Proc. 2007 Computer Measurement Group Intl. Conf., San Diego, CA, Dec. 2-7, 2007.

AN EMPIRICAL EVALUATION OF COMMUNICATION AND PROCESSING OVERHEAD IN SERVICE ORIENTED ARCHITECTURES

Daniel A. Menascé Vinod Dubey
Department of Computer Science The Volgenau School of IT & Engineering
George Mason University
4400 University Dr., Fairfax, VA 22030, USA
menasce@gmu.edu vdubey@gmu.edu

Abstract

Most modern applications are networked and make heavy use of various communication protocols such as TCP/IP and SOAP over HTTP in case of Web Services. This paper presents the results of an experimental evaluation of the communication and inherent processing overhead of these two protocols in the context of client/server applications. Experiments were carried out to determine the impact of the size of reply message as well as the complexity of the SOAP reply message in terms of number of elements. Analysis of variance was used to determine the statistical significance of the differences obtained.

1 Introduction

Service Oriented Architectures (SOA) enable a multitude of service providers (SP) to provide loosely coupled services to consumers that may only be known to them at run-time. The binding between service consumers and service providers is established after the service and its description are discovered. An example of technology that can be used to support SOAs is Web Services. The basic protocol used to access services in the Web Services context is SOAP (Simple Access Object Protocol), an XML-based protocol that allows a service consumer to invoke services in a Remote Procedure Call style.

The most common transport protocol for SOAP is HTTP, which in turn runs over TCP. In this paper we are interested in assessing the basic communication and processing overhead of SOAP/HTTP. We use as a benchmark plain TCP as the mechanism for service consumers to request services from service providers. It is qualitatively expected that SOAP, for being more complex, has a higher communication and processing overhead when compared with plain TCP. A quantitative evaluation is needed though to determine if indeed the difference is significant at a high confidence level.

For the purpose of our analysis, we implemented a simple service that receives a request and returns a reply, whose content is not relevant to this study, of varying message size. The following experiments were conducted:

- One-byte reply message: this experiment is aimed at determining the basic communication and processing overhead of the two protocols in question, regardless of message size. These experiments allow us to use the protocol as the single factor in the analysis.
- Impact of message size: in this case, the size of the reply message was varied from one byte to 20K bytes and the impact of this factor was analyzed. An analysis that includes two factors—message size and protocol used—was conducted.
- Complexity of the SOAP reply message: in this experiment, the message size was kept constant but the number of elements in the SOAP reply message was varied from one to 20. A single factor (the message complexity) analysis was conducted in this case.

The rest of the paper is organized as follows. Section two reviews basic concepts in Web Services and introduces the notation used in the rest of the paper as well as some basic relatonships. Section three describes the experimental setup. The next section presents the results of the three experiments described above. Finally, section five presents some concluding remarks.

2 Basic Concepts and Notation

2.1 Web Services

Web services have become one of the most popular integration enabler technologies in the IT industry today. A Web service is a software component or application exposed through standard eXtensible Markup Language (XML) Internet protocols that allow applications built on different hardware and software platforms to interoperate easily [7]. The collection of XML-based protocols that support the realization of Web services comprises: a) Web Service Definition Language (WSDL) [6], b) Simple Object Access Protocol (SOAP) [5], and c) Universal Description, Discovery and Integration (UDDI) [4]. WSDL is an XML-based format for service definition. SOAP is an XML-based protocol that provides access to Web Services; it provides a format to exchange messages between the applications. UDDI is used to register and discover Web services described in WSDL. The most commonly used application transport protocols for Web services applications is HTTP, although other protocols such as JMS or SMTP could be used as well.

Typically, an application based on Web services consists of a service provider and one or more service consumers. The service provider make data and applications available for use via services and the service consumer uses these services provided by the service provider. Alternatively, service providers can publish their services to a UDDI server and service consumers can discover these services from the UDDI server and then communicate with the service providers directly. Additionally, Web services-based applications are increasingly using the Enterprise Service Bus (ESB) to provide a loosely coupled integration scheme and thus avoid the disadvantages of a tightly coupled integration.

