ABSTRACT
Handheld mobile devices with wireless capability are gaining popularity. SOAP is a text-based protocol for Web services, but it has high overhead and its suitability for resource-constrained devices over wireless networks needs to be reevaluated. SOAP uses HTTP; HTTP in turn uses TCP as the underlying transport protocol for transmitting messages. However, TCP has a high overhead and high network latency. In this paper, a benchmark of the performance of different underlying transport protocols for SOAP is reported. We show that SOAP-over-HTTP and SOAP-over-TCP are inefficient and lead to high latency and transmission overhead for wireless networks. The results also show that SOAP-over-UDP provides much higher throughput compared to SOAP-over-HTTP.

Categories and Subject Descriptors
I.6.4 [Computing Methodologies]: Simulation and Modeling – Model Validation and Analysis.

General Terms
Measurement, Performance, Experimentation.

Keywords
SOAP, wireless networks, benchmarking, HTTP, TCP, UDP, mobile Web services, SOAP binding.

1. INTRODUCTION
Mobile technologies have gained much attention recently. More people are using handheld devices such as mobile phones, PDAs and laptops for their personal and business use.

With the fast development of wireless technologies, there is also a growing interest in extending traditional web applications to mobile environments. In such heterogeneous environments, a standardized communications mechanism such as Simple Object Access Protocol (SOAP) is needed to provide interoperability among different mobile platforms. However, SOAP was originally designed for wired networks; it does not address the challenges of wireless communications and the resource limitations of mobile devices such as slow CPU, memory constraints and limited battery life. Many issues need to be resolved in order integrate Web services in mobile applications [2]. SOAP performance is one of the most critical integration issues and has attracted research attention in the area of mobile Web services. Other issues are context-awareness, adaptability and security.

Existing research on SOAP performance [5, 6, 12, 13, 15 and 19] has found that current implementations of SOAP over HTTP are slower than other middleware technologies such as Java RMI and CORBA. These works are explained further in Section 3. Despite SOAP performance issues, XML, SOAP and WSDL together provide a framework for data exchange across various computing platforms and environments, and Web services are preferred over other middleware technologies for reasons of interoperability between heterogeneous systems.

The binding implementation of SOAP over HTTP is universally used on the Internet nowadays, but this implementation has some drawbacks especially in mobile environments. The disadvantages of using SOAP over HTTP in mobile computing are mostly due to the nature of HTTP and TCP protocols themselves. In specific, when a SOAP message is sent over HTTP, a TCP connection must be established, the connection is then closed once the SOAP message is sent. Both the opening and closing of a connection require a three-way handshake. Thus, sending SOAP messages over HTTP causes high overhead resulting in high transmission cost and latency.

Another reason for the low performance of SOAP in mobile environments is the nature of mobile communication networks which are characterized by low in bandwidth and intermittent connections. In addition, mobile applications have specific domain requirements such as security, responsiveness and availability of services. Therefore, there are some technical requirements that need to be met in order to deploy Web services in the mobile systems.

Recently, a SOAP-over-UDP specification has been proposed by BEA, Microsoft, Lexmark and Ricoh [13]. UDP is a simpler
protocol than TCP (HTTP uses TCP as its underlying protocol) and it provides a number of benefits over TCP as follows:

- The setup time associated with sending a message is reduced. This is because there is no connection required to send a UDP datagram.
- The bandwidth required to transmit UDP packets is smaller compared with TCP since the size of UDP packets is smaller. This benefit is particularly useful in wireless networks where customers are usually charged by the amount of data transmitted.
- With multi-casting nature of UDP, some new messaging models such as publish/subscribe of mobile Web services can be employed. This will be beneficial in sending notifications or alerts to multiple users.

This project undertook a benchmarking on the performance of different transport protocols namely HTTP, TCP and UDP as SOAP’s underlying protocol in wireless environments. The main objective of this benchmark is to see how different SOAP binding options, including SOAP-over-HTTP (with TCP as underlying transport), SOAP-over-TCP and SOAP-over-UDP perform in terms of execution time and throughput.

