Paper

Web Services Efficiency in Disadvantaged Environment

Joanna Śliwa[a], Tomasz Podlasek[a], and Marek Amanowicz[a,b]

[a] Military Communication Institute, Zegrze, Poland
[b] Military University of Technology, Warsaw, Poland

Abstract—The article presents results of web services (WSs) efficiency tests carried out in the testbed emulating disadvantaged network environment. The authors discuss the advantage of different WS adaptation techniques that allow to minimize the XML message size (i.e. compression, filtering and binary coding) and the size of JPEG image attachment (i.e., resolution reduction, decreasing colour depth, JPEG compression). The presented results show the efficiency of selected methods that adapt the web services realization to the possibilities of the network. The article is summarized by conclusions and recommendations in terms of sending XML SOAP messages in disadvantaged networks.

Keywords—compression, filtering, image resize, SOA in tactical networks.

1. Introduction

Service oriented architecture (SOA) is one of the crucial enablers to achieve network enabled capability (NEC) and mission effectiveness [1]. It provides means for information sharing among various elements of federation of systems (FoS). It also offers services to arbitrary service consumers, meets their information needs, takes part in a business process carried out within military operation and thus, supports achieving the net-centricity.

Many of the countries have chosen web services (WS), as the most interoperable and easily extendable SOA platform, widely used in commercial applications and supported by the biggest and most advanced software developer companies in the world. WSs are described by a wide range of standards that deal with different aspects of services realization, transport, orchestration, semantics, etc. They provide means to build a very flexible environment that is able to dynamically link different system components to each other. These standards are based on the extensible markup language (XML) that was developed for the enterprise systems and operates in high bandwidth links. XML became very popular because it solves many interoperability problems, even though it adds a significant overhead.

The utilization of web services in SOA-based systems operating in NEC environment has been shown in many international experiments (e.g., coalition warrior interoperability demonstration (CWID) [2], currently CWIX). They prove that the WS technology improves collaboration, interoperation and information sharing in federation of systems (FoS). In order to achieve efficient information exchange between the users and to give the biggest advantage to the operators, the SOA solutions need to work with different types of information and communication systems. The challenge is though, to use this simple in concept and providing big flexibility means of communications on all command levels, including low bandwidth tactical communications systems.

Tactical network environment used by the lowest echelons of command has many limitations that make it difficult to provide reliable communications. It usually copes with high error rates, intermittent connectivity problems, radio silence and frequent disruptions. It also often changes its topology. Data rates of the low tens of kilobits per second and below are common.

Given their maturity level and the software tool support, web services would be the best choice for all command levels. However, the above-mentioned factors characterizing tactical disadvantaged environment provide limitation for the XML-based SOA information distribution mechanisms. The application of web services in tactical systems has been though, subject to experiments. According to the NATO C3 Agency (NC3A) report on using WS in tactical domain [3], web services continue to function adequately even at the lowest levels of network capacity, although their performance is diminished. They remain promising even in low bandwidth links, as long as very fast response times are not required.

In order to fully take advantage of SOA and web services methods for adapting WS realization in tactical networks must be provided. It is, though, very important to discover the level of WS applicability in a disadvantaged environment and to investigate the efficiency of different optimization mechanisms that enable the performance of simple XML SOAP communications to improve.

The remainder of this paper is organized as follows: Section 2 presents the test scenario and the results of tests that were carried out by our team, and discusses the possibilities of WS realization optimization methods; Section 3 presents the summary and recommendations for SOA solutions in low bandwidth tactical environment.

2. SOA Web Services Efficiency Tests' Results

The objective of the tests was to discover the edge network parameters below which web services cannot be realized. As the web services provider, the authors used own implementation of the blue force tracking (BFT) service that sends unit tracking information consistent with the STANAG 5527 NFFI (NATO Friendly Forces Information) standard [4]. This service was augmented with the possibility to send image sensor information (JPEG files). The authors focused on verifying the efficiency of:

– pure SOAP message exchange,

– SOAP message filtered out (allowing to send only obligatory NFFI information about the unit),

– compressed SOAP message exchange,

– exchange of JPEG images of different sizes.

