Wireless SOAP: Optimizations for Mobile Wireless Web Services

Naresh Apte[1]
DoCoMo USA Labs
181 Metro Drive
San Jose, CA, USA
+1 (408) 573-1050
apte@docomolabs-usa.com

Keith Deutsch[2]
DoCoMo USA Labs
181 Metro Drive
San Jose, CA, USA
+1 (408) 573-1050
deutsch@docomolabs-usa.com

Ravi Jain[3]
DoCoMo USA Labs
181 Metro Drive
San Jose, CA, USA
+1 (408) 573-1050
jain@docomolabs-usa.com

ABSTRACT

We propose a set of optimization techniques, collectively called *Wireless SOAP (WSOAP)*, to compress SOAP messages transmitted across a wireless link. The *Name Space Equivalency* technique rests on the observation that exact recovery of compressed messages is not required at the receiver; an equivalent form suffices. The *WSDL Aware Encoding* technique obtains further savings by utilizing knowledge of the underlying WSDL by means of an offline protocol we define. We summarize the design, implementation and performance of our Wireless SOAP prototype, and show that Wireless SOAP can reduce message sizes by 3x-12x compared to SOAP.

Categories and Subject Descriptors

C.2.2 Network Protocols: *Applications, Algorithms, Protocol architecture*. C.2.6 Internetworking: *Standards*

General Terms: Applications, Networks, Services, Compression, Performance.

Keywords: SOAP, Wireless, Web Services, WSDL.

1. INTRODUCTION

Mobile wireless clients suffer from limited bandwidth, energy, computing and storage resources, which makes the use of XML, with its verbose and redundant nature, as well as XML-based protocols, problematic [4][5][7]. Some examples of XML inefficiencies are text-serialization of numbers, end tags, and namespaces. Several optimizations have been proposed that apply to XML in general, e.g. [10], [4], [6], [13] and [2]. SOAP as a protocol built on top of XML suffers from the same issues, and adds a few on its own. For example, studies show that using SOAP uses 3x-10x more bandwidth than Java RMI [12], [8]. Techniques for compressing SOAP messages have thus been proposed in the literature.

We propose a set of optimization techniques, collectively called *Wireless SOAP (WSOAP)*, which rests on two observations. The first is that while prior techniques attempt to encode or compress SOAP messages to allow recovery in exact form, this is not required, and recovery in an equivalent form suffices; we call this approach *Name Space Equivalency (NPE)*. The second is that if the sender and receiver are aware of the underlying WSDL, substantial additional bandwidth savings can be obtained. We develop a simple synchronization protocol for use between the mobile device and gateway; typically it would need to be executed only once, and offline. We call this *WSDL Aware Encoding (WAE)*.

We have designed, implemented and evaluated WSOAP; our experiments indicate that WSOAP can deliver significant bandwidth reduction compared to generic SOAP. Note that secondary benefits, such as reduced message loss and communication energy, can also accrue.

We have also compared WSOAP to a generic off-the-shelf text compression algorithm, namely Jzlib [14], as well as existing SOAP optimization approaches such as plain WBXML [13] and Differential Encoding [12]. WSOAP differs from these approaches as follows. Observe that generic text compression is oblivious to inherent structure in an XML document, and is typically compute-intensive [6], a drawback for mobile devices. On the other hand the Wireless Application Protocol (WAP) Binary XML (WBXML) specification standardized by W3C [13] provides binary encoding of a generic XML document designed for compact transmission with no loss of semantic information; however WBXML is oblivious to SOAP message structure. Other XML-aware compression algorithms proposed in the research literature [4] [10] typically use static, non-adaptive techniques and are guaranteed to be exact, unlike the NPE approach we take in WSOAP. The Differential Encoding technique (DiffEnc) [12] is oriented specifically towards compressing SOAP messages by *guessing* the message structure at both the source and the sink, creating a skeleton based on that, and sending only the differences between the skeleton and the actual message. DiffEnc fundamentally assumes the skeleton is the same at both the service provider and consumer; absent a synchronization technique it can easily be fooled by differences in SOAP packaging by different SOAP processors. In addition, the use of XPATH expressions and its obliviousness to recurring elements in a SOAP message make this approach inefficient.

2. WIRELESS SOAP ENCODING

The basic principles in WSOAP are to (1) Provide static encoding based on SOAP schema; (2) Leverage WSDL service description to create adaptive encoding for Web Service interfaces; (3) Require functional message equivalence rather than exact reproduction; (4) Limit computational cost by favoring codification and lookup over computation wherever possible; and (5) Codify using a binary packaging scheme.

For the purposes of discussion we assume that the mobile device connects via a wireless link to a mobile gateway, which in turn connects via a wired network to the Web Service, as shown in Figure 1. WOAP operates over the wireless link only.

---

[1]Current contact: naresh_apte@yahoo.com
[2]Current contact: deutsch@alum.mit.edu
[3]Author for correspondence.

WWW 2005, May 10--14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

NPE. As a first step, the set of SOAP element tags can be codified as application-specific tags, as provided by WBXML [11]. This approach has drawbacks, in particular that it largely defeats the purpose of XML namespaces. NPE, in contrast, exploits the fact that within any given document, the choice of a specific prefix string to denote association with a namespace is arbitrary. For example, within a document, the tags <soap:Envelope> and <s0:Envelope> are equivalent as long as the prefixes soap and s0 are associated with same namespace.

