

Adaptive Protocols for Agent Migration

Ichiro Satoh*

Department of Information Sciences Ochanomizu University /
Japan Science and Technology Corporation
2-1-1 Otsuka Bunkyo-ku Tokyo 112-8610, Japan
E-mail: ichiro@is.ocha.ac.jp

Abstract

A framework for building adaptive network protocols for agent migration over a network is presented. A key idea of the framework is to introduce mobile agents as first-class objects. That is, the framework allows network protocols for agent migration to be naturally implemented within mobile agents and then dynamically deployed at network nodes by migrating the agents that carry the protocols. A prototype implementation was built on a hierarchical mobile agent system, and several practical protocols for agent migration were designed and implemented. These protocols can have significant contributions to active network technology as well as mobile agent technology.

1 Introduction

Over the past several years, there has been a lot of work in the area of mobile agents. Mobile agents have been used in the development of various networked applications. These applications often need to dynamically customize network processing for migrating agents over a network. For example, a mobile agent for electronic commerce needs to be transformed into an encrypted bit stream before transferring itself over a network. In mobile computing settings, network protocols for agent migration should tolerate network disconnection. However, existing mobile agent systems cannot dynamically adapt their network processing to the requirements of visiting agents and changes in their surrounding environments. This is because their network protocols are statically embedded inside their runtime systems. Therefore, it is nearly impossible to extend and configure the protocols dynamically.

This paper addresses the dynamic customization of network processing for agent migration, rather than for data transmission. We describe a new framework for dynam-

cally deploying and changing network protocols for agent migration. Our framework is characterized by two key ideas. The first is to apply active network technology to a network infrastructure for mobile agents. The second is to construct network protocols for agent migration within the agents themselves. That is, our mobile agent-based protocols can transmit mobile agents as first-class objects to their destinations. Also, the dynamic deployment of the mobile agent-based protocols can be naturally and easily performed by the migration of the agents that support these protocols. Our framework is built on a hierarchical mobile agent system called *MobileSpaces* [6] that can hierarchically organize multiple mobile agents and introduce mobile agents as service providers for other mobile agents.

This paper is organized as follows. Section 2 surveys related work and Section 3 explains our approach to customizable network processing for agent migration. Section 4 presents several mobile agent-based protocols running on the system. Section 5 makes some concluding remarks.

2 Background

Mobile agents are autonomous programs, which can travel between locations, i.e., agent platforms or runtime systems running on different computers. Many mobile agent systems have been released over the last few years. A mobile agent often must visit several nodes to perform its task and thus is required to make an application-specific and network-dependent itinerary. However, it is difficult to determine the itinerary at the time the agent is designed or instantiated because the network topology cannot always be known. Aglets [5] overcomes this problem by using the notion of an itinerary pattern to shift the responsibility for navigation from an application-specific agent to a framework library. ADK [3] separates the travel itinerary of an agent from its behaviors by building a mobile agent from a set of component categories: navigational components responsible for a travel itinerary, performer components responsible for executing one or more management tasks on each node,

*Currently with National Institute of Informatics.

and reporter components that determine how the results to be collected and reported back. However, both the itinerary pattern in Aglets and the navigational components in ADK must to be statically and manually embedded in their mobile agents. Consequently, their agents cannot dynamically change their itineraries and thus cannot travel beyond familiar networks. Therefore, we need an approach for efficiently navigating mobile agents among several nodes, without any limitation on the reusability of application-specific agents. Our framework is similar to active network technologies [8] and also makes several contributions to the technologies. Mobile agent technology has been used for prototype implementations of active networks. For example there have been many attempts to apply mobile agent technology to the development of active networks [1, 4], since mobile agents can be regarded as a special case of mobile code technology, which is the basis of most existing active network technologies. In contrast, we apply active network technology to mobile agent technology.

3 Architecture

The framework presented in this paper provides a self-configuring infrastructure for mobile agents. It can deploy and configure network protocols for agent migration according to the requirements of visiting agents and to changes in the network environment.