2.1. SOAP Messages Filtering

The first experiments were focused on showing how the filtering (extracting only obligatory NFFI XML schema data) and the Gzip compression of NFFI messages can decrease the time needed to receive response from the service. The authors assumed that a response SOAP message consists of 30 objects.
BFT service based on NFFI can send tracks of objects. The track following NFFI 1.1 XML schema contains 3 types of information about the object:

– positional data – obligatory: track source, data/time, coordinates; optional: bearing, speed, reliability, inclination;

– identification data (optional) – unit symbol, unit short name;

– operational status data (optional) – footprint, strength, status code, alert, remarks.

Filtering is a process of cutting out all optional information elements of the NFFI message from the original message (so called "long track"), leaving only the obligatory ones (so called "short track"). This process results in limiting the SOAP message size (see Fig. 1).

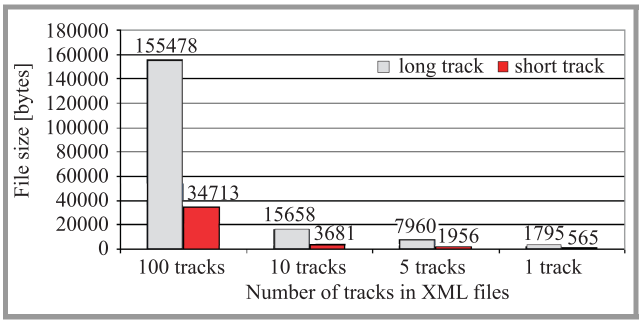

Fig. 1. Filtering gain.

The results of the experiment show that the use of filtering with Gzip compression remarkably reduces the size of SOAP messages and makes it possible to exchange NFFI messages properly through low bandwidth networks. By using compression we are able to provide information exchange in circumstances where the available bandwidth is about 2 kbit/s. Simultaneously, the time when the application receives a response was decreased below 7 s (see Fig. 2).

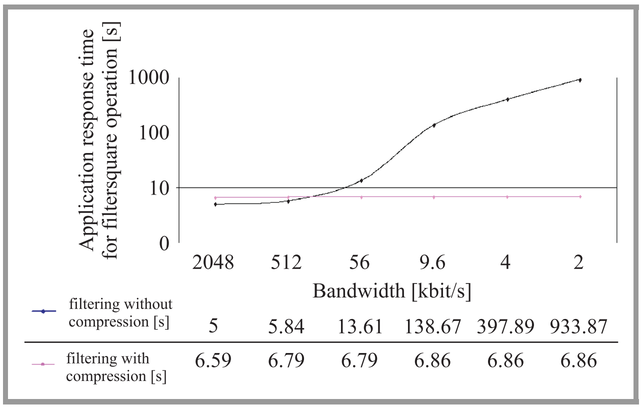

Fig. 2. Application response time for mediation service operation (in seconds) of available throughput.

As depicted in Fig. 2, the same information sent without using compression through links with low throughputs takes much more time (application response time is about 934 s). It is unacceptable in case of exchanging short-life information (e.g., quickly changing positions of forces, sensor information). The results show that it is recommended to use the compression when the traffic flows trough tactical, low bandwidth networks, particularly where the available throughput is below 56 kbit/s.

2.2. Gzip Compression and Efficient XML

Standard web services traffic bases on the exchange of SOAP messages usually sent in the request-response mode. The most popular and widely supported by SDKs SOAP binding is HTTP over TCP/IP. This protocol stack performs well in a stationary high bandwidth networks and over the Internet. However, its performance in a tactical environment is usually diminished [3]. In order to evaluate the level of web services performance in a disadvantaged environment, the authors carried out tests of "pure" SOAP communications and compression/binary encoding techniques that significantly reduce the message size. The web service producer was the Blue Force Tracking service.
In general, we distinguish "lossy" and lossless compression techniques. In case of a data transmission system, designers are interested in using only lossless ones. These compression algorithms usually exploit statistical redundancy in such way, as to represent the sender's data more concisely without error. The lossless compression uses the fact that most of the real-world data has statistical redundancy. However, lossless data compression algorithms will always fail to compress data that has high entropy

(high disorder). This applies to already compressed data, random data or encrypted data. In such cases the attempts to compress data will usually result in an expansion.

