users to discover appropriate service offerings. The architecture makes use of agents that communicate together behalf on the entities they represent such as user, marketplace and service providers. Section 4 discusses the performance results obtained with a FIPA-OS based implementation on Linux computers. Finally, Section 5 concludes the paper.

2 Related Work

The Foundation of Intelligent Physical Agents (FIPA) [1] is an organization for defining standards for multi agent systems. The key focus of FIPA is to specify communication and inter-operability between agents in heterogeneous environments.

In FIPA every agent is located on a platform. The different platforms are then linked together. A platform consists of three main parts: The agent management system for the agent life cycle management (management of platform, starting and deleting of agents, access control etc.), the directory facilitator, which provides yellow pages services, and an agent communication channel, which enables agents to communicate with each other. Each new agent has to register at the agent management system and at the directory facilitator. It can get information about other agents from the directory facilitator and can then contact the agents over the agent communication channel. These other agents can stay on the same platform or on another one, as long as the other platform is indirectly or directly linked together with this platform. The main part of FIPA is the definition of the communication between agents. Two agents are communicating with each other using a set of pre-defined protocols.

FIPA-OS [2] is an open source implementation of the FIPA standard. It is a component-based toolkit implemented in pure Java and supports most of the FIPA experimental specifications currently under development. Recently, a small version of FIPA-OS aimed at PDAs and smart mobile phones has been developed within the IST project Crumpet [3]. The IST Shuffle project uses an agent-based approach to control resources in UMTS networks. The project aims to create a novel architecture for efficient, scalable and robust real-time control of third generation mobile systems in the context of realistic business models of network providers, service providers and customers. This goal shall be reached with intelligent software agents complying to the FIPA standard.

3 System Architecture

3.1 Overall Architecture

The goal of this work is to realize a marketplace for temporary Internet access services via mobile devices. To realize a market it needs a buyer, a seller and a marketplace entity. These entities run on different computers and are connected over the Internet. The seller entity, representing the user, should support the user in service discovery and negotiation and manage the network connectivity of the portable device on which it runs. This means that it should change automatically to the access point where it has bought access time from the seller (representing an Internet service provider, ISP). The design should be adaptable to different connection technologies and to more complex and dynamic situations. This has to be considered especially for the software design. Based on all these requirements, we developed an overall system architecture that can be decomposed into three layers:

- The first layer is the network layer. It consists of the physical networks and computers such as the home network of the user or the hot spot location, where the user may go to and connect his portable device to the Internet, e.g. a wireless LAN hot spot. At such a hot spot there might be two ISPs offering wireless network access. The ISPs must have the wireless transmission environment and systems for IP address management such as DHCP servers. The ISP might also use extended services set IDs (ESSIDs), wired equivalent privacy (WEP), or password based schemes for access control. Another important part of the hot spot is the reception area. This is a public access point where users without a contract can connect to and negotiate a contract with one of the ISPs in order to connect their access points. This public access point should have some kind of filtering mechanisms such as a firewall to prevent mis-use.

- The second layer consists of software supporting mobility and network connectivity. Unfortunately, most of today's operating systems do not support mobility sufficiently yet. To allow seamless handover between different access points including IP address change and to be able to connect securely to the home network, the Secure Mobile IP (SecMIP) software developed in [4] has been chosen in our work. SecMIP is a combination of Mobile IP and IP Security. It allows a mobile client to roam freely in the Internet and maintain a Mobile IP tunnel to the home agent of its virtual private network. The Mobile IP tunnel is protected by an IP Security tunnel between the mobile node and the firewall of the home organization (cf. Figure 1). SecMIP has been implemented on various operating system platforms.

