Chapter XLVIII
Increasing the Performability of Wireless Web Services

Wenbing Zhao
Cleveland State University, USA

ABSTRACT

Wireless Web services are becoming a reality, if they have not already. The unique characteristics of the mobile devices and wireless communication medium, such as limited computing power, limited network bandwidth, limited battery life, unpredictable online time, mobility, and so forth,, imply that the infrastructure for wireless Web services will be very different from its wired counterpart. This chapter discusses the challenges and the state-of-the-art solutions to ensure highly performable wireless Web services. In particular, this chapter's focus is on three technical issues: optimization of the wireless Web services messaging protocol, caching, and fault tolerance. Finally, limitations of the current approaches and an outline of future research directions on wireless Web services are also discussed.

INTRODUCTION

With the evolution of the Internet technology, e-commerce, e-healthcare, and e-government services have become ubiquitous over the wired computer networks. There are obvious advantages in extending such services to mobile device users that connect to the Internet wirelessly. The possibility of conducting transactions reliably over wireless links also opens the door for business owners to offer additional services catering to mobiles customers specifically. Furthermore, the Web services technology, due to its strength in loose-coupling, extensibility, and interoperability, has also been gaining momentum to become the dominating enabling technology for the next generation of e-services, including those offered wirelessly to mobile

devices (Ellis & Young, 2003). Thus, a case can be made for the merging and extending Web services with mobile technologies.

The Web services technology essentially transforms the Web from a publishing medium to a programmable platform. This transformation greatly improves the degree of automation of Web-based transactions, and makes it easy to compose value-added services by integrating existing services. However, great care is needed to extend the Web services technology, which was primarily designed to run on powerful stationary computers over wired networks, to mobile devices over wireless networks. This is so because of the aforementioned limitations of the operating environment, such as limited computing power, limited network bandwidth, limited battery life of mobile devices, unpredictable mobile user online time, and mobility. In this chapter, I introduce the background, challenges, and state-of-the-art solutions to enable the offering and consuming of Web services from mobile devices over wireless networks. I focus particularly on how to improve the performability of the wireless Web services. (Performability is an umbrella term referring to the runtime performance, availability, reliability and security of a service.) Business owners should strive to offer highly performable wireless Web services to succeed in this domain.

BACKGROUND

The Web Services Concept

There is no universal definition of the term Web services and its interpretation varies dramatically. On one end, it refers to any services offered over the World Wide Web. On the other end, only the services enabled by the Web services technology are referred to as Web services. In this chapter, I use the latter interpretation. The Web services technology refers to the set of standards that enable automated machine-to-machine interactions over the Web. The corner stones of the Web services technology include eXtensible Markup Language (XML) (Bray et al., 2006), HyperText Transfer Protocol (HTTP), Simple Object Access Protocol (SOAP) (Gudgin et al., 2007), and Web Services Description Language (WSDL) (Christensen et al., 2001).

XML is designed to facilitate self-contained, structured data representation and transfer over the Internet. XML is extensible because it allows users to define their own tags. The extensibility of XML makes it the essential building block for Web services. HTTP is an application-level protocol following the client-server interaction model. HTTP is designed to share and access Web resources (i.e., hypertext objects). However, HTTP is a stateless protocol in that it does not keep track of the state across different HTTP requests. The design of HTTP not only removes unnecessary complexity, it ensures high degree of scalability as well. As the name suggests, SOAP was originally designed to conduct remote procedure calls over the Web. It has evolved to become the main communication protocol to exchange XML documents. Like many public-domain application-level protocols, such as SMTP, a SOAP message contains a SOAP Envelop and a SOAP Body. A SOAP message often contains an optional SOAP Header element, and a Fault element if an error is encountered by the sender of the SOAP message. WSDL is an XML-based language used to describe Web services. For each Web service, the corresponding WSDL document specifies the available operations, the messages involved with the operations, and a set of endpoints to reach the Web service. Due to the use of XML, WSDL is also extensible. In particular, it allows the binding of multiple different communication protocols and message formats.

To enable dynamic services publishing and discovery, a Universal Description, Discovery and Integration (UDDI) (Clement et al., 2004) service could be used. UDDI provides the standard way for Web services providers to describe their services, and the consumers to search and discover the available services. However, the experiments of offering global-scale public-domain UDDI registries were not successful, as evidenced by the shutdown of once well-known UDDI registries provided IBM, Microsoft and SAP (Krill, 2005). One reason for this failure is perhaps due to the current business requirement, *i.e.*, inter-enterprise transactions can rarely take place before the relevant legal documents are signed by the companies involved in the transactions. There are also unresolved questions related to ownership, responsibility and enforceability of transactions occurring through UDDI. Nevertheless, UDDI has been gaining momentum to be used as a Web services registry *within* each enterprise. Thus, from an architecture point of view, the Web services platform consists of the Web services providers, the Web services consumers, and the UDDI registries that broker the providers and the consumers, as shown in Figure 1.