One of the most popular and efficient lossless compression algorithms is Gzip[1], which was created as a fusion of two algorithms: Lempe-Ziv (LZ77) and Huffman Coding. It is based on the DEFLATE algorithm, which was designed to replace LZW and other patented data compression. It finds duplicated strings in the data to be compressed. The second occurrence of a particular string is replaced by a pointer to the previous one, in form of a pair represented by the distance and length. Distances are limited to 32 kB and lengths are limited to 258 B. When a string does not occur anywhere in the previous 32 kB, it is represented as a sequence of accurate bytes.

This characteristic of Gzip indicates that it can be satisfactory only with data that has some order. This makes it inappropriate, e.g., for already compressed files and the encrypted ones. In the tests, the authors were using this algorithm to compress SOAP messages – a text document in XML format. For this data type, Gzip compression achieves a high gain. However, for other data formats, e.g., already compressed JPEG images, which can be also sent as attachments to SOAP, it does not yield satisfactory results.

In Fig. 3, there has been depicted the message size comparison for different number of NFFI tracks sent in original XML SOAP messages and in SOAP messages compressed with Gzip.

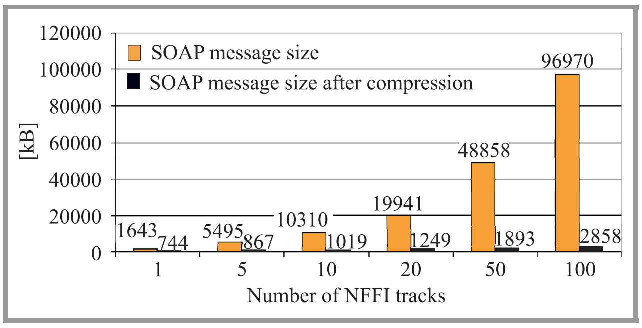

Fig. 3. SOAP message size comparison – original SOAP with compressed SOAP message.

It is clear that the bigger the message is – the best performance we get. This performance can be described as compression gain (CG) which the authors define as follows:

$$CG = 100 - \frac{size\_of\_compressed\_message}{size\_of\_original\_message} 100 \quad (1)$$

Compression gain can be, though, even 97% for 100 kB XML SOAP message, but reaches only 54% when the original message size is 1.6 kB (see Fig. 4).

[1]More information can be found Gzip home page, http://www.gzip.org/

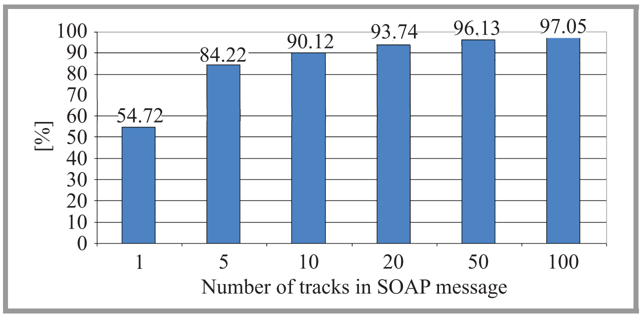

Fig. 4. Compression gain.

It should be noted that the process of compressing the message is resource consuming and takes considerable amount of time. On the server with processor with 4 cores (2.8 GHz) it took even 30 ms for the biggest 100 – track message (see Fig. 5).

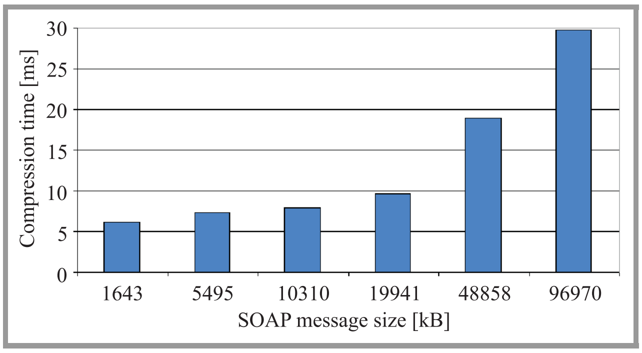

Fig. 5. Compression time for different sizes of SOAP message.