Our framework is built on the MobileSpaces system [6].¹ The system is characterized by two novel concepts: agent hierarchy and inter-agent migration. The former means that one mobile agent can be contained within another mobile agent. That is, mobile agents are organized in a tree structure. The latter means that each mobile agent can migrate to other mobile agents as a whole, with all its inner agents. Each agent can freely move into any agent in the same agent hierarchy except into itself or its inner agents, as long as the destination agent accepts it. A container agent is responsible for automatically offering its own services and resources to its inner agents and can subordinate its inner agents. Therefore, an agent can directly instruct its inner agents to move to another location. Therefore, the system can treat mobile agents as first-class objects, in the sense that mobile agents can be passed to and returned from other mobile agents as values.

Therefore, our framework allows various operations for mobile agents, including network processing for agent migration, to be naturally constructed and performed by other mobile agents. That is, network protocols for agent migration are implemented by mobile agents. These mobile agent-based protocols can transmit other agents to the destinations of the agents in the most appropriate way. The pro-

¹Details of the MobileSpaces mobile agent system can be found in our previous paper [6].

ocols can be autonomously and dynamically deployed at nodes by migrating themselves. Therefore, our framework offers a self-configurable infrastructure of network processing for mobile agents. Furthermore, they can be constructed and reused through a single programmable abstraction for composition and refinement of mobile agents. As a result, our framework can greatly simplify the development of active networks.

4 Mobile Agent-Based Protocols for Agent Migration

Our hierarchical structure for mobile agents enables network protocols for agent migration to be organized hierarchically. Also, agent migration in a hierarchy is introduced as a basic mechanism for accessing services provided by the underlying layer. That is, A mobile agent can be designed to provide its service to its inner agents. When a mobile agent is preparing for a trip over a network, the agent migrates itself into an agent for providing appropriate network processing in the same agent hierarchy and then the agent automatically transfers the visiting agent or migrates itself to its destination, or delegates other agents in the same agent hierarchy. Since network protocols are performed by mobile agents, the protocols can be dynamically and autonomously deployed at nodes by migrating the corresponding agents to the nodes. Hereafter, we present three basic protocols for agent migration. Since these protocols are given as abstract classes in the Java language, we can easily define further application-specific protocols by extending the basic protocols.

4.1 Point-To-Point Channels for Agent Migration

Agent migration between different computers is performed by mobile agents, called *transmitter*, instead of the core system. Transmitter agents are responsible for establishing point-to-point channels for agent migration as link-layer channels and can automatically exchange their inner agents through their common communication protocol. Since they are mobile agents, they can be dynamically deployed at nodes and then automatically performed. After an agent arrives at a transmitter agent from the upper layer, the arriving agent indicates its final destination. The transmitter suspends the arriving agent (including its inner agents), then serializes its state and codes. It transforms the serialized agent into a form appropriate for its communication protocol if necessary. Next, it sends the serialized agent to a coexisting transmitter agent located at the destination. The transmitter agent at the destination receives the data and then reconstructs the agent (including its inner agents) and migrates it to the destination or specified agents for offering upper-layer protocols.

Since each runtime system can be equipped with more than one transmitter agent, upper-layer protocols can dynamically select a suitable agents in their requirements and migrate their inner agents to the selected transmitter agents. We have already implemented several transmitter agents based on data communication protocols such as TCP, UDP, and SMTP. We provide a virtual class in Java that can be specialized to create transmitter agents for various protocols.

4.2 Routing Mechanisms for Agent Migration

Application-specific mobile agents often need to travel to multiple nodes to perform their tasks. However, it is difficult to determine the itinerary at the time the agent is designed or instantiated. Therefore, we introduce two approaches for determining and managing the itinerary of agents. These approaches are built on transmitter agents running on nodes and correspond to kinds of application-specific routing protocols.