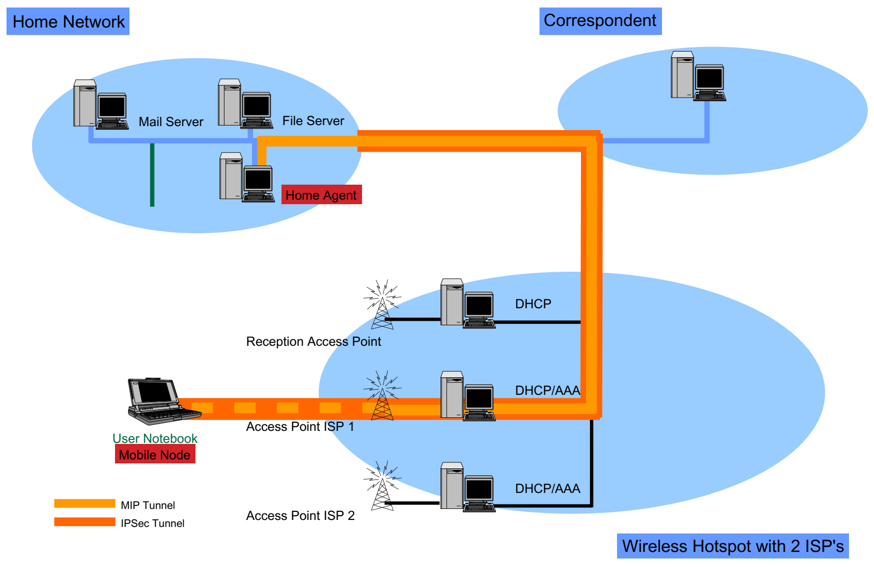


Figure 1: Secure Mobile IP

- The third layer is the layer mainly addressed by the work presented in this paper. The software supports the user in getting connectivity during times he is out of office. It contacts ISPs, negotiates contracts between the users and the service providers, and establishes network connections. This layer needs to have some kind of intelligence to perform these tasks. To design and implement this software we make use of agent technology. FIPA has been chosen as a basis because intelligent static agents are more appropriate for achieving the goals of this work than mobile agents. The chosen platform is FIPA-OS, which is required to run on each actor in the target scenario, i.e. at the user, the marketplace provider and the wireless Internet Service Providers (ISPs). FIPA-OS is open source and supports various operating systems. Since FIPA-OS is completely written in Java, Java Version 1.3 has been used as the programming language. This also requires a Java virtual machine running on each computer. The main entities of this layer are the user agent, located on the portable device of the user, the marketplace agent and the ISP agents, both likely installed at the hot spot location. Figure 2 shows a typical scenario where the user is in his home environment. He can use the services in the home environment such as email and file server access and can communicate with a correspondent node over the Internet. If the user plans to go to a place out of office, where he intends to connect to his home network, he can tell his user agent about his planned travel. The user agent is installed together with a FIPA-OS platform on the users portable device. The user agent contacts the corresponding marketplace agent, which returns a list of ISP agents. The user agent then negotiates a contract with one of these ISP agents. It gets the configuration data from the selected ISP after the negotiating a contract with the ISP agent. With this configuration data, it can later establish a network connection with the portable device at the desired location. If the users visits a hot spot without having a contract with an ISP, the user agent can contact an ISP at this hot spot over the reception access point in order to get a contract and the necessary network configuration data.

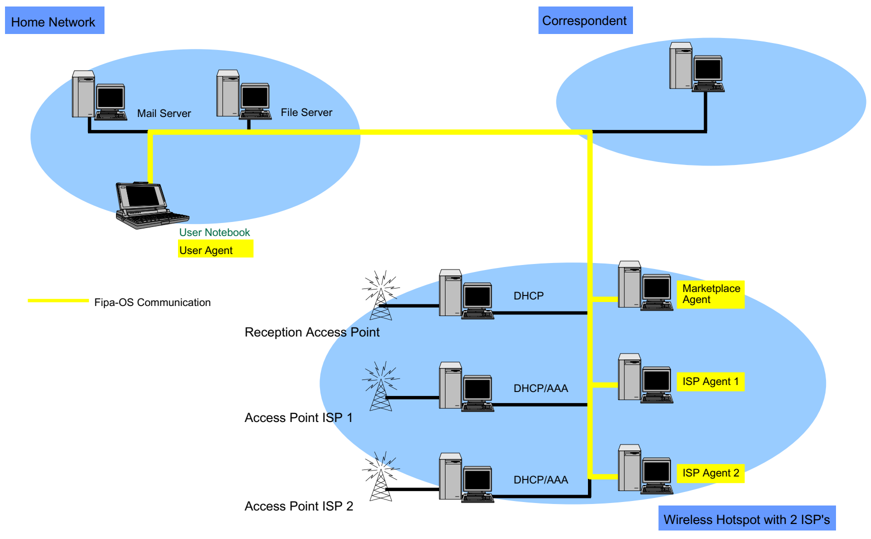


Figure 2: Agents in the target scenario

3.2 Agent-Based Marketplace Design

For a simple market, only a seller and a buyer are needed. However, seller and buyer have to find each other. Marketplaces are places where seller and buyer can meet each other in order to negotiate and sell / buy services. We propose three entities (agents) for the buyer, the seller, and the marketplace.