Figure 1 shows a typical sequence of operations in Web Services. First, a service producer publishes the description of its services in a UDDI register. This enables a service consumer to discover the service and understand through its WSDL description how to invoke its services. Finally, the service consumer directly invokes the services of the service producer using SOAP.

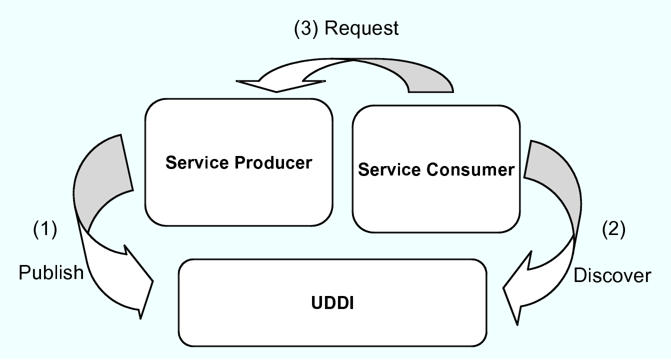


Figure 1: Web services sequence of operations.

SOAP can be used to invoke remote Web Services using a Remote Procedure Call (RPC) mechanism. However, SOAP messages can be quite involved and require significant decoding of the XML format before the actual service can be executed. In this paper, we provide an empirical evaluation of the overhead incurred in using SOAP (see Fig. 2-b) when compared with the simple use of TCP (see Fig. 2-a).

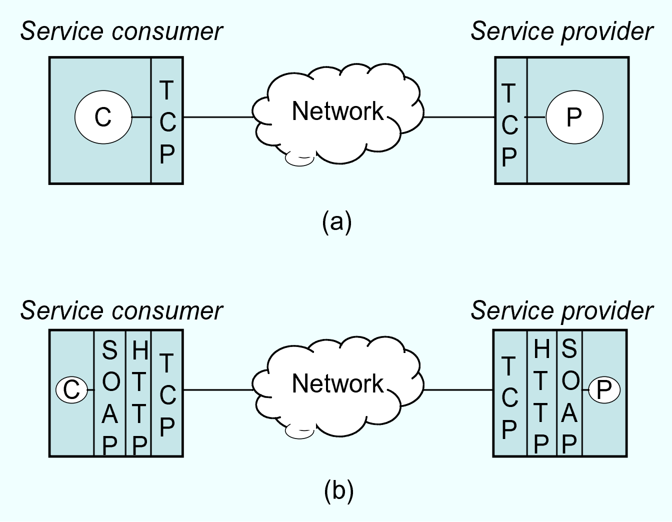


Figure 2: Interaction between the consumer of a service and a provider. (a) Through TCP (b) Through SOAP over HTTP.

2.2 Notation and Basic Relationships

We introduce in this section some of the basic concepts and notation used throughout the paper. A client machine with M_c client threads submits requests over a TCP connection to a server machine. We consider two possible communication protocols for the client to communicate its request to the server and receive the reply: 1) TCP and 2) SOAP over HTTP. We assume that the client's request to the server is always constant in size and equal to 1 byte. The reply size is equal to m bytes which will be varied in our experiments since we are interested in studying the effect of message size in the response time of a request for the two protocols considered.

We now provide some simple expressions for the response time of requests under these protocols using the notation of Table 1. If TCP is the communication protocol used, the response time is given by

$$RT_{\text{TCP}}(m, t_s) = t_{\text{TCP}}(1) + t_s + t_{\text{TCP}}(m) \quad (1)$$

where the first term is the communication time to send the request to the server over a TCP connection, the second is the processing time at the server, and the third term is the time for the reply to return to the server over a TCP connection.