Table 1. Message Size (bytes)

Service	Original message size	DiffEnc	Jzlib	WBXML	WSOAP (NPE only)	WSOAP (NPE+ WAE)
CricScore Request	516	292	264	464	138	42
CricScore Response	34,266	182,812	2,140	22,562	11,095	7,205
Stock Quote Req	335	429	231	272	149	53
Stock Quote Res	10,257	88,293	1,606	4,166	3,925	3,278

WAE. If the gateway and the mobile client both have access to the WSDL for each Web Service interface used by the client application, the WSDL can be analyzed to create the requisite coding tables. We propose a protocol for synchronizing the WSDL coding tables in the gateway with the client. See Fig. 2.

3. EXPERIMENTS AND RESULTS

Our experiments used the following freely available software: for Jzlib [14]; for WBXML, we used KXML [9]; for DiffEnc, we used diffxml [3]. All implementations were written in pure Java, running under JRE 1.4.2 with Windows XP SP1 on a Pentium (M) 2.79 GHz machine with 496 MB RAM. For workload, since there is no common accepted SOAP benchmark available, we chose two example messages, a cricket score service [1] and a stock quote service. See Table 1 for results.

We see that WSOAP can reduce message size by 3x-12x compared to SOAP. It outperforms DiffEnc and WBXML by large factors; in some cases DiffEnc in fact results in message size explosion. However, WSOAP only outperforms Jzlib for messages that consist largely of structured XML. For messages that consist largely of unstructured text data (such as SOAP response messages) Jzlib achieves 2x-3x better compression than WSOAP. While computation time results are omitted in this summary for brevity, we find the superior compression of Jzlib comes with significant increases in computation time.

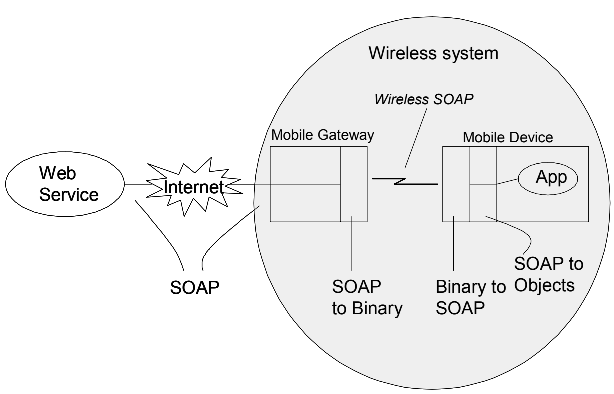

Figure 1. Scope of WSOAP implementation.

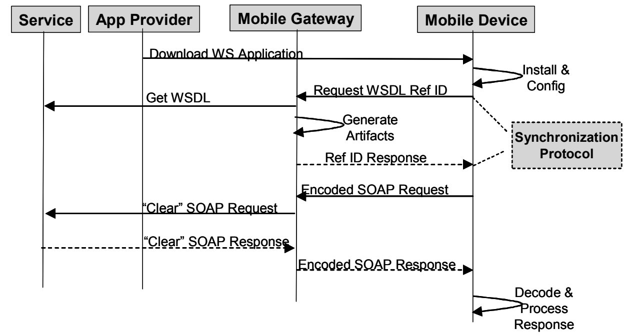

Figure 2. WSOAP WSDL-aware synchronization protocol.

4. FUTURE WORK

Future efforts planned for WSOAP include: (1) hybrid techniques to merge the advantages of generic compression and SOAP-aware compression; (2) automated WSDL analysis; and (3) application to other XML-based and WS-* protocols.

Acknowledgements. Our thanks to Raghu Dendukuri and Satya Seethasridhar for help during the performance experiments.

REFERENCES

[1] CricScore Statistics Web Service. http://crickscore.com/services/Statistics?wsdl.

[2] XML is a poor copy of Ess Expressions. http://c2.com/cgi/wiki?XmlIsaPoorCopyOfEssExpressions.

[3] XML Diff and Patch Utilities. http://diffxml.sourceforge.net/

[4] Cheney J. Compressing XML with Multiplexed Hierarchical PPM Models. IEEE Data Compression Conference, 163--172, 2001

[5] Crane, A. Does XML Suck?. http://xmlsucks.org, May 2002.

[6] Giradot, M. and Sundaresan, N. Millau: an encoding format for efficient representation and exchange of XML over Web. 9th International WWW Conference, May 2000.

[7] Govindaraju, M. In Lecture Notes Introduction to Grid Computing: Fall 2003. See http://www.cs.binghamton.edu/~mgovinda/courses/introToGridComputing-Fall03/., Sep. 2003.

[8] Kohlhoff, C., Steele, R. Evaluating SOAP for High Performance Business Applications: Real-Time Trading Systems. In World Wide Web Conference, May 2003.

[9] KXML. http://www.kxml.org/

[10] Liefke, H. and Suciu, D. XMill: an efficient compressor for XMl data. In ACM SIGMOD, 153-164, 2000

[11] Özden, M., A Binary Encoding for Efficient XML Processing, Masters' Thesis, Technische Universität Hamburg-Harburg, 2002.

[12] Werner, C., Buschmann, C. and Fischer, S. Compressing SOAP Messages by using Differential Encoding. IEEE International Conference on Web Services, July 2004.

[13] Wireless Application Protocol Forum, Ltd. Binary XML Content Format Specification. WAP Forum, 2001.

[14] JZlib. http://www.jcraft.com/jzlib/