The buyer entity represents the user in our case. It is therefore called the user agent. Its main task is to buy the desired services on behalf of the user. Additional tasks are communication with the user, negotiation with the ISPs and the appropriate configuration of the user's portable device. Therefore, the user agent has been split into three entities: the travel assistant (TA), the negotiation agent (NA) and the configuration tool (CT). The main motivation behind this design choice is that the negotiation agent as a FIPA agent needs a running FIPA-OS platform. However, for many tasks of the user agent, the negotiation agent is not really needed. In this case, the FIPA-OS platform need not to be started but only the travel assistant. Moreover, the system should be extensible by new agents.

The travel assistant is the interface to the user and the controlling entity of the user agent. It supports several functions related to user travelling. In particular, it looks for connectivity during the time, when the user is out of office. The travel assistant communicates with the user over a graphical user interface (GUI) and delegates tasks to the negotiation agent and the configuration tool. It starts and stops the FIPA-OS Platform and the negotiation agent. The negotiation agent is an intelligent software agent that contacts the marketplace agents to get information about available locations and ISPs offering services there. It also negotiates a short-time service level agreement with eligible ISP agents. The configuration tool gets from the travel assistant all the negotiated contract information such as start time and duration of the service, the wireless network technology to be used and configuration data such as IP addresses, ESSIDs, WEP keys or user names and passwords. Later, it tries to establish the desired connections in time.

The ISP agent represents the seller, i.e. an ISP at a certain hot spot in our case. The main task of the ISP agent is to sell services on behalf of the ISP. It indicates its presence by subscribing to the according marketplace agent. If a user needs Internet access time at the hot spot, it performs SLA negotiations with the agent representing the interested user. In order to do this, it needs to have information about the ISP, the access point, available resources, and pricing schemes.

The role of the marketplace agent is to bring seller (ISP agent) and buyer (user agent) together. There are some different approaches how to design the marketplace entity. One approach is that the marketplace agent helps buyers (sellers) to find eligible sellers (buyers) and then let seller and buyer perform the deal by themselves. The approach is oriented to yellow pages and requires seller and buyer to be intelligent. The marketplace agent has only intermediation functionality. An other approach would be that the marketplace not only brings the buyer and seller together, but also performs the negotiation between them. This approach requires less intelligent sellers and bidders. They can just indicate their offers and bids to the marketplace. In cases, where the market should be controlled and where high trust is needed, such a centralized system would make much sense. The advantage of the decentralized approach is better flexibility and independence. If a seller wants to change his strategy it is far easier to change the corresponding agent instead of trying to indicate a new strategy to a marketplace agent. Thus, we have chosen the first approach for the realization of the marketplace. Figure 3 summarizes the agents and their relation between them.

A marketplace could be a huge directory covering all sellers in a large geographical region. On the other side, the markets can be organized in smaller units. We have chosen the latter approach: There is a marketplace for every access hot spot. The main motivation for that decision was that the marketplace agent can be located physically at the reception area. This allows users without contracts to contact this local marketplace agent and the local ISP agents more easily. In addition, this distributed approach is much more scalable.

Another issue is the information exchange between the user agent and the marketplace agent that has to provide some information about the sellers. We aim to have a flexible marketplace agent that is able to handle different kinds of queries. Some user agents may only need the addresses of ISP agents that sell a certain product, others might desire more information and filter eligible ISPs by themselves. This approach allows the user agent to give some information about sellers and recommendations to the user. In particular, the marketplace agent gives some information about the registered ISPs such as the supported technology or acceptable payment methods to the user agents. The user agent can then decide about eligible ISPs or inform the user about indicated options that were not supported by the user. The main task of the marketplace agent representing a geographical hot spot location for Internet access is to inform interested parties about all the ISP offering services within its area. This requires that all these ISPs have to register with the marketplace agent. The marketplace gives then the information out to interested negotiation agents upon request.