The response time when SOAP is used is given by

$$RT_{\text{SOAP}}(m, t_s, p) = t_{\text{SOAP}}(1, 1) + t_s + t_{\text{SOAP}}(m, p) \quad (2)$$

m	Message size in bytes.
p	Number of elements in the SOAP message.
$t_{\mathrm{TCP}}(m)$	Communication time, in sec, over a TCP network as a function of m.
$t_{\mathrm{SOAP}}(m,p)$	Communication and processing time of a SOAP message, in sec, as a function of m and p
t_s	Server processing time, in sec.
M_c	Number of client threads
$RT_{\mathrm{TCP}}(m,t_s)$	Response time, in sec, for a request as a function of m and t_s when TCP is used.
$RT_{\mathrm{SOAP}}(m,t_s,p)$	Response time, in sec, for a request as a function of m, t_s, and p when SOAP is used.

Table 1: Notation.

where p is the number of fields of the SOAP reply message. Since SOAP messages travel over HTTP, which lies on top of TCP, the overhead $O_{\mathrm{SOAP}}(m,p)$ due to processing SOAP messages at the server plus the additional communication overhead due to the XML in SOAP is given by

$$\begin{aligned} O_{\mathrm{SOAP}}(m,p) &= [t_{\mathrm{SOAP}}(1,1) - t_{\mathrm{TCP}}(1)] + \\ & \quad [t_{\mathrm{SOAP}}(m,p) - t_{\mathrm{TCP}}(m)]. \end{aligned} \quad (3)$$

This overhead includes the HTTP processing plus SOAP message processing at either end.

3 Experimental Setup

To enable the experiments conducted in this paper, we designed and developed two different client/server applications using a J2EE platform to assess and quantify the impact of network and protocol complexities and message sizes on response times in distributed environments: a socket-based client/server applications for the TCP case and a Web services -based client/server application to assess SOAP/HTTP.

More specifically, the experiments were conducted using the J2EE/Weblogic 8.1 Application/Web server platform running on a Windows/XP machine with 2 GB of RAM and a 1.5 GHz Intel Pentium processor. The Java-based client and server run on two different machines connected via the Internet in a LAN environment. The Weblogic server uses the Servlet multi-threaded model to host Web services. The environment was configured to use five threads per Web service. The remaining configuration parameters for the deployment of the services in the server were used out of the box. We used Sun's Java Version 1.4.2_05 and the Xerces XML parser that were part of the Weblogic platform.

TCP connections were established before data collection started since the purpose of the study did not intend to include the time spent in TCP's three-way handshake. A large number of requests were sent between the service consumer and the service provider in a loop. Response times were accumulated in an in-memory data structure and written to a file after all requests were processed. The output file was opened in MS Excel to allow us to conduct analysis of variance (ANOVA).

ANOVA is a technique used to determine the level of statistical significance of one or two factors in an experiment as well as the source of the variability in the results [2]. MS Excel can be used for that purpose through the Tools → Data Analysis menu. The two versions of ANOVA used here are "Anova: Single Factor" and "Anova: Two-Factor With Replication."

