

# Improving the performance of Web Services in Disconnected, Intermittent and Limited Environments

Joakim Johanson Lindquist

November 4, 2015





# **Abstract**

In this thesis I investigate different techniques to improve the performance of Web services in typical tactical network environments.



# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Background and Motivation . . . . .	13
1.1.1	Service-Oriented Architecture . . . . .	13
1.1.2	Military Networks . . . . .	15
1.2	Problem Statement . . . . .	15
1.3	Premises . . . . .	16
1.4	Scope and Limitations . . . . .	16
1.5	Research Methodology . . . . .	16
1.6	Contribution . . . . .	16
1.7	Outline . . . . .	16
<b>2</b>	<b>Background</b>	<b>17</b>
2.1	Related Work . . . . .	17
2.1.1	Proxies . . . . .	17
2.1.2	Configuration . . . . .	17
2.2	Web services . . . . .	18
2.2.1	XML . . . . .	18
2.2.2	Service descriptions: WSDL . . . . .	18
2.2.3	SOAP . . . . .	18
2.2.4	Other Web services . . . . .	19
2.3	Disconnected, Intermittent and Limited . . . . .	19
2.3.1	Network metrics . . . . .	19
2.3.2	Other constraints . . . . .	19
2.4	Optimization techniques . . . . .	20
2.4.1	Where to place the optimization? . . . . .	20
2.4.2	Compressing the payload . . . . .	21
2.4.3	Reducing overhead of SOAP . . . . .	21
2.5	Proxies . . . . .	21
2.6	Requirement Analysis . . . . .	21
2.7	Summary . . . . .	22
<b>3</b>	<b>Design and Implementation</b>	<b>23</b>
3.1	Overall Design . . . . .	23
3.2	Proxy . . . . .	23
3.2.1	Squid . . . . .	23
3.2.2	Apache Camel . . . . .	23
3.2.3	Tuning application server configuration . . . . .	23

3.2.4	Alternative transport protocols	23
3.3	Summary	23
<b>4</b>	<b>Testing and Evaluation</b>	<b>25</b>
4.1	Evaluation Tools	25
<b>5</b>	<b>Conclusion and Future Work</b>	<b>27</b>
5.1	Conclusion	27
5.2	Future Work	27

# **Acronyms**

**DIL** Disconnected, Intermittent and Limited. 13, 14, 17, 18

**NEC** Network Enabled Capability. 11

**SOA** Service-Oriented Architecture. 11–13



# List of Tables

2.1 Optimization possibilities . . . . .	20
2.2 Transport protocols . . . . .	21
2.3 Proxy requirements . . . . .	22



# **List of Figures**

1.1 The three roles in SOA(from [3]) . . . . . 14

3.1 Architectural overview of proposed design . . . . . 23



# **Chapter 1**

## **Introduction**

Military units may operate under conditions where the reliability of the network connection is low. They can operate far from existing communication infrastructure and rely only on wireless communication. Such networks are often characterized by unreliable connections with low date rate and high error rates making data communication difficult. In a military scenario it is necessary for units at all operational levels to seamlessly exchange information across different types of communication systems. To NATO, this concept is referred to as Network Enabled Capability (NEC). In a feasibility study, NATO identified the Service-Oriented Architecture (SOA) and Web Services as key enablers[1].

Web services are well tested in civil environments where the network is stable and the data rate is abundant. However, military networks may suffer from high error rates and very low date rate, which can leave the Web services unusable. In this thesis I investigate how these challenges can be overcome by applying optimization techniques at different layers of the protocol stack.

### **1.1 Background and Motivation**

NATO is a military alliance consisting of 28 member countries[2] and which primary goal is to protect the freedom and security of its members through political and military means. In joint military operations the relatively large number of member countries can be a challenge when setting up machine-to machine information exchange. Differences in communication systems and equipment can attribute to make data communicating difficult. In order to address this issue, NATO has chosen the SOA concept[3].