The results of the benchmarking will highlight the strengths and weaknesses of each binding option. With this information, we will be able to determine the most suitable binding to use for different kinds of applications. For example, some applications do not require the guaranteed delivery provided by TCP. In other scenarios, the request-messaging model of HTTP may not be appropriate; for example, a traffic update monitoring application that periodically sends observation data to a processing center may not need to receive any response from the server. In these cases, an alternative transport protocol other than HTTP and TCP with lower overhead may be more suitable.

The rest of the paper is organised as follows: Section 2 presents background information on Web services, SOAP and SOAP binding. The related work of other research on similar areas is discussed in Section 3. Section 4 will discuss the benefits of SOAP-over-UDP. The experiments and performance evaluation are presented in section 5. Section 6 concludes the paper with some highlights contributed by this work and plans for future work.

2. BACKGROUND

In this section, background information on SOAP and the SOAP binding framework are presented.

2.1 SOAP

Web services are an emerging technology, allowing interoperable communication between applications on different platforms. The deployment of Web services in mobile environment, which is called mobile Web services, is also increasing rapidly. However, due to the nature of mobile wireless networks, existing middleware infrastructure cannot efficiently support Web services on mobile devices.

Simple Object Access Protocol (SOAP) is a standard for Web services transportation. SOAP is designed with the aim of replacing traditional remote communication methods such as DCOM, CORBA and RMI. The main benefit of SOAP is interoperability. It allows applications written in different languages and deployed on different platforms to communicate with each other over the network. Furthermore, SOAP can be carried on top of any underlying protocols (such as HTTP, TCP, UDP, BEEP and SMTP). Thus, SOAP creators have defined a binding framework for SOAP instead of a fixed binding. Specifically, the SOAP binding framework specification [8] provides a high level of flexibility in terms of how SOAP messages are transmitted.

2.2 SOAP Binding

A SOAP binding is the boundary between SOAP and the underlying protocol. Such binding has to specify the syntactic and semantic rules for passing outgoing (incoming) SOAP messages between SOAP and the underlying protocol.

2.2.1 SOAP-over-HTTP

Over the Internet, HTTP is the protocol that is most widely used for SOAP binding. SOAP-over-HTTP is also the only concrete binding specification defined in the SOAP binding framework proposal. This is because HTTP is one of the core protocols of the Internet and is universally supported by Web servers. It also has the benefit of being able to pass through firewalls, which makes it a convenient candidate for transporting SOAP.

A SOAP message can be transported using HTTP by encapsulating a SOAP request into the message body of a HTTP GET or HTTP POST. Similarly, a SOAP response can be encapsulated into the body of a HTTP response.

2.2.2 SOAP-over-TCP

Similar to SOAP-over-HTTP, in order to transport a SOAP message using TCP as a direct underlying protocol, a SOAP message must be contained into the data octets part of a TCP packet. There is not yet official specification for SOAP binding with TCP, however, Apache Axis and Microsoft WSE (Web Service Enhancement) 2.0 already include APIs that enables the sending of SOAP messages via TCP channel [1 and 14].

2.2.3 SOAP-over-SMTP

Apart from the HTTP binding, the SOAP-over-email (or SOAP-over-SMTP) binding is also presented in the W3C specification. However, unlike the HTTP binding, which forms part of the SOAP standard, the SOAP-over-SMTP binding is only presented in the specification as an example to demonstrate the realization of the SOAP binding framework. In the SOAP-over-email binding, SOAP messages are encapsulated in the bodies of emails. This allows synchronous messages exchange between web services.

3. RELATED WORK

This paper focuses on the performance of different SOAP bindings in wireless environment. In this section, existing work on SOAP performance and web services in wireless environment are discussed. Earlier authors have looked at SOAP performance from different perspectives. A number of studies compared SOAP with other traditional distributed technologies (e.g. [5, 10, 12]) while others evaluated the performance of different SOAP implementations [16]. From these previous studies, it is noted that the current implementations of SOAP using HTTP as a transport protocol is slower than existing middleware technologies and other communication protocols such as CORBA and Java RMI.
In [4], the limitations of SOAP for scientific computing are investigated. Their experiments compared SOAP with Java RMI by sending large arrays of doubles; the results showed that SOAP is a factor of ten slower than Java RMI. Several techniques are proposed in this work to improve SOAP performance, specifically targeted at systems that require high performance for distributed scientific computing. Gryazin and Seppala compared SOAP and CORBA for mobile devices and found that the performance of web services does not depend on the implementation of SOAP [11]. They also concluded that SOAP is more suitable for larger scale systems with non-critical architectures while CORBA is more robust and flexible for mobile applications.