4 Empirical Analysis of Communication Overhead

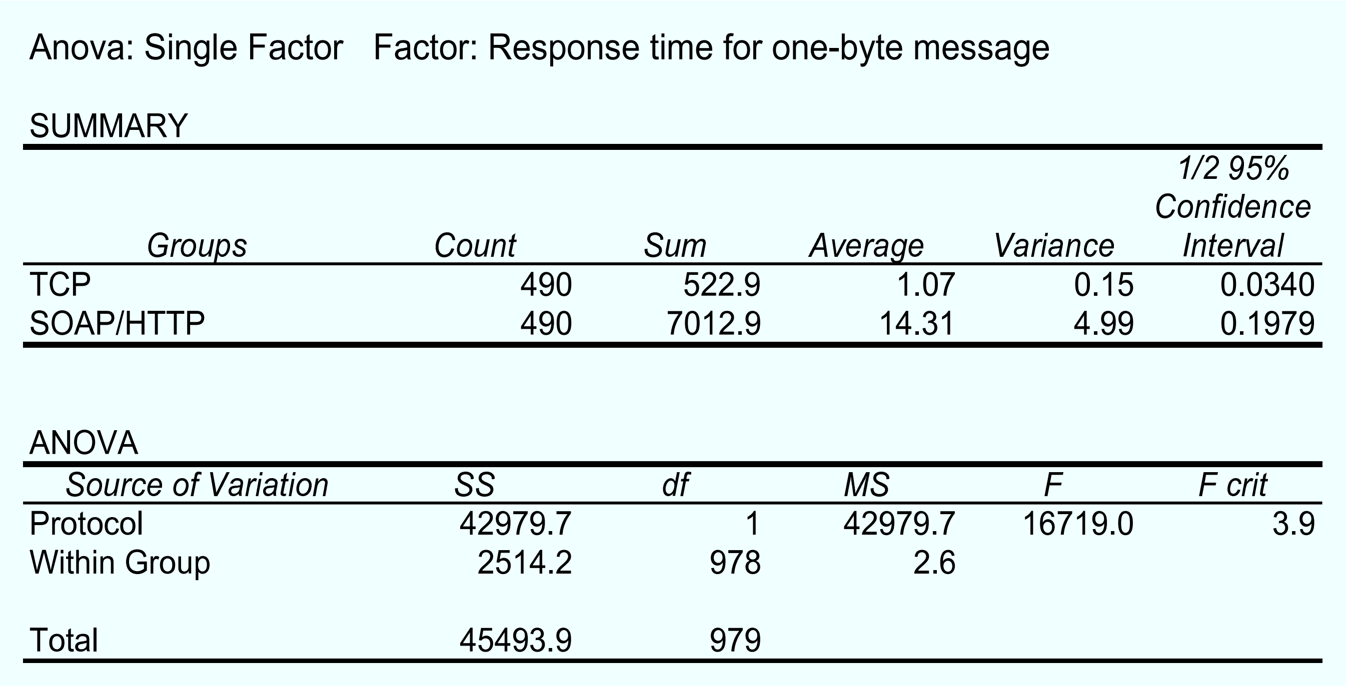
We conducted a series of experiments aimed at quantifying and comparing the influence of the two protocols, message size (m), and the complexity of SOAP messages (p) on the response time. In order to isolate the impact of these factors, we used a small (i.e., one-byte request) and constant request size and a zero server processing time for the request (i.e., $t_s = 0$).

4.1 One-byte Response Time Experiments

These experiments were run with server responses equal to one byte in size. The purpose of this experiment was to verify the impact of the two protocols on response time in terms of their constant overhead regardless of message size.

Figure 3 shows the results of one-way ANOVA on the one-byte response times. The factor considered is the protocol: TCP or SOAP/HTTP. The ANOVA results show that the hypothesis that the response times are the same for each of the two protocols is rejected at the 95% confidence level because the F value (i.e., 16719) is greater than the critical F value (i.e., 3.9). This analysis shows also that 94.5% (i.e., $(42,979.7/45,493.9) \times 100$) of the variance can be explained due to the difference in the protocol used and 5.5% of the variance is due to experimental error, i.e., other factors not accounted for.

The figure also shows that the average response time for one-byte TCP requests is 1.07 msec with a 95% confidence interval of ± 0.0340. The average response time for one-byte SOAP/HTTP requests is 14.31 msec (i.e., 13.4 time larger than in the TCP case) with a 95% confidence interval of ± 0.1979.



Anova: Single Factor   Factor: Response time for one-byte message

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*	*1/2 95% Confidence Interval*
TCP	490	522.9	1.07	0.15	0.0340
SOAP/HTTP	490	7012.9	14.31	4.99	0.1979

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*F crit*
Protocol	42979.7	1	42979.7	16719.0	3.9
Within Group	2514.2	978	2.6		
Total	45493.9	979			

Figure 3: Single factor ANOVA results for the one-byte message response time where the factor is protocol type (TCP and SOAP/HTTP).