As compression is a resource consuming process, it has been proven that in an Ethernet with 100 Mbit/s throughput, it does not give considerable effects. The compression gain resulting from minimizing the message size is limited by the fact that the machine needs time for compressing and decompressing the message (see Fig. 6). The application response times for a compressed and a not compressed SOAP message transmission are though very similar when the network links are quite fast and reliable.

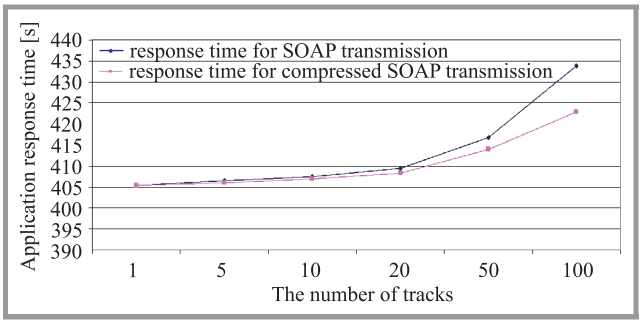

Fig. 6. Application response time for original SOAP transmission and with Gzip compression.

The true advantage of compression can be seen when the network is degraded with limited throughput, high error

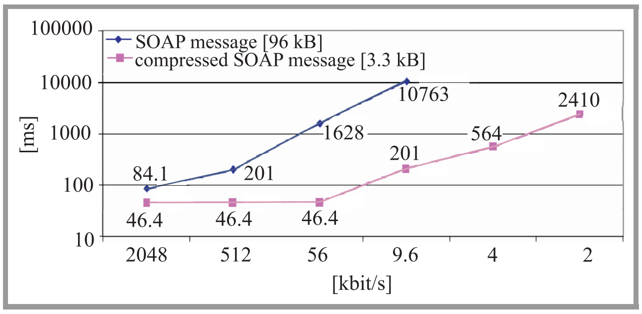

Fig. 7. Application response time in disadvantaged network for compressed and original 100 – track SOAP message. Network parameters: changing throughput (2048 kbit/s, 512 kbit/s, 56 kbit/s, 9.6 kbit/s, 4 kbit/s, 2 kbit/s), packet error rate = 0, delay = 0.

rates and high delays. The authors carried out tests of Gzip compression in an emulated disadvantaged environment, where links were configured by variable values of the throughput, delay and packet error rate (PER) (see Fig. 7, Table 1).

Table 1
Minimum throughput requirements at different values of per and delay

	PER [%]	Delay [ms]	Minimum throughput [kbit/s]
SOAP without compression	≤ 10	≤ 1000	9.6
	≥ 25	≥ 100	512
SOAP with compression	≤ 10	≤ 100	2
	≤ 10	> 100	4
	≥ 25	> 1000	56
	≥ 50	0	2048

In a network with very good quality parameters (PER = 0, delay = 0), an original 100 – track SOAP message could not be sent in a network with throughput equal 4 and 2 kbit/s. For 9.6 kbit/s, the application response time amounted to about 11 seconds, however, when the network parameters turn to degrade, this time is being extended (e.g., 134 s for delay = 100 ms, PER = 0). In comparison, a compressed SOAP message is transmitted in 2.56s for 9.6 kbit/s link, 5.62 s for 4 kbit/s link and in 28.8 s for 2kbps link, through links with the same QoS.

The results (see Table 1) showed that SOAP without compression can be sent through links not slower than 9.6 kbit/s, with PER $\leq$ 10% and delay $\leq$ 1000 ms. When PER $>$ 10%, the throughput must not lower than 512 kbit/s with a delay not lower than 100 ms. Compressed SOAP can be sent by 2 kbit/s links with PER $\leq$ 10% and delay $\leq$ 100 ms. If the delay is higher (over 100 ms), the throughput should amount to at least 4 kbit/s. When PER $>$ 10%, the lowest throughput is 56 kbit/s (delay $\leq$ 1000 ms). Sending compressed SOAP through a very bad link, where PER $\geq$ 50% can be done with the lowest throughput of 2048 kbit/s.

Apart from using the Gzip compression algorithm, it is possible to reduce the size of SOAP message with other mechanisms, including an additional efficient coding, like Efficient XML (EXI) [5], [6] or Fast Infoset.