4.2.1 Navigator Agent Approach

The first approach offers a service provider, called *navigator*, for conveying inner agents over a network, as shown in Figure 1. Each navigator agent is a container of other agents and travels with them in accordance with a list of nodes statically or algorithmically determined, or dynamically based on the agent’s previous computations and the current environment. That is, a navigator agent can migrate to the next place as a whole with all its inner agents. Each navigator has a routing mechanism for managing a routing table consisting of nodes the navigator agent needs to visit. It maintains a list of nodes to be visited and provides methods to dynamically add and remove elements from this list. Whenever a navigator agent moves to a new place, the agent accesses a local SNMP agent in order to update its own routing table and then evaluates the table to determine what the next hop should be. The interaction between a navigator agent and its inner agents is based on event-based communication. Upon arrival at a place, the navigator propagates certain events to its inner agents, instructing them to do something during a given time period. After the events have been processed by the inner agents, the navigator continues with its itinerary.

4.2.2 Forwarder Agent Approach

The second approach is based on a service provider, called a forwarder agent, for redirecting moving agents to new destinations. Each forwarder agent is a mobile agent and is designed to stay at nodes and automatically transfer its inner agents to specified nodes through appropriate transmitter agents as shown in Figure 2. Consequently, a forwarder

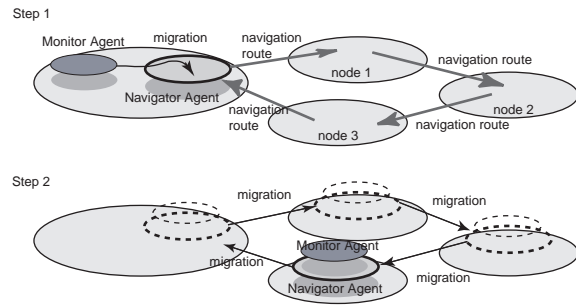


Figure 1. Navigator agent for traveling among nodes with its inner agent.

agent can be regarded as a programmable router for mobile agents.

The use of forwarder agents allows various routing schemes used in wired and wireless networks to be easily performed and evaluated. Such forwarder agents are dynamically deployed at nodes and coordinate with each other to redirect moving agents to their destinations. That is, when an agent requests a forwarder agent to migrate to its destination, the forwarder agent makes an effort to transfer the moving agent to the destination. However, if the destination is not reachable, it tries to transfer the moving agent to another forwarder agent running on an intermediate node as near to the destination as possible. Each forwarder agent will repeat the entire process in the same way until the agent arrives at the destination.

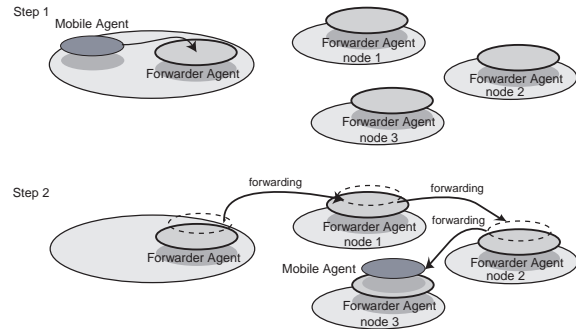


Figure 2. Routing agents for forwarding the next nodes.

4.3 Protocol Distribution

Given a dynamic network infrastructure, a mechanism is needed for propagating mobile agents for supporting protocols to where they are needed. The current implementation

of our framework provides the following three mechanisms: (1) mobile agent-based protocols autonomously migrate to nodes at which the protocols may be needed and remain at the nodes in a decentralized manner; (2) mobile agent-based protocols are passively deployed at nodes that may require them by using forwarder agents prior to using the protocols as distributors of protocols; and (3) moving agents can carry mobile agent-based protocols inside themselves and deploy the protocols at nodes that the agents traverse. This mechanism can improve performance in the expected common case of agent migration, i.e., a sequence of agents that follow the same path and require the same processing. All the mechanisms are managed by mobile agents, instead of by the runtime system. As a result, the deployment of transmitter agents needs to be performed by other transmitter agents. For practical reasons, our current implementation provides a built-in transmitter agent, which can deploy other agents to specified nodes through extended HTTP running on TCP/IP communication. It offers the bootstrapping capability needed to install other protocols.