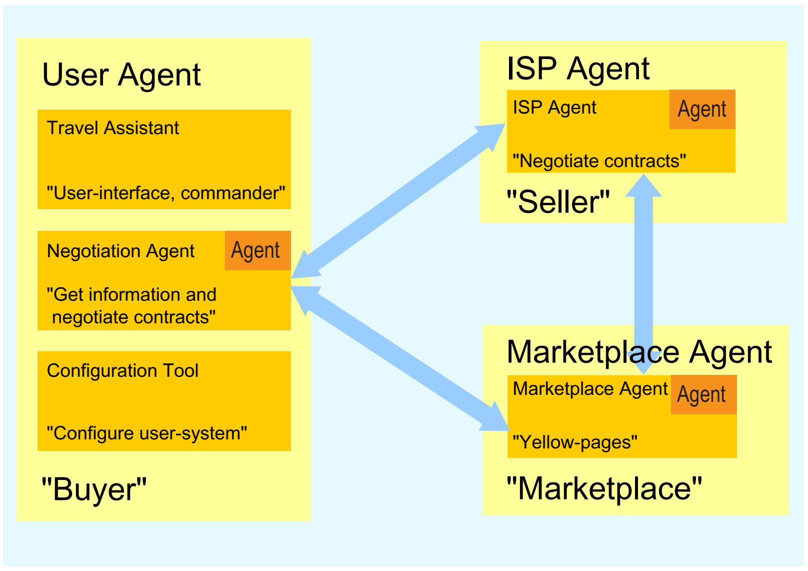


Figure 3: Agent realizing a marketplace for wireless Internet services

3.3 Agent Interactions

In this subsection we describe in more detail how the agents interact with each other in order to fulfill the desired task. For agents it is important that they can understand each other. The usage of standard protocols, in particular FIPA protocols, is a significant step in order to solve this problem. The agents have to know which protocols to use in order to start a conversation (dialogue) with an other agent. Moreover, the agents have to understand the content they send to each other. This requires to clearly define the content to be exchanged. All participants in a conversation have to know the Java objects included in the messages. In our case, three objects are exchanged between the agents and the objects need to be serialized for transmission.