Fast Infoset specifies a standardized binary encoding for the XML information set [7]. It uses the existing and proven ASN.1 standards. The specification is standardized as an ITU-T Recommendation within ITU-T Rec. X.891 [8] and ISO/IEC 24824-1 [9].

Efficient XML is a specification of binary coding of the XML data. EXI is a very compact representation of the XML information set that is intended to simultaneously optimize the performance and the utilization of computational resources. The EXI format uses a hybrid approach drawn from the information and formal language theories, plus practical techniques verified by measurements for entropy encoding of the XML information. To efficiently encode XML event streams, the EXI format uses a relatively simple algorithm, which is acquired for a fast and compact implementation, and a small set of data type representations. The EXI specification consists of the grammar production system and the format definition. EXI is compatible with XML at the XML information set level, rather than the XML syntax level. It permits to encapsulate an efficient alternative syntax and grammar for XML, while facilitating at least the potential for minimizing the impact on XML application interoperability. EXI is schema-“informed”, which allows utilizing the available schema information to improve the compactness and performance. It also uses a grammar-driven approach that achieves very efficient encodings.

Table 2
Compression gain and data processing time using Gzip, FI, EXi and EXI with compression for different message sizes

Compression mechanism	SOAP message size					
	0.6 kB		12.3 kB		406 kB	
	CG [%]	RTD [ms]	CG [%]	RTD [ms]	CG [%]	RTD [ms]
XML	–	2.2	–	2.8	–	9
Gzip	49.3	1.7	95.3	2.5	98.2	19.5
FastInfoset	30	1.9	53.5	3	55.2	11.9
EfficientXML	49.7	0.23	93.2	0.9	94.4	3
EfficientXML + compression	71.8	0.44	97.4	1.7	99	6.8

The subject of the tests was the efficiency of these compression techniques. The MCI team also investigated this problem in terms of using: an open source Gzip compression algorithm, a binary form of XML Fast Infoset and EXI (the EXIficient implementation[2]). The authors tested three different sizes of SOAP messages (0.6 kB, 12.3 kB

[2]More information can be found on EXIficient web site, http://exificient.sourceforge.net/

and 406 kB) in a simple web service, and measured the compression gain (CG) and the round-trip delay (RTD) time (see Table 2). The verification proved the high gain in compressing XML messages more efficient with large files. The most efficient compression mechanism is binary form of efficient XML, especially with compression, which, for big amounts of data, achieved compression gain equal 99%.

2.3. Image Resolution Change

Apart from XML, SOAP messages can easily carry image attachments. In the emulated environment, the authors also tested the efficiency of reducing JPEG images resolution. WANem application was used to emulate high error rates, delay and PER values. Three different JPEG images were sent (JPEG image with original size 486 kB and its two smaller equivalents –50% and 10% of it original size) (see Table 3).

Table 3
Minimum throughput requirements for images and their transmission time at different values of PER

JPEG (2706 × 3657)	PER [%]	Minimum throughput [kbit/s]	Transmission time [s]
JPEG 486 kB (100%)	0	56	80–100
JPEG 240 kB (50%)	≤ 10	9.6	250–299
JPEG 58 kB (10%)	≤ 10	9.6	62–87

The results show that it is possible to send still images in low throughput channels, however with significant value of the transmission time. Reducing their resolution can decrease this time (see Table 3, Fig. 8).

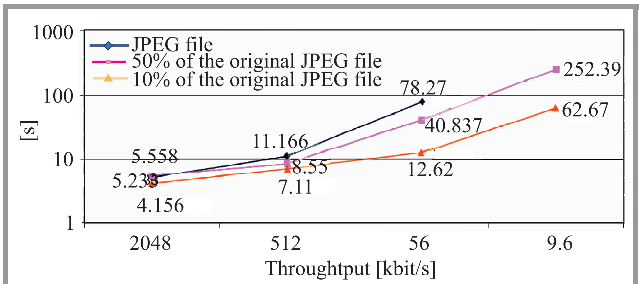

Fig. 8. Application response time for different image sizes. Network parameters: changing throughput (2048 kbit/s, 512 kbit/s, 56 kbit/s, 9.6 kbit/s), packet error rate = 0, delay = 0.