4.4 Current Status

The MobileSpaces mobile agent system has been implemented in the Java language (JDK1.1 or later version). The core system is constructed independently of the underlying system and can run on any computer with a 1.1-compatible Java runtime. Also, we provided several useful libraries for constructing network protocols within mobile agents. Also, we have developed various mobile agent-based protocols, for example transmitter agents for TCP, UDP, and SMTP, forwarder agents, navigator agents for traveling among multiple active nodes, as presented in this paper. These protocols were written in Java and tested using the MobileSpaces systems.

Even though our implementation was not built for performance, we have conducted a basic experiment on agent migration. The cost of an agent migration in an agent hierarchy was measured to be 5 ms, including the cost to check whether the visiting agent is permitted to enter the destination agent or not. The cost of agent migration supported by transmitter agents allocated on two computers was measured to be 30 ms. A transmitter agent can communicate with another by using an application-level protocol for agent transmission whose mechanism is modeled on that of the HTTP protocol over TCP/IP communication. On the sender side, a transmitter agent serializes and transfers the codes and state of an agent (including its inner agents) to the transmitter on the receiver side and waits for an acknowledgment message. The second result is the sum of the marshaling, compression, opening TCP connection, transmission, acknowledgment, decompression, security and consistency verifications, and unmarshaling. The moving agent

is a simple navigator agent and consists of basic callback methods and contains two child agents. Its data size is about 3 Kbytes (zip-compressed).

5 Conclusion

In this paper we have shown how to apply active network technology to the construction of self-configuring infrastructure for mobile agents. Our active network framework focuses on the dynamic deployment and configuration of network protocols for agent migration. The framework uses the notions of agent hierarchy and inter-agent migration. Therefore, a mobile agent can be treated as a first-class object. Therefore, network protocols for agent migration can be naturally implemented as mobile agents. Also, the protocols can be dynamically deployed at network nodes by migrating the agents that offer the protocols. The framework and various mobile agent-based protocols are implemented with a Java-based mobile agent system, called MobileSpaces. We presented several mobile agent-based protocols, for example network environment-dependent protocols for establishing point-to-point channels for agent migration and application-specific routing protocols for navigating agents among multiple nodes.

Acknowledgments

We want to thank the anonymous reviewers for their helpful advices.

References

- [1] C. Baumer, and T. Magedanz, The Grasshopper Mobile Agent Platform Enabling Short-Term Active Broadband Intelligent Network Implementation, Proceedings of Internal Working Conference on Active Networks, pp.109–116, LNCS, Vol.1653, Springer, 1999.
- [2] A. Bieszczad, B. Pagurek, and T. White, Mobile Agents for Network Management. IEEE Communications Surveys, Vol. 1, No. 1, Fourth Quarter 1998.
- [3] T. Gschwind, M. Feridun, and S. Pleisch, ADK: Building Mobile Agents for Network and System Management from Resuable Components, Technical University of Vienna, TUV-1841-99-10, 1999.
- [4] A. Karmouch, Mobile Software Agents for Telecommunications, IEEE Communication Magazine, vol. 36 no. 7, 1998.
- [5] B. D. Lange and M. Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley, 1998.
- [6] I. Satoh, MobileSpaces: A Framework for Building Adaptive Distributed Applications Using a Hierarchical Mobile Agent System, Proceedings of International Conference on Distributed Computing Systems (ICDCS'2000), pp.161-168, IEEE Computer Society, April, 2000.
- [7] I. Satoh, MobiDoc: A Framework for Building Mobile Compound Documents from Hierarchical Mobile Agents, Proceedings of Symposium on Agent Systems and Applications / Symposium on Mobile Agents (ASA/MA'2000), LNCS, pp.113-125, Springer, 2000.
- [8] D. L. Tennenhouse et al., A Survey of Active Network Research, IEEE Communication Magazine, vol. 35, no. 1, 1997.