An experimental evaluation of SOAP performance in business applications has been presented in [12]. In this work, SOAP is compared with Financial Information eXchange (FIX) protocol which also uses a text-based wire representation as SOAP, and with CDR, a common binary wire format. The results demonstrated that it is possible for text-based protocols to achieve equivalent performance to binary protocols, but the text-based nature of XML is not sufficient to explain SOAP’s efficiency. Other projects have been working to improve SOAP performance by compression and caching mechanisms (e.g. [5, 6, 7 and 15]). It is shown in [15] that both clients and servers in poorly connected networks and clients with resource-constrained devices can benefit from compression. Devaram and Andresen [7] showed that an increase of 800% in performance can be achieved by caching SOAP payloads at the client side.

Interest in the usage of web services in mobile business applications is increasing. In a recent paper, Gebauer and Shaw suggested the deployment of web services in mobile environment will contribute to the success of mobile commerce [10]. Sun Microsystems has also released the specification for J2ME Web services to provide the capabilities of accessing remote XML-based Web services and parsing XML data to the J2ME platform. J2ME with kVM (kilo Virtual Machine) is a stripped down version of the Java Virtual Machine designed to have a small memory footprint between 40 and 80 kilobytes. The J2ME Web service specification provides standards and programming models for the Java community to develop web service applications for embedded devices.

kSOAP [8] is a SOAP API which provides an open source and small footprint implementation of XML aimed at developing applications for mobile devices using J2ME. In [18], the performance of Web services on J2ME and kSOAP clients is measured on a PDA and an IBM Thinkpad laptop. These clients access to a temperature service and a translation service which are available from the Internet. Their results showed that the main bottlenecks are the low available bandwidth causing large transmission time and the high cost associated with XML parsing.

Our evaluation study differs from others in the way that we provide a benchmark for various SOAP binding options in a wireless environment. Our results will be useful for other researchers to examine the effectiveness in performance of these transport protocols to the development of web service in wireless networks. From the results, we argue that SOAP-over-UDP provides certain performance benefits over the traditional SOAP-over-HTTP binding.

4. SOAP-over-UDP

Although HTTP is widely adopted as the de-facto binding option for SOAP messaging in existing Web services, it is inefficient when supporting Web services in wireless environments. HTTP uses TCP as the underlying transport, which performs inadequately in wireless environments due to a number of limitations. In particular:

- The congestion avoidance mechanism in TCP assumes packet losses are always due to congestions. However, in a wireless network, packet losses are usually due to disconnections and transmission errors.
- TCP requires a connection to be established before any payload data can be transmitted. Moreover, as data is received, acknowledgement packets are sent. This leads to additional overhead which may not be justifiable where bandwidth and client power are limited and reliable transmission of packets is not required.
- SOAP messages that carry only a small amount of data can finish transmitting while the TCP connection is still in its slow start phase. This results in poor utilization of the available bandwidth. The problem is particularly severe in wireless environments due to high round trip time.

To deal with these problems, we investigate SOAP-over-UDP as an alternative binding for SOAP messaging. UDP is a lightweight, unreliable and connectionless protocol. Due to its simplicity, UDP has little overhead and offers fast packet delivery. These characteristics may be desirable for applications where high transfer rate is valued over reliability, for example, real-time multimedia streaming. UDP is also suitable for quick exchange of information that is small in size.

5. EXPERIMENTS AND PERFORMANCE EVALUATION

This section presents the experiment setup and the results analysis. The experiment was intended to identify the performance tradeoffs when selecting different transport protocols for SOAP. Three SOAP bindings including SOAP-over-HTTP, SOAP-over-TCP and SOAP-over-UDP are tested and compared to each other.