In the tests, reducing the image size was decreasing their resolution (see Fig. 9). However, image modifications can also include decreasing the colour depth and decreasing the image quality (inherent in the JPEG coding). Yet, changing the colour scale to greyscale in fact does not

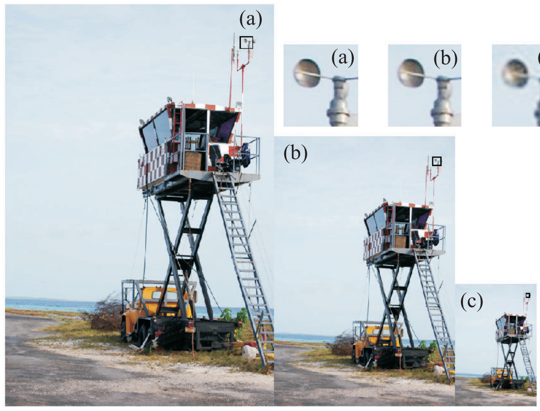

Fig. 9. Effect of reducing the image resolution for three dimensions 2706 × 3657 (a); 1913 × 2585 (b); 855 × 1156 (c).

Fig. 10. Changing the original JPEG colour depth to greyscale.

give much gain (673 kB of original size to 565 kB in the greyscale – see Fig. 10).
In contrast, the reduction of the JPEG image quality can give surprisingly good results. The most common and widely supported one is a lossy JPEG compression inherent to the nature of this image coding method. The degree

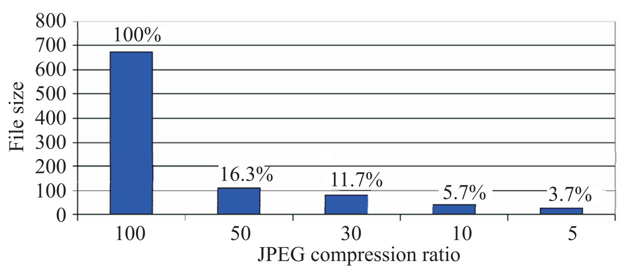

Fig. 11. JPEG compression ration versus image file size.

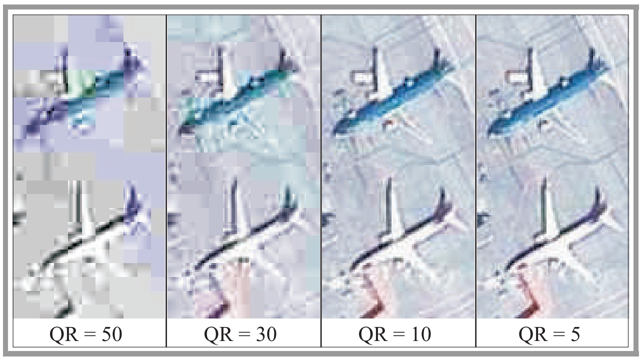

Fig. 12. Image quality reduction for a segment of an image and different values of JPEG quality ratio (QR).

of the compression can be adjusted, allowing a selectable tradeoff between the storage size and the image quality. In fact, decreasing the image quality, significantly reduces its size. Compression ratio equal 50 can reduce the size at over 83% of the original size (see Fig. 11), not degrading significantly the subjective image quality that can be assessed by the viewer. This is most visible when the user wants to zoom in a particular segment of the image where the details are most important (see Fig. 12).

3. Summary and Recommendations

The tests carried out by the authors proved that adaptation of the SOAP message by compression and modification of the image attachment (if present) can reduce the size of this message and improve its transmissions performance in a disadvantaged environment. The message size reduction can be achieved by the use of Gzip compression run in the application server at the HTTP level. This is a method very simple to use and widely supported by the application servers. As shown, compression is a time consuming process but its application during communication over links that have 56 kbit/s and below is highly recommended.

There should be also considered the application of binary coding techniques, especially the EXI, that provide good efficiency both in terms of compression gain and processing time. The measured web service response time for EXI was shorter than for Gzip, but the authors expected a greater difference for EXI solution. It may be caused by some deficiency in the used open source EXI implementation [9]. The measured compression ratio proved that EXI can significantly limit the size of SOAP messages (95% for medium messages), however, Gzip has also very good results (94% for medium messages). FI binary encoded messages gave worse results (53% for medium messages). The results proved that the Efficient XML standard is very promising, however, its efficiency strongly depends on the implementation.