#### **1.1.1 Service-Oriented Architecture**

Service-Oriented Architecture (SOA) is an architectural pattern where application components provide services to other components over a network. SOA is built on concepts such as object-orientation and distributed computing and aims to get a loose coupling between clients and services. In

their reference model for SOA[4], the Organization for the Advancement of Structured Information Standards (OASIS) define SOA as:

*Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.*

In SOA business logic is divided into smaller chunks of logic, referred to as *services*. A service can be business related, e.g a patient register service, or a infrastructure service used by other services and not by a user application. Services are provided by *service providers* and are consumed by *service consumers*. The service provider is responsible for creating a service description, making the service available to others and implementing the service according to the service description. Services are made available to service consumers through a form of *service discovery*. This can be a static configuration, or more dynamic with a central *service registry*, where service providers publish service descriptions. Service consumers find the services they need by contacting the service registry. The communication between services occur through the exchange of standardized messages.

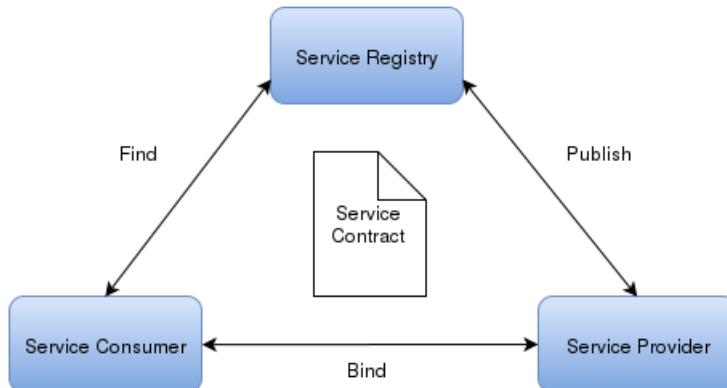


Figure 1.1: The three roles in SOA(from [3])

Following the SOA principles dictates a very loose coupling between services and the consumers of those. This allows software systems to be more flexible, as new components can be integrated with minimal impact on the existing system. Another aspect of loose coupling is with regard to time, which enable services and its consumers to not be available at the same instance of time. This enables asynchronous communication. Loose coupling with regards to location allows the location of an service to be changed without needing to reprogram, reconfigure, or restart the service consumers. This is possible through the usage of service discovery, which is dynamically retrieval of the new location of the service.

Furthermore SOA enables service implementation neutrality. The implementation of service is completely separated from the service description. This allows re-implementation and alteration of a service without affecting the service consumers. Thus this can attribute to keep development costs low and avoiding proprietary solutions and vendor lock-in. Another benefit with SOA is re-usability by dividing common business processes into services, which may help cost reduction and avoids duplication. SOA is only a pattern and the concepts can be realized by a range of technologies. The most common used approach is the Web service family of standards, using the SOAP messaging protocol. To achieve interoperability between systems from different nations and vendors, NATO has chosen the Web service technology in order to realize the SOA principles. This allows member nations to implement their own technology as long as they adhere to the standards. Web services are discussed in detail in section 2.2.

### **1.1.2 Military Networks**

Web services can be used in strategic military networks as they have network infrastructure with the same characteristics as civil networks.

### **Tactical Networks**

Tactical networks are characterized by that the units are deployed to operate in a battlefield, which means there is no existing communication infrastructure. They use tactical communication equipment which includes technologies like VHF, UHF, HF, tactical broadband and satellites[3]. Examples of such units are mobile units like vehicles, foot soldiers and field headquarters. The tactical network connects deployed headquarters with mobile units. These types of networks are unpredictable and may have very low date rate, possibly high delay, high error rates and frequent disconnections. They are often called disadvantaged grids or Disconnected, Intermittent and Limited (DIL) which is the term used in this thesis. NATO studies has identified such networks to have the following characteristics:

*Disadvantaged grids are characterized by low bandwidth, variable throughput, unreliable connectivity, and energy constraints imposed by the wireless communications grid that link the nodes[5].*

The characteristics of these networks and what challenges they impose are discussed in detail in section 2.3.