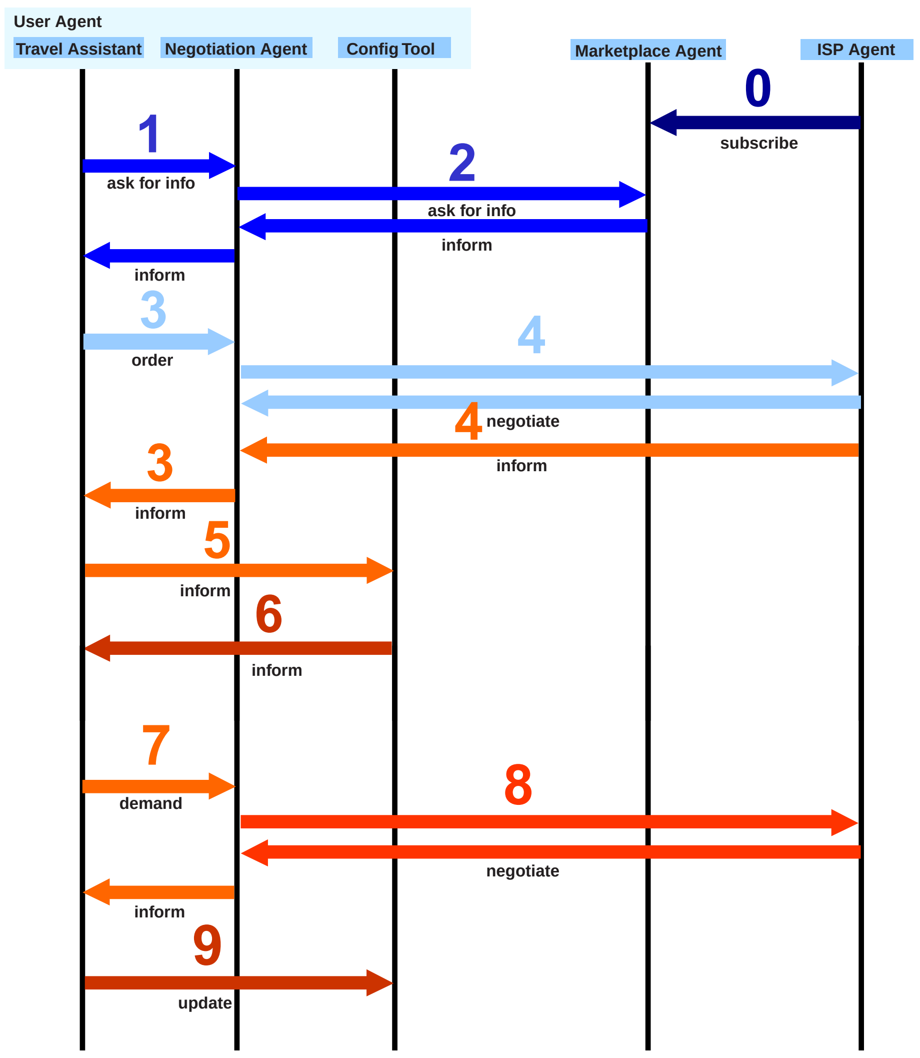


Figure 4: Agent Interaction

Figure 4 shows the interactions among the various agents. First, an ISP agent has to register / subscribe at a marketplace agent delivering the ISP name, the access point technology and acceptable payment schemes (step 0). This step is supported by the FIPA-Subscribe-Protocol. Then, the travel assistant begins its operation by asking the negotiation agent to provide information about available marketplaces (1). This is done by internal communication based on method invocation between objects. The negotiation agent contacts the marketplace agent using the FIPA-Request-Protocol (2) and delivers the information about them (e.g. location and registered ISPs) back to the travel assistant. After that, the user finishes the interaction with the travel assistant and gives a list with all desired connections and SLA information including target Quality-of-Service (QoS) values to the negotiation agent (3). The negotiation agent then contacts the potential ISPs and performs negotiations based on the SLA parameters proposed by the user (4). The protocol depends on the type of negotiation and can be a FIPA-Contract-Net-Protocol (Figure 5) or a FIPA-EnglishAuction-Protocol. The negotiation agent may now accept an offer from an ISP and receive configuration information after transmitting accounting information to the ISP agent. The negotiated SLA and the necessary configuration data are delivered to the travel assistant via the negotiation agent. In step 5, the travel assistant initiates the configuration of network and protocol parameters in order to be able to use the selected service to the configuration tool, which returns an acknowledgement (6). Steps 7-9 are performed in case of SLA modifications and similar to steps 3-5. For the negotiation in step 8 between the negotiation agent and the ISP agent the FIPA-Request-Protocol is used.