4.2 Protocol and Message Size Comparison

This next set of experiments is aimed at measuring the influence of message size on the response time on each of the two protocols. Figure 4 displays the average response time results (in msec) as well as their 95% confidence intervals for reply message sizes ranging from 1 byte to 20,000 bytes. In this experiment, $p = 1$ for all message sizes.

Since the confidence intervals do not overlap for each message size, we can say, as expected, that SOAP/HTTP has a much higher overhead [3]. In fact, the average overhead $O_{\mathrm{SOAP}}(m, 1)$ ranges from 13 to 20 msec and the SOAP/HTTP average response time is between 2 and 13.5 times bigger than that when TCP is used.

We performed 2-way ANOVA with replication on the response times. The two factors considered were the protocol (TCP or SOAP/HTTP) and the message size (1 byte, 100 bytes, 500 bytes, 1500 bytes, 3000 bytes, 4500 bytes, 5000 bytes, 10000 bytes, 15000 bytes, and 20000 bytes). A summary of the data is shown in Fig. 5. The figure shows for each message size and protocol, the average and variance of the response time values. For the case of SOAP/HTTP, we used $p = 1$.

The results of the 2-way ANOVA (see Fig. 6) indicate that at the 95% level, a) message size is a significant factor (because the computed F value of 50.1 is greater than the critical F value of 1.9), b) the choice of protocol is signifi-

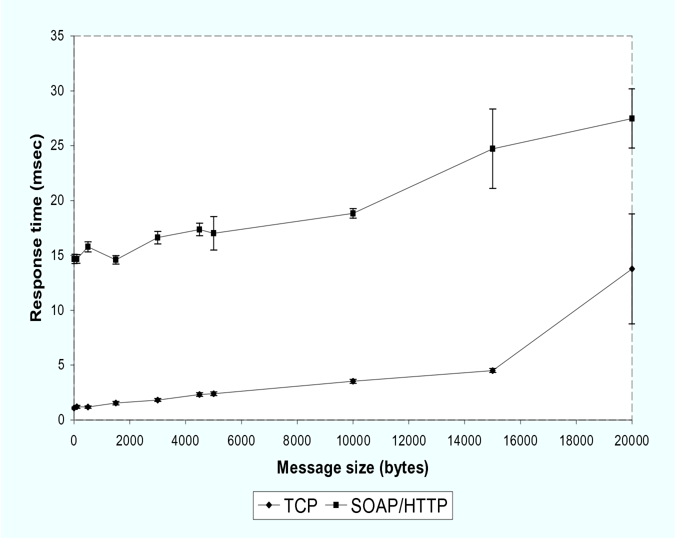
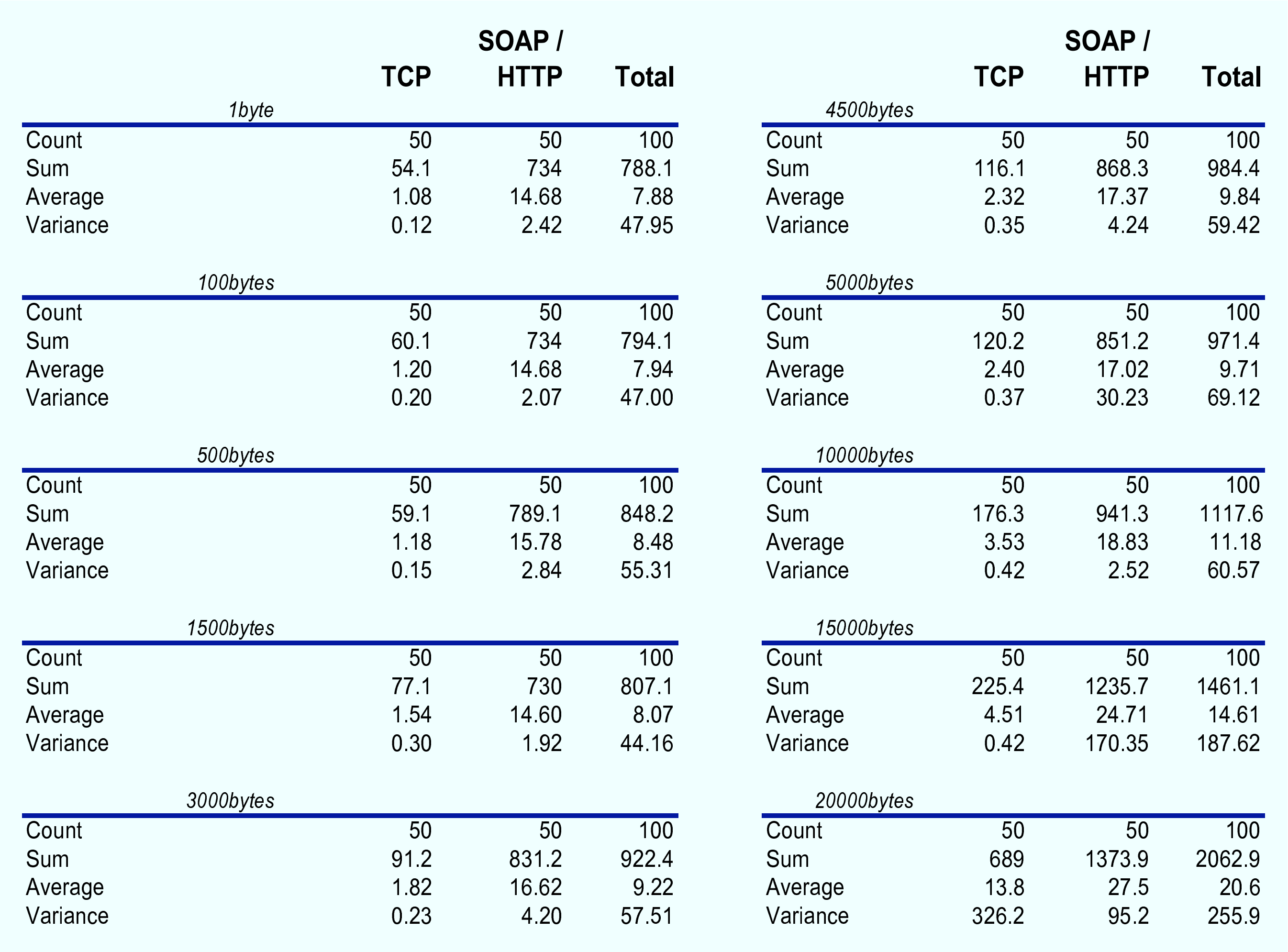