## **1.2 Problem Statement**

Most of the Web Service solutions used today are aimed for civilian use and does not necessarily perform well in military environments. In contrast to civilian networks where the date rate is abundant, mobile tactical networks may suffer from high error rates and low date rate. Adapting Web service

solutions meant for civil networks directly for military purposes may not be possible. Therefore, Web services needs to be adapted in order to handle network challenges. However, it can be very expensive to alter existing Web service technology and incorporate proprietary solutions. A NATO research task group has previously identified the foundation on open standards to avoid tighter coupling between service providers and consumers[3]. It is much better to use commercial off-the-shelf software. By placing the optimization in proxies, the Web services can remain unchanged.

The goal of this thesis is to investigate different optimization techniques that can be applied in order to improve Web service performance in DIL networks. In order for the clients and services to remain interoperable the optimization techniques will be placed in proxies. The Web Services will communicate as normal, while all network traffic is tunneled through a proxy. The Web Service itself does not need to pay attention to the bad connectivity, the proxy will choose the appropriate protocol and configuration.

### **1.3 Premises**

The Web services does not need to be changed.

### **1.4 Scope and Limitations**

Snevre inn oppgaven.

Sikkerhet er ikke adressert i denne oppgaven.

### **1.5 Research Methodology**

Denning.

### **1.6 Contribution**

The outcome of this thesis is an recommendation regarding which optimizations techniques which can be used in DIL to enhance the performance of Web services.

### **1.7 Outline**

Hvordan er resten av oppgaven strukturert.

# **Chapter 2**

# **Background**

## **2.1 Related Work**

Diskuterer eksisterende arbeid.

IST-090 ser på muligheten av å bruke SOA på taktisk nivå, med spesielt fokus på disadvantaged grids. Fokuserer på mulighet for Web services across disadvantaged networks. Samt Data Distribution Service(DDS) på taktisk nivå.

Three key issues that needs to be addressed in order to apply Web services in tactical networks has been identified in previous work[6]:

### **End-to-end connections**

Attempting to establish and maintaining connections in DIL environments can lead to increased communication overhead and possible complete breakdown of communication. The dependency on end-to-end connection should be removed.

### **Network heterogeneity**

Different technologies used. Different performance in networks may lead to buildup of data in buffers, risking loss of information.

### **Web Service overhead**

Web services generate overhead. Optimization approaches should seek to reduce the network traffic generated by Web services.

#### **2.1.1 Proxies**

DSproxy, AFRO proxy.

#### **2.1.2 Configuration**

IST-90: Configure HTTP on the application server or ESP to prevent time-outs. Anbefaler at hvis man trenger å gjøre propensiære optimaliseringer, så

burde de plasseres i proxier.

## 2.2 Web services

Web services are client and server applications that communicate over a network and can be used to implement a service-oriented architecture. There exists many definitions of Web services where the core principles are the same, but the finer details may vary. The World Wide Web Consortium has defined Web Services as[7]:

*A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.*

This definition points out a set of standards that enables machine-to-machine interactions. These standards are discussed in the following sections. Figur her.

### 2.2.1 XML

Extensible Markup Language(XML) is a markup language and is considered as the base standard for Web services. An XML document consist of data surrounded by tags and is designed to be both machine and user readable. In a Web service, XML is used to tag the data.

### 2.2.2 Service descriptions: WSDL

Web Services Description Language(WSDL) is an interface definition language that using XML describes functionality offered by a Web service. The interface describes available functions, data types for message requests and responses and binding information about the transport protocol, as well as address information for locating the service.

### 2.2.3 SOAP

SOAP is an application level XML based protocol specification for information exchange in the implementation of Web services. It is transport protocol agnostic and can be carried over various protocols. The far most used is HTTP over TCP, but other protocols such as UDP and SMTP can be used as well. A SOAP message consist of a optional header and a required body. The header can contain information not directly related to the message, such as routing information for the message. The body contains the data being sent, known as the payload.