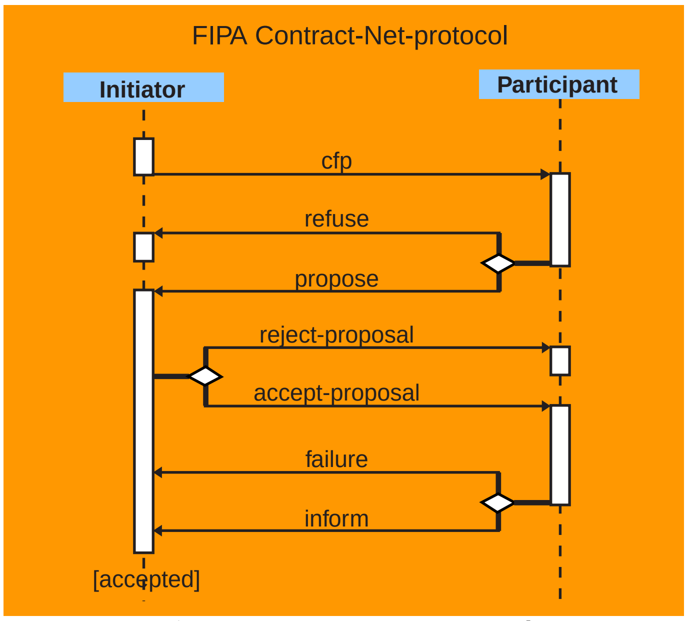


Figure 5: FIPA-Contract-Net-Protocol

3.4 Market Issues

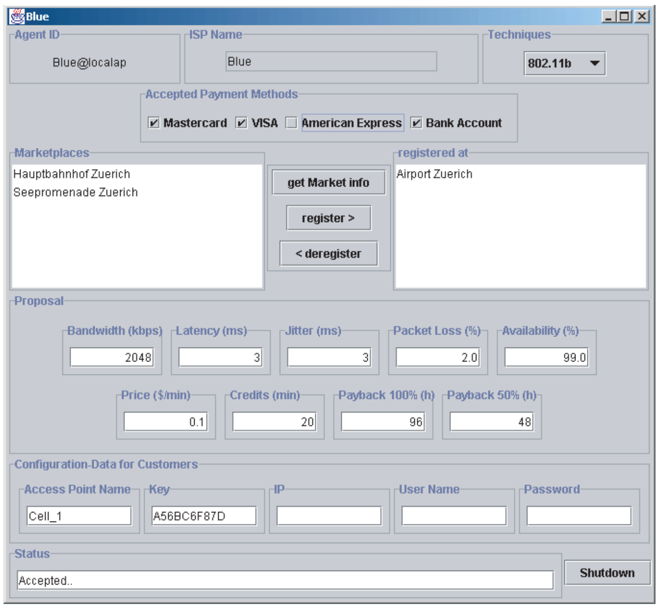
For price determination the following alternatives are possible: reverse-auctions, negotiations and fixed price. Auctions do not make much sense, because the delay for fixing the price would be too large. For stock markets, the offered service might not be homogeneous enough, since the service offerings need to be tailored to individual customer needs. Until now, we have implemented a fixed price approach, where the ISP agents return a fixed price dependent on bandwidth requirement and network connection time. At the user agent, a decision function according to [5] has been implemented in order to allow users to weight QoS parameters such as delay, bandwidth, packet loss etc. differently. The user can also specify preferences concerning price and contractual issues. For each parameter, the difference between the actual and the ideal value is calculated and weighted. The offer with the lowest sum of weighted differences is preferred.

3.5 Security Issues

Agent-based negotiations implicitly introduce security risks. The negotiating entities should not only encrypt the exchanged information that might include sensitive data such as credit card numbers, but also authenticate each other. Confidential information exchange could be achieved using the Secure Socket Layer (SSL) for Java RMI. Authentication mechanisms could rely on public key infrastructures (PKI). The FIPA-OS version we used does not include any PKI support. However, this topic is currently being addressed by the research community [6].

3.6 Graphical User Interfaces

The different agents need also to provide a graphical user interface (GUI) in order to exchange information with a human who controls and configures the agent. The GUI of the ISP agent (Figure 6) allows to retrieve information from places and to register / deregister ISP agents at certain marketplaces. Moreover, it allows to compose service offerings including price information and network configuration data. The GUI for the marketplace agent displays the information about the registered ISP agents.


Blue

Agent ID: Blue@localap
ISP Name: Blue
Techniques: 802.11b

Accepted Payment Methods: [✓ Mastercard] [✓ VISA] American Express [✓ Bank Account]

Marketplaces:
Hauptbahnhof Zuerich
Seepromenade Zuerich

registered at:
Airport Zuerich

Proposal

Bandwidth (kbps)	Latency (ms)	Jitter (ms)	Packet Loss (%)	Availability (%)
2048	3	3	2.0	99.0

Price ($/min)	Credits (min)	Payback 100% (h)	Payback 50% (h)
0.1	20	96	48

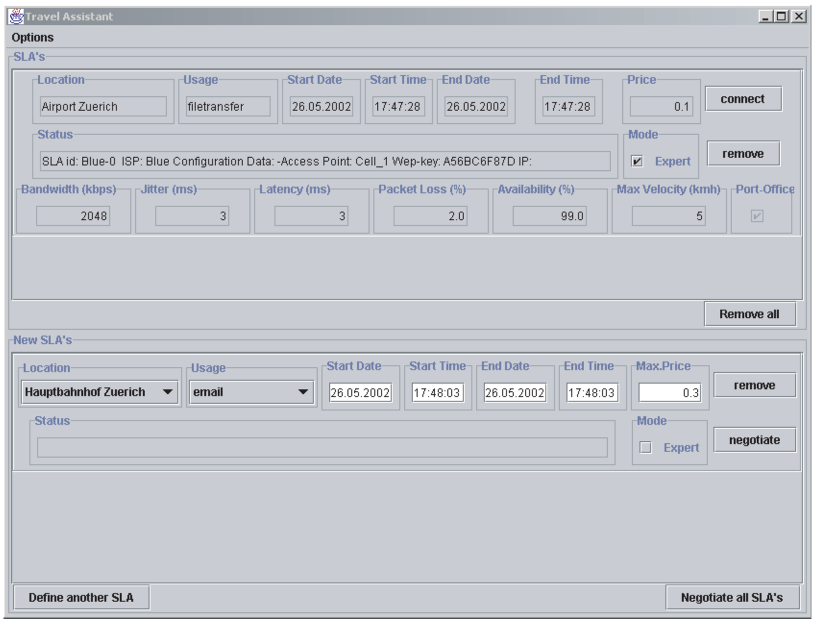
Configuration-Data for Customers

Access Point Name	Key	IP	User Name	Password
Cell_1	A56BC6F87D			

Status: Accepted..