Apart from tracking information, units located at the lowest command levels need recognition of the surrounding terrain. The sensor image services, though, need to be provided also through the disadvantaged tactical networks. The results of the tests provided in this article prove that sending high resolution big images through disadvantaged networks is related to a long transmission time (79 s for 56 kbit/s, impossible transmissions in 9.6 kbit/s network, almost 1 minute for the smallest file experimented). It is necessary to establish other methods for sending image sensor information through disadvantaged low bandwidth links. These can be fragmentation into smaller parts, or, e.g., decreasing the quality of the image. This latter one is very efficient, however only for a user that does not need to zoom in the image to see it in great details.

The presented results show the efficiency of selected methods that adapt the web services realization to the possibilities of the network. The original service producers can be equipped with such a functionality or the WS traffic can be redirected through a content proxy that would help to improve the communication at the lowest command levels. What remains to be done is the mechanism that would limit the impact of connection losses and other connectivity problems.

Acknowledgment

This work was carried out within the project PBZ – MNiSW – DBO-02/I/2007 supported by the R&D funds allocated for the years 2007-2010.

References

[1] M. Booth, T. Buckman, J. Busch, B. Caplan, B. Christiansen, R. van Engelshoven, K. Eckstein, G. Hallingstad, T. Halmai, P. Howland, V. Rodriguez-Herola, D. Kallgren, S. Onganer, R. Porta, C. Shawcross, P. Szczucki, and K. Veum, *Detailed Report Covering a Strategy and Roadmap for Realizing an NNEC Networking and Information Infrastructure*, vol. II of *NATO Network Enabled Feasibility Study*, version 2.0, NATO C3 Agency, June 2005.

[2] R. Haakseth, T. Gagnes, D. Hadzic, T. Hafsøe, F. T. Johnsen, K. Lund, and B. K. Reitan, "SOA – Cross Domain and Disadvantaged Grids", NATO CWID 2007, FFI-rapport 2007/02301, Norwegian Defence Research Establishment.

[3] G. Babakhani, J. Busch, C. Dumas, R. Fiske, B. Holden, H. Lægreid, R. Malewicz, D. Marco-Mompel, V. Rodriguez-Herola, "Web trends and technologies and NNEC core enterprise services, version 2.0", Technical Note 1143, NATO C3 Agency, The Hague, Dec. 2006.

[4] "Interim NFFI Standard For Interoperability of FTS", AC322(SC5)N(2006)0025, NC3B Information Systems SC, 16 Dec. 2006.

[5] M. Cokus, T. Kamiya, "Efficient XML Interchange Working Group Public Page" [Online]. Available: http://www.w3.org/XML/EXI/

[6] J. Schneider, T. Kamiya, "Efficient XML Interchange (EXI) Format 1.0", W3C Proposed Recommendation, 20 Jan. 2011 [Online]. Available: http://www.w3.org/TR/2011/PR-exi-20110120/

[7] J. Cowan, R. Tobin, "XML Information Set (Second edition)", W3C Recommendation, 4 Febr. 2004 [Online]. Available: http://www.w3.org/TR/xml-infoset/

[8] "Information technology – Generic applications of ASN.1: Fast Infoset", ITU-T Rec. X.891, May 2005.

[9] "Information technology – Generic applications of ASN.1: Fast Infoset", ISO/IEC 24824-1, May 2007.

[10] K. Lund *et al.*, "Using Web Services to Realize Service Oriented Architecture in Military Communication Networks", Norwegian Defence Research Establishment, *IEEE Commun. Mag.*, pp. 47–53, 2007.

Tomasz Podlasek was born in 1984 in Grójec, Poland. He graduated from the Faculty of Electronics and Telecommunications Technology of Military University of Technology (2009). At the moment he is a researcher in Military Communication Institute in Zegrze, Poland. He is working in the area of efficient realization of web services in disadvantaged networks. His main areas of interests are: Java programming, advanced programming technologies, data structures, semantic technologies and web services.

e-mail: t.podlasek@wil.waw.pl
Military Communication Institute
Warszawska st 22A
05-130 Zegrze, Poland

Joanna Śliwa, Marek Amanowicz – for biographies, see this issue, p. 53.