5.1 Experiment Setup

A test system was set up to measure the performance of three above-mentioned transport protocols in SOAP communication. The machines used for testing are two desktop PCs which have similar hardware configuration as follows: Pentium III, 966Mhz CPU, 256 Mb RAM and running Java 2 SDK 1.4.1. Redhat Linux 9 is used as their operating systems. Both machines are equipped with an 802.11b wireless antenna and connected to each other through a Netgear WG602 wireless router. The experiments are tested under a wireless connection rate of 54Mbps.

The SOAP framework used in testing were Apache Axis 1.2 (on the server side) [1] and kSOAP 2.0 (on the client side) [7], Tomcat 5.0 is used as the server to host the Web service. Both the client and the server are written in Java. The test driver was adopted from the Sun WSTest 1.0 [17]. The test service includes the following methods:
• `echoVoid`: echoing empty message, no deserialization or serialization;
• `echoString`: echoing message of type string;
• `echoDouble`: echoing message of type double;
• `echoStruct`: echoing an array of different sizes with each element consisting of a structure. The repeating structures within the array contain one element each of type integer, float and string data types. The longer the array size is, the more effort is required in deserializing and reserializing the SOAP object to and from XML document;
• `echoList`: echoing a linked list of any size, each element consists of the same structure used in `Struct` as defined above.

Each method is an `echo` function in which a client sends a data message of a particular data structure to the server and the server responds back to the client with the same message that the server received. The test is highly configurable. The number of concurrent running clients, the startup (time allocated for warm up the system, running (the interval when throughput is measured) and ramp-down (time allocated for completing operations in flight) time can be specified. The sizes of the data in each service call were also varied.

### 5.2 Transmission Overhead Measurements

The average size of each message type for different bindings which are derived through the experiments is shown in Figure 1. The size of a SOAP message is the sum of the packet overhead and the message payload. Figure 2 describes the message overhead in terms of connection overhead and packet overhead of using HTTP, TCP and UDP binding for different types of messages. The packet overhead represents the SOAP payload and the protocol header. It can be seen that TCP and UDP have very similar packet overheads. However, the connection overhead of using SOAP-over-TCP makes TCP more expensive than UDP.

![Figure 1: Number of bytes sent for different message structures](image1)

These messages measured were request messages with a minimal payload for each type of message structure thus the overhead is due to connection establishment handshake including the transmission of three separate packets (SYN, SYN-ACK and ACK). The overhead of UDP is only restricted to 8 bytes.

![Figure 2: Connection overhead vs. Packet overhead for different bindings](image2)

It is interesting to note that for messages carrying only a small number of parameters (e.g. `void` message and `double` message), using UDP can reduce message size by almost 50% compared to HTTP. This is critical for a mobile client that sends simple SOAP requests to a server in a repetitive pattern and as such significant saving in transmission cost and client energy assumption can be achieved.

### 5.3 Response Time

In this section, the impacts of using different binding options on response time and throughput are reported. Figure 3 shows the time it takes for a client to receive the response message from the server which has 5 clients connecting to it at the same time.

![Figure 3: Average response time for 5 clients](image3)

For all types of message requests, SOAP-over-UDP is significantly faster than SOAP-over-TCP and SOAP-over-HTTP. The low responsiveness of TCP and HTTP is mainly due to the additional cost in encoding and decoding the HTTP and TCP headers and also time spent on connection establishments. The following figure plots the average response time while there are...
different numbers of clients running simultaneously (e.g. 5, 20, 35, and 50 clients).

As illustrated from the above chart, the response time for SOAP-over-UDP is about half of the response time for SOAP-over-TCP and one-third of that for SOAP-over-HTTP. This experiment shows that SOAP-over-UDP scales quite well with a large number of concurrent clients requesting for a service at the same time.

5.4 Throughput
Another important metric in benchmarking is throughput. Figure 5 illustrates the throughput (number of transactions processed per second) in the 5 clients scenario. As expected from the response time results, SOAP-over-HTTP and SOAP-over-TCP provide significantly lower throughput compared to SOAP-over-UDP in most cases. However, if the request and response message are large (in case for echoList method) the throughput of SOAP-over-UDP is not significantly higher than other binding options.

5.5 Packet Drops
During the experiments with SOAP-over-UDP, the numbers of packets dropped during transmissions are recorded and shown in the following table.