Figure 6: ISP Agent GUI

The user agent GUI (Figure 7) allows interaction with the user. First, it provides the available host spot locations to the user. Moreover, the user can define preference profiles for different applications, e.g. file transfer, email, video conferencing, web browsing, in order to define bandwidth and QoS requirements for each type of application. The default values for the selected services are used for SLA negotiation unless the user overwrites these values. Additional parameters such as start and stop time of the network service have to be defined prior to SLA negotiation. The user has also to define the network interfaces supported by its computer and the payment schemes he is willing / able to use.



Travel Assistant

Options

SLA's

Location	Usage	Start Date	Start Time	End Date	End Time	Price
Airport Zuerich	filetransfer	26.05.2002	17:47:28	26.05.2002	17:47:28	0.1

Status: SLA id: Blue-0 ISP: Blue Configuration Data: -Access Point: Cell_1 Wep-key: A56BC6F87D IP:

Mode: [✓ Expert]

Bandwidth (kbps)	Jitter (ms)	Latency (ms)	Packet Loss (%)	Availability (%)	Max Velocity (kmh)	Port-Office
2048	3	3	2.0	99.0	5	✓

New SLA's

Location	Usage	Start Date	Start Time	End Date	End Time	Max.Price
Hauptbahnhof Zuerich	email	26.05.2002	17:48:03	26.05.2002	17:48:03	0.3

Status:

Mode: Expert

Figure 7: User Agent GUI

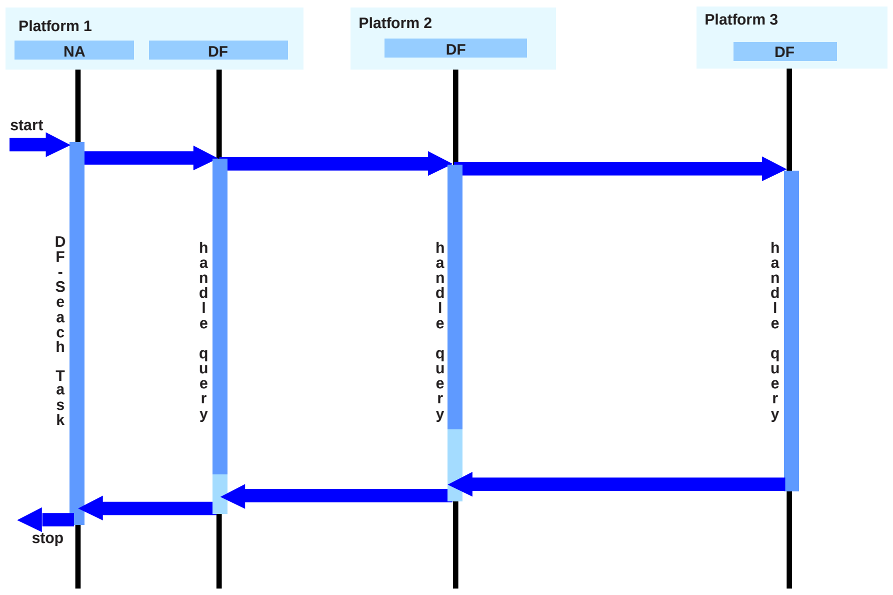


Figure 8: Directory Facilitator Search across 3 Platforms

4 Performance Evaluation

For the performance evaluation Linux based agent implementations and IEEE 802.11 WLANs have been used.

4.1 Directory Facilitator

In a first test, we evaluated the performance for finding other registered agents in a cluster of computers. This is supported by the directory facilitator functionality which allows to search for specific agents on a platform and to forward such requests across different platforms that are linked together. The directory facilitator search is initiated by the negotiation agent and forwarded across a chain of 1-3 platforms (Figure 8). Platforms 1 and 2 were running on 333 MHz computers, while platform 3 was running on a 525 MHz computer. For a single platform, the directory facilitator search took approximately 900 ms in average, while the time increased to 3 s when introducing a second platform. However, introducing a third platform did not increase the search time. Obviously, platform 2 forwards the requests further and the search operations are performed in parallel at platforms 2 and 3.

4.2 Negotiation between Negotiation Agent and ISP Agent

While the directory facilitator search is only performed at system initialization, other conversations among the agents are more time-critical. First, we looked at one of the most complex conversations between the negotiation and the ISP agent: the negotiation that is based on the FIPA-Contract-Net-Protocol. When negotiation agent and ISP agent are running on a single computer (333 MHz), the conversation takes 605 ms in average. Distributing these two agents to two computers (333 MHz) increases the negotiation time slightly to 625 ms. Running the negotiation agent on a 333 MHz computer and the ISP agent on a 525 MHz computer reduces the negotiation time to 496 ms in average. These results show that ISP agent processing is rather costly and distributing agents to different computers can result in accelerating the negotiations.