Figure 4: Response time (in msec) vs. message size (in bytes) for TCP and SOAP/HTTP.



1byte

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	54.1	734	788.1
Average	1.08	14.68	7.88
Variance	0.12	2.42	47.95

100bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	60.1	734	794.1
Average	1.20	14.68	7.94
Variance	0.20	2.07	47.00

500bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	59.1	789.1	848.2
Average	1.18	15.78	8.48
Variance	0.15	2.84	55.31

1500bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	77.1	730	807.1
Average	1.54	14.60	8.07
Variance	0.30	1.92	44.16

3000bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	91.2	831.2	922.4
Average	1.82	16.62	9.22
Variance	0.23	4.20	57.51

4500bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	116.1	868.3	984.4
Average	2.32	17.37	9.84
Variance	0.35	4.24	59.42

5000bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	120.2	851.2	971.4
Average	2.40	17.02	9.71
Variance	0.37	30.23	69.12

10000bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	176.3	941.3	1117.6
Average	3.53	18.83	11.18
Variance	0.42	2.52	60.57

15000bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	225.4	1235.7	1461.1
Average	4.51	24.71	14.61
Variance	0.42	170.35	187.62

20000bytes

	TCP	SOAP / HTTP	Total
Count	50	50	100
Sum	689	1373.9	2062.9
Average	13.8	27.5	20.6
Variance	326.2	95.2	255.9

Figure 5: Summary data for 2-factor ANOVA (protocol and message size) for response time results.

cant (because the computed F value of 1708 is greater than the critical F value 3.9), and c) there is significant interaction between message size and protocol choice (since 3.2 is greater than the critical value of 1.9).