Table 1: Average packet drops rates in SOAP-over-UDP

<table>
<thead>
<tr>
<th>Message Types</th>
<th>5 Clients</th>
<th>20 Clients</th>
<th>35 Clients</th>
<th>50 Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>echoVoid</td>
<td>7</td>
<td>10</td>
<td>18</td>
<td>40</td>
</tr>
<tr>
<td>echoString</td>
<td>12</td>
<td>18</td>
<td>35</td>
<td>86</td>
</tr>
<tr>
<td>echoDouble</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>90</td>
</tr>
</tbody>
</table>

As can be seen from the above table, the number of packet drops increases as the number of simultaneous clients rises. Also, the longer a request takes; the higher the packet drop rate is. For example, the echoList method takes longer time to process than the echoVoid method; thus with the same number of clients accessing to the same method, the packet drop rate of the echoList case is much higher than the echoVoid case. As a result, with applications those have small transmitted message sizes; the unreliability of UDP does not affect much on its total performance. However, when large SOAP messages are required in certain applications, SOAP-over-UDP suffers high packet drops rates. This issue will need to be addressed to justify the usage of this kind of binding over the conventional SOAP-over-HTTP option.

6. Applications
This section provides some examples of various applications that are suitable for the different SOAP bindings.

6.1 SOAP-over-HTTP
The main advantage of the HTTP binding is that it allows SOAP messages to pass through firewalls and be processed by web servers. However, due to the size of the HTTP header, SOAP-over-HTTP incurs relatively high overhead when it is used to send simple SOAP messages. Following are some examples of applications where SOAP-over-HTTP is suitable:

- **Translation service** - where clients provide text in one language and the web service performs translation to another language and sends it back to the client.
- **Payment service** - where the web service collects sensitive credit card or banking information from clients, process the transaction and reply with a receipt/confirmation. HTTPS can be used instead of HTTP to provide security.

6.2 SOAP-over-TCP
Our results show that SOAP-over-TCP does not provide much benefit over SOAP-over-HTTP in terms of throughput and execution time. However, the packet size of a simple SOAP request using TCP is around 40% smaller than the equivalent request using the HTTP binding. As a result, TCP can be used for applications where clients send short requests to the server to request large amount of data, and the data needs to be transported reliably. An example of such an application may be:

- **Scientific data transfer** - where clients sent requests to retrieve large scientific data collections (e.g. planetary observations or gene bank DNA sequences.)

6.3 SOAP-over-UDP
The results obtained in Section 5 show that SOAP-over-UDP provides significantly higher throughput compared to SOAP-over-HTTP and SOAP-over-TCP. SOAP-over-UDP is most suitable for applications where short SOAP messages are sent frequently and reliability is not of concern. In addition, as UDP supports multicasting and broadcasting, it can be used in push-based web services. Some example applications where SOAP-over-UDP is suitable include:
• Postcode lookup - where clients provide a location and the web service reply with the corresponding postcode or vice versa.
• Stock quote push - where the server broadcast stock prices to a large number of mobile clients. The frequency of such push could vary from very high to very low as it depends on the movement of stock value.

7. Conclusion
Because of the low bandwidth and limited processor power on handheld devices, the existing implementation of SOAP using HTTP binding (with TCP as the underlying transport) is not sufficient for mobile clients to access mobile Web services. This paper has highlighted the limitations of SOAP over HTTP binding in mobile network and proposed an implementation of reliable UDP binding for SOAP. Also, we studied the performance of SOAP-over-HTTP, SOAP-over-TCP and SOAP-over-UDP.

We argue that while HTTP and TCP have many benefits consisting of the ability to pass through firewalls and reliability, these features may not needed by all applications. Our testing, we show that using HTTP or TCP binding for SOAP lead to high overhead and high execution time and low throughput compared to using UDP.

The next step of the project will be studying to design a reliable UDP binding for SOAP to provide reliability for SOAP message transmission, and compare its performance to the unreliable UDP binding. The improved SOAP binding option to be proposed will be more reliable in a way that they can handle disconnections caused by the instability of wireless communications.

8. Acknowledgement
This project is support by the ARC (Australian Research Council - under the Linkage project scheme, no. LP0218853) and SUN Microsystems grant no. 7832-030217-AIS.

9. REFERENCES