#### **2.2.4 Other Web services**

However, there also exist other types of Web services which does not follow the previously discussed standards. RESTful web services let users manipulate data using a set of stateless operations. It uses exclusively HTTP. REST services are easy to understand and have gained a lot of traction in the civil industry in the latest years.

### **2.3 Disconnected, Intermittent and Limited**

The DIL concept refers to three characteristics of a network. As discussed in the introduction, military tactical networks may suffer from these constraints. These limitations are discussed in this section.

**Disconnected** Military units that participating in a tactical network are highly mobile and may disconnect from a network either voluntarily or not. This causes topology changes. Unplanned loss of connectivity can be due to various reasons, such as loss of signal or equipment malfunction. The disconnected term refers to that nodes in the network may be disconnected for a long time, possibly for multiple days.

**Intermittent** Nodes in a DIL may loose connection temporarily before reconnecting. The duration range from seconds to minutes.

**Limited** The Date rate, how many bits that are sent per second, is limited in DIL networks. Various aspects that affects the date rate are discussed in the next section.

#### **2.3.1 Network metrics**

**Link throughput** The link throughput is influenced by how large distance there is between the units communicating.

**Link reliability** How much of the arriving data that is correct. This is called *bit error rate* or *packet error rate*. With high error rates, more data to be transmitted again due to the data arriving being incorrect. This contributes to longer transmission time. In a military setting, an enemy may deliberate sabotage the network with jamming, causing higher error rates.

**Link latency** The communication technology in use influences how fast data transmission can be done. Long delay may cause that the application sending data timing out.

#### **2.3.2 Other constraints**

Energy constraints imposed by the wireless communication grid. The battery capacity and the transmission range of the communication equipment

for mobile units may be limited. Another issue is that in some cases military units are required to enter radio silence in order to avoid being detected by the enemy. During radio silence units may only receive data and not send any.

## 2.4 Optimization techniques

The Web service technology enable interoperability between systems, but also increase the information overhead, requiring higher data rate demands. Employing Web Services developed for use in civilian networks directly into a DIL environment may not perform satisfactorily. To increase the performance we can apply optimization techniques. There are many approaches and optimization techniques which can be applied at different levels of the protocol stack. In the coming sections the different optimization techniques are presented and a overview is presented in Table 2.1. Another issue that needs to be addressed is, when we have identified optimization techniques, where do we place them? In the application itself or in a proxy? This is discussed in the next section.

### 2.4.1 Where to place the optimization?

One approach is to modify the Web service application itself. However, this would mean that every application that is used in a tactical network would require modification. This would require a lot of resources and severely limit the flexibility of using Web services. Another solution is, by using proxies, we can place the optimization there without altering the Web Services themselves. The only thing required to do is to setup the application to send and receive data through the proxy. The proxy will take care of the optimization for tactical networks. This seems like a more reasonable approach and is explored in this thesis.

Protocol Stack	optimization possibilities
The application	Optimize the application
Web service messaging: SOAP	Optimize SOAP, e.g XML compression
HTTP/TCP, UDP or other transport protocols	SOAP is transport agnostic. Other protocols can be used.
IP	NATO NEC feasibility study states that all protocols should be over IP.
Lower layers	Not in the scope of this thesis.

Table 2.1: Optimization possibilities.

### **2.4.2 Compressing the payload**

The first optimization techniques deals with the optimization of the encoding. By compressing the Web service payload, we can reduce the amount of data that need to be sent.

- GZIP
- EFX(Efficient XML).

Previous experiments shows EFX has the compression results with GZIP as the second best alternative[8].

### **2.4.3 Reducing overhead of SOAP**

HTTP/TCP is the most used transport protocol for SOAP messages, but since SOAP is transport protocol agnostic different protocols can be used. Experiments show that this is possible.