The next series of tests have been performed with two ISP agents. The negotiation agent and one ISP agent were running on a 333 MHz computer each, one ISP agent was running on the 525 MHz computer. The negotiation time increased to 926 ms in average.

Finally, we tested the maximum capacity of an ISP agent by permanently issuing negotiations from a negotiation agent to an ISP agent. If negotiation agent and ISP agent were running on the same computer (333 MHz), 3.1 negotiations per second could be achieved. When distributing negotiation agent and ISP agent to two computers (both 333 MHz), the number increased to 3.8 negotiations per second. Using five negotiation agents communication to a single ISP agent increased the number even to 5.4 negotiations per second.

5 Conclusions

The paper described the realization of a marketplace for trading wireless network services based on agent technology. The FIPA technology has proven to be a viable candidate for the design and implementation of a marketplace where users and service providers can meet. In particular, several components such as FIPA protocols have been very helpful. The preliminary performance results indicate that at least in scenarios with a few users in a hot spot area, the marketplace implementation achieves reasonable performance even on legacy computer systems. The concept also has significant potential for including more intelligence into the agents. Nevertheless, the performance of platforms need to be improved significantly and more systems such as PDAs and mobile phones need to be supported by such agent platforms.

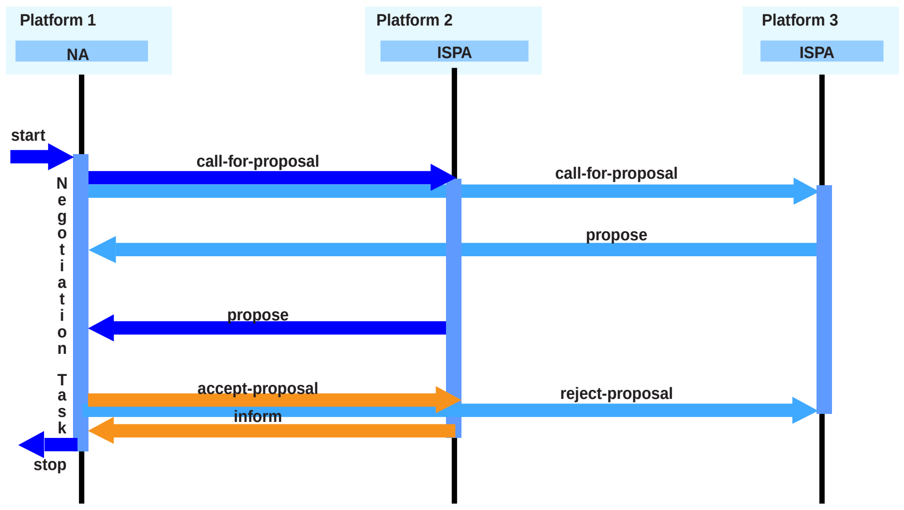


Figure 9: Negotiations between 2 ISPs on 3 Platforms

Acknowledgements

The authors would like to thank Swisscom AG for providing infrastructure support to implement the architecture presented in this paper.

References

1. P. D. O'Brien, R.C. Nicol: FIPA-towards a Standard for Software Agents, BT Technology Journal, Vol.16, No.3, July 1998
2. S. Poslad, P. Buckle, and R. Hadingham. The FIPA-OS agent platform: Open Source for Open Standards, Firth International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents, pp. 355-368, 2000.
3. S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle and A. Zipf: CRUMPET: Creation of User-friendly Mobile Services Personalised for Tourism, Second International Conference on 3G Mobile Communication Technologies, March 26-29, 2001, London
4. Marc Danzeisen and Torsten Braun: Secure Mobile IP Communication, Workshop on Wireless Local Networks at the 26th Annual IEEE Conference on Local Computer Networks (LCN'2001), Tampa, USA, November, 15-16, 2001
5. J. Morris, P. Ree, P. Maes: Sardine: Dynamic Seller Strategies in an Auction Marketplace, ACM Electronic Commerce 2000, October 17-20, 2000, Minneapolis, Minnesota
6. Yuh-Jong Hu: Some Thoughts on Agent Trust and Delegation, Fifth International Conference on Autonomous Agents, May 2001