Chapter 2.21
A Neural Network–Based Mobile Architecture for Mobile Agents

Anand Kuppuswami
University of Western Sydney, Australia

ABSTRACT

Wide area network (WAN) offers advantages like providing myriad services available on globally diversified computers with reasonably simple process. The ability to dynamically create networks offers the processing powers of various processors at our command. With the advent of protocols like SOAP and Web services, the consumption of services are more organized. In spite of various advances in communication techniques, the consumption of services through mobile gadgets is still only at the research level. The major impedances in implementing such systems on a mobile network are the latency factor, abrupt disconnection in service, lower bandwidth and minimal processing power. The mobile agent's paradigm proves to be an effective solution to various issues raised. It has received serious attention in the last decade and several systems based on this paradigm have been proposed and built. All such systems have been designed for a static network, where the service providers and the requestors are connected to the server on a permanent basis. This chapter presents a new framework of managing the mobile environment and the participating nodes with active intelligent migration. The functioning of the mobile agents in such a scenario is also presented.

INTRODUCTION

Mobile devices foster a new set of applications that are receiving enormous support in the global electronic community. This is primarily because of the ability of mobile devices to connect and reconnect with each other dynamically. The ability to create a network of devices "on the fly" can be adjudged as a major advantage of mobile gadgets as compared with the previous computer networks. The ability of being able to communicate and, more importantly, process information irrespective of time and place is a very promising feature offered by mobile technologies, as also ratified by Gray, Kotz, Nog, Rus, and Cybenko (1996). With the advent of the Internet and, especially, in communication, almost all actions related to

an application have been reduced to a few mouse clicks. As is obvious in our day-to-day usage, the Internet is a vast repository of information and services. Even though both Internet and related mobile technologies are very promising, their usage by business is still full of hurdles. For example, accessing a simple HTML page in a mobile gadget requires various configurations in them. This can be attributed to low latency, physical obstruction, and network connectivity of the mobile gadgets. The mobile gadgets have low memory, processing capabilities, and are prone to sudden failure (Jipping, 2002). These factors unveil the problem of client server architecture (Gray et al., 2002b). Often, having a well-thought-out networking architecture is crucial to successful usage of this technology. For example, the huge amount of data that is required to be transported due to various multimedia applications to the client's site for further processing is not possible without a good architecture. This created the need for remote working where code could migrate to a different machine, execute at the new machine, and return with the result. The whole concept of working remotely started with the message passing and remote procedure calls (RPCs) provided by Java (Birrell & Nelson, 1984). But their application was limited, as the client could use only those services provided by the server. If the requested service is not present, then the client has to make intermediate or "Bridge" function class in order to get to the final result. This process results in wastage of resources and bandwidth. As an alternative approach, small subprograms could be written and passed on to the service provider to execute locally. Network Command Language (NCL) (Meandzija, 1986), remote evaluation (REV) (Stamos & Gifford, 1990), and SUPRA-RPC (Stoyenko, 1994) employ this idea in their architecture. All these architectures had one major concern: the code, once migrated to a different machine, cannot remigrate at the end of execution. The architecture also lacked in active coordination between different programs. The absence of inter-communication protocols leads to poor performance of the system.

The novel approach of transportable programs (Gray, 1995; Cybenko, Gray, Wu, & Khrabrov, 1994) offers a promising solution for various issues raised. Transportable agents or mobile agents, as they are called now, are autonomous programs that can migrate from one machine to another machine in the network. By migrating to the machine having the resource, the agents have the advantage of working on site where the resource is present and also use the processor's power. This eliminates all the middleware that is required for transporting the data to the client's site. The mobile agent's paradigm provides an effective solution to the problem of low latency, poor interface, and bad network conditions (Gray et al., 1996). The middleware and the communication control mechanism form the major workload in a client-server architecture; by eliminating them, we can build a better working environment and increase the efficiency of the system. The code and state of code is migrated to another machine for resuming its execution. This also eliminates the interface required for service access. The fact that there is no need for permanent connection makes it very suitable to the mobile environment. The ad hoc client-server model is overridden by the peer-peer model which matures into grid computing, where the machines can act as client or server depending on the environment. The programmer is swayed away from traditional multi-tier architecture to grid computing (Lauvset, 2001).

The majority of the mobile agents present in the literature is designed for a static network architecture. Baring a few mobile agents (Cabri, Leonardi, & Zambonelli, 2002; Kendall, Krishna, Pathak, & Suresh, 1998), intelligence is not embedded into them. We present a new set of mobile agents which work in a volatile mobile environment. Embedded with intelligence, agents can act autonomously, collaborate with other agents, and reduce the