<b>Transport Protocol</b>	<b>Summary</b>
HTTP over TCP	Widely used. Breaks down in DIL environments.
Military Message Handling System(MMHS)	Optimize the application
Stream Control Transmission Protocol(SCTP)	Features multihoming.
Advanced Message Queuing Protocol(AMPQ)	Application level protocol. Employs a broker architecture with store-and-forward capabilities.
SOAP directly on TCP	Its possible

Table 2.2: Transport protocols

## **2.5 Proxies**

A proxy is a server that acts as intermediary for requests between a client and server over a network. Proxies can be used to accomplish different tasks, from caching web content to bypassing filtering and censorship. We can group proxies into three types, forwarding proxies, reverse proxies and gateway proxies.

## **2.6 Requirement Analysis**

As discussed in section 2.1, the dependency on end-to-end connections needs to be removed. This can be done by adding proxies to the network. Mobile units have to carry batteries with them and the capacity is therefore limited. Advanced compression techniques may reduce the overhead, but also requires more battery. This trade-off needs to be considered.

<b>Requirement</b>	<b>Priority</b>
Receive and forward HTTP 1.X requests	1
Allow modifications on the payload	1
Allow configuration of HTTP timeouts	1
Keep HTTP-connection alive	1
Support protocol X and y	2

Table 2.3: Proxy requirements

## 2.7 Summary

# Chapter 3

# Design and Implementation

## 3.1 Overall Design

## 3.2 Proxy

### 3.2.1 Squid

Squid is a fully-featured HTTP/1.0 proxy.

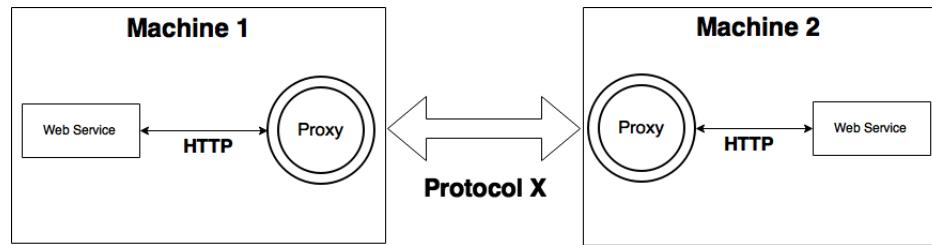


Figure 3.1: Architectural overview of proposed design

### 3.2.2 Apache Camel

### 3.2.3 Tuning application server configuration

### 3.2.4 Alternative transport protocols

## 3.3 Summary



## **Chapter 4**

# **Testing and Evaluation**

### **4.1 Evaluation Tools**



# **Chapter 5**

## **Conclusion and Future Work**

### **5.1 Conclusion**

### **5.2 Future Work**



# Bibliography

- [1] P. Bartolomasi et al. *NATO network enabled capability feasibility study*. 2005.
- [2] NATO. *NATO - Member Countries*. [http://www.nato.int/cps/en/natohq/nato\\_countries.htm](http://www.nato.int/cps/en/natohq/nato_countries.htm). Accessed: 2015-05-04.
- [3] F. Annunziata et al. “IST-090 SOA challenges for disadvantaged grids”. In: (Apr. 2014).
- [4] OASIS et al. *Reference Model for Service Oriented Architecture 1.0 OASIS standard*. <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>. Accessed: 06. 10. 2015. Oct. 2006.
- [5] A. Gibb et al. “Information Management over Disadvantaged Grids”. In: *Task Group IST-030/ RTG-012, RTO-TR-IST-030* (2007). Final report of the RTO Information Systems Technology Panel.
- [6] F.T Johnsen et al. “IST-118 - SOA recommendations for Disadvantaged Grids in the Tactical Domain”. In: (2013).
- [7] Hugo Haas and Allen Brown. *Web Services Glossary*. <http://www.w3.org/TR/ws-gloss/\#webservice>. Accessed: 2015-05-06.
- [8] Frank T. Johnsen and Trude Hafsoe. “Using NFFI Web Services on the tactical level: An evaluation of compression techniques”. In: 13th International Command and Control Research and Technology Symposium (ICCRTS). Seattle, WA, USA, 2008.