The fact that there is significant interaction at the 95% confidence level between message size and protocol size is corroborated by the fact that the curves of Fig. 4 are not parallel [2].

The results of Fig. 6 indicate that 14.2% of the variability is explained by the difference in message size and 53.9% by the protocol used.

4.3 SOAP Overhead

In order to understand the impact of the number of elements in the SOAP message on the response time, we ran experiments with a fixed reply message size of 20K bytes but with different number of elements (1, 2, 4, 5, 8, 10, and 20). The results of this analysis are shown in Fig. 7. The top part of that figure shows the average and variance of the response time for the different number of elements in the SOAP message. The bottom part of the figure shows a one-factor ANOVA which indicates that a 95% confidence level, the number of elements in a SOAP message is a significant factor (because the F value of 3.2 exceeds the critical value of 2.1). The number of elements in a SOAP message explains 5.2% of the variance in the results of the experiments.

5 Concluding Remarks

This paper presented the results of an experimental evaluation of the communication and inherent processing overhead of using TCP and SOAP/HTTP in the context of SOAs. Three experiments were conducted: one-byte reply message, message size and protocol variation, and complexity of the SOAP reply message. The results show that:

- One-byte reply experiments: the use of SOAP/HTTP make a statistically significant difference at the 95% level with SOAP/HTTP exhibiting a response time over 13 times larger than plain TCP.
- Message size and protocol variation: at a 95% confidence level, a) message size is a significant factor, b) the choice of protocol is significant, and c) there is significant interaction between message size and protocol choice.
- Complexity of the SOAP message: at a 95% confidence level, the number of elements in a SOAP message is a significant factor in the response time.

It is important to emphasize that in addition to the specific numerical results obtained in this paper, this work shows the importance of sound experimental work in experimental Computer Science [1].

References

[1] D. Feitelson, "Proceedings of he Workshop on Experimental Computer Science," San Diego, CA, June 13-14, 2007.

[2] R. Jain, *The Art of Computer Systems Performance Analysis*, John Wiley, 1991,

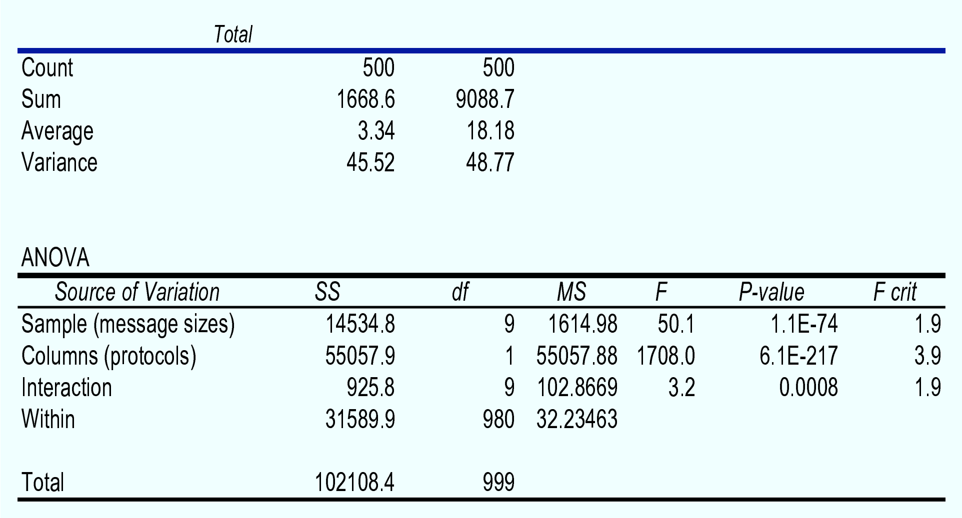
[3] D.A. Menascé, V.A.F. Almeida, and L.W. Dowdy, *Performance by Design: Computer Capacity Planning by Example*, Prentice Hall, Upper Saddle River, 2004.

