Supporting Predictable Service Provision in MANETs Via Context-Aware Session Management

Radu Handorean, Washington University in St. Louis, USA
Rohan Sen, Washington University in St. Louis, USA
Gregory Hackmann, Washington University in St. Louis, USA
Gruia-Catalin Roman, Washington University in St. Louis, USA

ABSTRACT

The increasing ubiquity of wireless mobile devices is promoting unprecedented levels of electronic collaboration among devices interoperating to achieve a common goal. Issues related to host interoperability are addressed partially by the principles of the service-oriented computing paradigm. However, certain technical concerns relating to predictable interactions among hosts in mobile ad hoc networks (MANETs) have not yet received much attention. We introduce “follow-me sessions,” where interactions occur between a client and a service, rather than a specific service provider. A client, thus, may exploit several service providers during the course of its interaction with a given service. This redundancy mitigates the effects of mobility-induced disconnections, thereby facilitating reliable communication. The switching of service providers is done using a combination of strong process migration, context-sensitive binding and location-agnostic communication protocols. This paper covers the architecture and implementation of a middleware that supports follow-me sessions and shows how this middleware mitigates issues related to proxy-based service-oriented architectures in mobile ad hoc networks. We support our claims via technical evaluation of our approach.

Keywords: ad hoc networks; algorithms; mobile technologies; service-oriented computing; software architecture; wireless technologies

INTRODUCTION

Mobile devices today have limited computational power and persistent storage. In scenarios that require the execution of a small, computationally intensive piece of code — for example, running a public/private key encryption algorithm — the mobile device is stymied by its own lack of computational power. However, most modern mobile devices have built-in 802.11b wireless capability that allows them to communicate with proximal hosts. This capability on a reference host can be used to leverage off the capabilities of proximal hosts to achieve the required goal; that is, CPU-intensive code can be pushed over the wireless link onto more powerful hosts who execute the code and return the result to the caller. In both cases, a service refers to the code another host runs for the cli-
ent. The client controls the service even if the service is executing on a remote host.

Allowing such behavior requires solving a key problem. Physical mobility of the devices coupled with the modest range of 802.11b wireless cards results in limited intervals of time when two devices can communicate with each other. This is especially true in MANETs, which are the target environment for our work. However, an interaction (defined as a bounded sequence of message exchanges) between two hosts may need more time to complete than the interval of connectivity between them. For reliable service provision, it is desirable that an interaction between the client and the service provider, once begun, reaches completion. In MANETs, physical movement of hosts is independent of application semantics, and we consider it undesirable for the application to impose mobility restrictions. One solution is to have the client partially complete the task with the help of some host, pause its work and resume it on another host. This stretches the processing of a task over multiple hosts as they fall within the client host’s communication range, each contributing pieces of computation towards finishing the entire task.

The additional code required to support this kind of pause-transfer-resume computing can make programming for mobile environments prohibitively complex. It is in the interest of keeping the programmer’s effort low that we introduce a layer of abstraction called follow-me sessions (FMS), defined as a mechanism that preserves the sense of interaction between a client application and a service by masking the disconnections between intervals of connectivity. In essence, FMS offer, within limits, continuity of service provision.

The contribution of this paper is a set of mechanisms that support the delivery of FMS by using one or a combination of four strategies: (1) strongly migrating a process to an alternate host; (2) moving just the data state consisting of partial results to an alternate provider that offers a similar service; (3) allowing the client to temporarily disconnect from the service provider while the provider continues processing; or (4) having the client take back the partial results temporarily until a suitable alternate service provider can be found. The use of these mechanisms ensures that the process offering a service can migrate from host to host, so that it is always on a host in proximity to the client as the host on which the client is resident moves through space. We also present a complete evaluation of our approach, which yields further insight into the relative merits of these strategies.

The remainder of the paper is organized as follows: First, we present background material on service-oriented computing (SOC) and the unique challenges of MANETs. This is followed by mechanisms and strategies for delivering FMS. Implementation details and evaluation results are covered next, and is followed by a discussion of our results before we conclude.

BACKGROUND AND RELATED WORK

In this section, we provide an overview of the two main approaches to SOC — Web services (Alonso, Casati, Kuno, & Machiraju, 2003) and proxy-based services. We also show why proxy-based SOC is better suited for mobile environments. We also provide a brief discussion of the unique challenges imposed by MANETs.

Approaches to SOC

In SOC, a service provider advertises some functionality (service) it wishes to share by placing a service advertisement in a publicly available service directory. Clients discover and use these functionalities at run time. In proxy-based SOC, the service advertisement is in the form of a proxy object that can be retrieved by clients and used as a local handle to the service process on a remote server. The client interacts with the proxy as it would with any other local resource. The proxy tunnels requests to the provider’s server in most cases. However, in some cases, the proxy itself can deliver the entire functionality of the requested service and does not need to tunnel client calls to its parent server. From the application