[4] OASIS, UDDI Version 3.0 Speciication, www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3

[5] W3C, SOAP Version 1.2, April 27, 2007, www.w3.org/TR/2007/REC-soap12-part0-20070427/

[6] W3C, Web Services Description Language, www.w3.org/2002/ws/desc/

[7] World Wide Web Consortium (W3C) - *http//www.w3c.org*

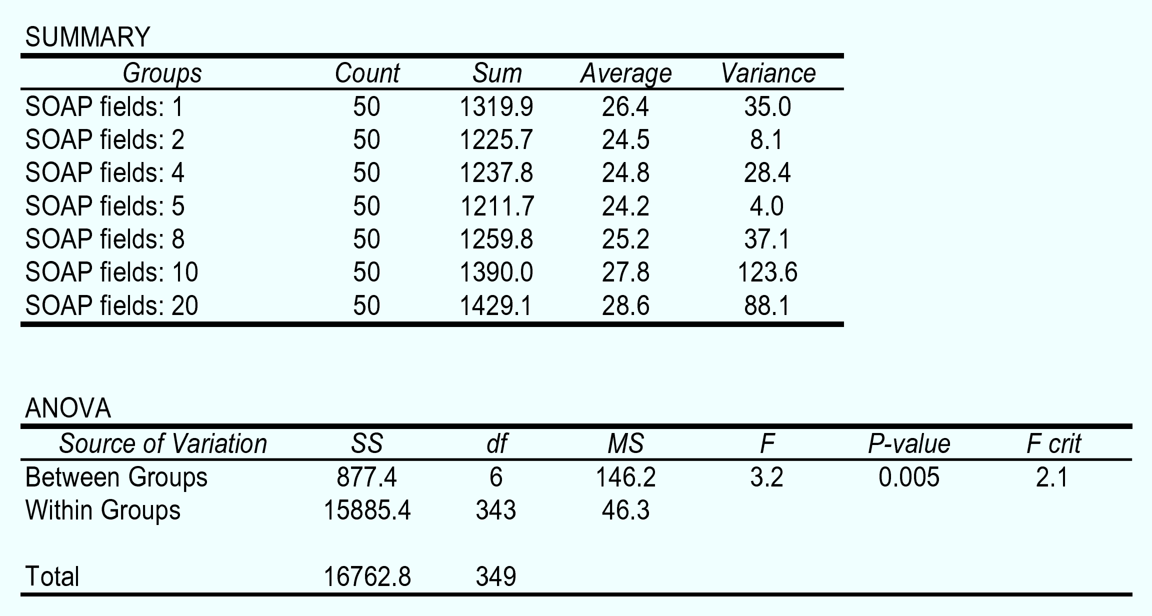


	Total	
Count	500	500
Sum	1668.6	9088.7
Average	3.34	18.18
Variance	45.52	48.77

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Sample (message sizes)	14534.8	9	1614.98	50.1	1.1E-74	1.9
Columns (protocols)	55057.9	1	55057.88	1708.0	6.1E-217	3.9
Interaction	925.8	9	102.8669	3.2	0.0008	1.9
Within	31589.9	980	32.23463			
Total	102108.4	999				

Figure 6: Results for 2-factor ANOVA (protocol and message size) for response time results.



SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
SOAP fields: 1	50	1319.9	26.4	35.0
SOAP fields: 2	50	1225.7	24.5	8.1
SOAP fields: 4	50	1237.8	24.8	28.4
SOAP fields: 5	50	1211.7	24.2	4.0
SOAP fields: 8	50	1259.8	25.2	37.1
SOAP fields: 10	50	1390.0	27.8	123.6
SOAP fields: 20	50	1429.1	28.6	88.1

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	877.4	6	146.2	3.2	0.005	2.1
Within Groups	15885.4	343	46.3			
Total	16762.8	349				

Figure 7: Results for 1-factor ANOVA for response time results where the factor is the number of fields in the SOAP message. Message size is fixed at 20K bytes.