Computer Standards & Interfaces 36 (2014) 801–807

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Enabling powerful GUIs in ISOBUS networks by transparent data compression

Natalia Iglesias *, Pilar Bulacio, Elizabeth Tapia

CIFASIS-CONICET Institute, Bv. 27 de Febrero 210 Bis, Rosario, Argentina
Facultad de Cs. Exactas e Ingenieria, A. Pellegrini 250, National University of Rosario, Argentina

ARTICLE INFO

Article history:
Received 17 January 2013
Received in revised form 22 November 2013
Accepted 20 January 2014
Available online 29 January 2014

Keywords:
ISO11783
Lossless data compression
GZIP
Bus utilization
Object Pool

ABSTRACT

As the functionality of ISOBUS compliant agriculture machines increases, demands on the underlying bus network capacity increase as well. Therefore, to prevent potential bottleneck performance of critical applications, bus utilization must be carefully optimized. In this paper, a methodology for transparent compression/decompression of Object Pool files arising from the use of powerful GUIs during network initialization time is presented. Comprehensive simulation experiments developed under CANoe.ISO11783 shows that data compression remarkably reduces bus utilization during ISOBUS network initialization time, thus enabling the use of powerful GUIs. Furthermore, simulation results suggest GZIP as the best performing method for transparent ISOBUS data compression.

1. Introduction

The number of electronic components over agricultural equipment has increased significantly [3] during the last years. This process has generated the need of a standardized agricultural Binary Unit System (bus). A bus is a bandwidth limited communication resource shared by agricultural systems, including tractors, implements and farm computers from different manufacturers that is used for information transmission purposes [16]. The development of the bus concept in agriculture is accomplished by the ISO 11783 standard [8], best known as ISOBUS, which specifies a serial data network for control and communication on agricultural tractors and implements.

The ISOBUS standard somewhat resembles the Open System Interconnection (OSI) standard developed in the mid-1990s for the open interconnection of data networks [9]. Briefly, the purpose of ISOBUS is to standardize the method and format to data exchanging between specific electronic systems including sensors, actuators, control elements, information storage, and display units mounted on tractors or implements. An ISOBUS network [17] is built from a number of Electronic Control Units (ECUs) which are interconnected through a CAN 2.0b [2] bus, working at 250 Kbits/s. Among the connected ECUs, some of them are designated to carry out very specific tasks such as the Task Controller (TC) and Virtual Terminal (VT) units. The VTs have a graphical display with I/O interfaces to make possible the external operator interaction. The VT functionalities (described in part 6 of the ISO 11783 standard, hereafter ISO 11783-6) are specified by the behavior of a set of objects, each of them characterized by a set of specific attributes. In addition, the VT interface of any reachable ISOBUS device is defined through a set of objects called Object Pool (OP). Multiple ECUs might concurrently transmit multiple OPs to a VT during ISOBUS network initialization. A VT is intended to receive and store OP file in a modifiable memory area with the size and number of allowable OPs limited only by the Virtual Terminal available memory.

A highly bus demanding type of ISOBUS applications are those performing network initialization tasks. These applications usually require the simultaneous transmission of multiple OPs from each of the connected ECUs to the designated VT. The need for transmitting complex OPs embedded in large configuration files, e.g., to support powerful GUIs (Graphical User Interfaces), increases transmission delays [18] and thus, degrades the performance of network initialization applications [6]. The situation gets worse when the number of connected ECUs is increased.

To prevent bus network overload when transmitting bulky OPs, the ISO 11783-6 contemplates the use of Run-Length Encoding (RLE) compression [5] just for picture graphic data included in the OPs. The choice of RLE obeys to the possibility of accomplishing real-time decompression without the need of a VT buffer. However, as more and more complex OPs are introduced to support operator friendly interactions with highly specific agricultural implements, the need for generalized data compression to prevent ISOBUS network overload might become more ubiquitous. Although a proper use of data compression methods for ISOBUS networks might reduce both bus network load by a factor of two or more [1], and ECUs storage requirements by working directly on compressed OPs, no method for generalized OP file compression is actually considered in ISO 11783-6.

* Corresponding author.
E-mail address: iglesias@cifasis-conicet.gov.ar (N. Iglesias).

0920-5489/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.csi.2014.01.007

In this paper, a method for the transparent extension of ISO 11783-6 data compression techniques beyond the use of RLE over the graphical pictures contained into OP files is proposed. The method is intended to solve the critical bus utilization during ISOBUS network initialization by data compression of OP files. For this purpose, a decompression module is transparently introduced at the VT which remains in accordance with the ISO 11783-6 specification. Such decompression module performs two basic tasks: if a compressed OP file is received, then it is decompressed; if an uncompressed OP file is received, then it is bypassed and sent to the VT.

This paper is organized as follows. In Section 2, a solution for the transparent ISO 11783-6 compression/decompression of OP files is presented. In Section 3, a variety of compression techniques for OP files and their impact on bus utilization are evaluated using the CANoe.ISO11783 [19]. Finally, in Section 4, conclusions and further work are presented.

2. Transparent ISO 11783-6 data compression

Efficient bus utilization is an important factor for ISOBUS network performance. Data compression techniques can provide remarkable bandwidth savings when transmitting bulky OP files carrying powerful GUIs. Assuming that the ECUs and the VT agree upon the use of a fixed compression/decompression algorithm, a compression module and a decompression module must be introduced at each ECU and VT, respectively. The compression module must be able to compress OP files and signal the transmission of compressed OP files with the purpose of alert to VT about their decompression task. Similarly, a decompression module must be able to detect the signaling of the received compressed OP files and decompress accordingly. Regarding to signaling, as is suggested in [10] for the SAE J1939 protocol [14], the ECUs that are using OP file compression service should set to `1` its reserved `R` bit in the identification field (*ID*) of ISOBUS messages.

Note that the implementation of the compression and decompression modules should not be a complex task. However, to accomplish data compression benefits in a transparent way, i.e., keeping ISOBUS compliant data exchange, compressed OP files must be properly intercepted at the VT. For this reason, we focus on the design of the decompression module that should satisfy the following points: the modularity and simplicity of the computational architecture, and the ability of running multiple decompression processes due to many-to-one (ECUs-to-VT) communication model network. The result is a simple application called Transparent Decompression (*TD*) (Fig. 1).

The basic *TD* is composed by 7 entities: two entities that deal with reception and transmission of messages from/to ECUs called ECU Receiver (*ERx*) and ECU Transmitter (*ETx*), respectively; two entities that deal with reception and transmission of messages from/to VT called VT Receiver (*VRx*)and VT Transmitter (*VTx*), respectively; an entity that classifies the received OP files in compressed or uncompressed called Message Identifier Filter (*MIF*);an entity that manages the creation of the decompression process called Decompression Dispatcher (*DD*); and a default entity that performs the decompression process called Decompression Machine (*DM*). A complete activity diagram is shown in Fig. 2; the starting point is the reception of an OP file from an ECU to a VT:

Activity 1 The *TD* application is started when the *ECU Receiver entity* receives an OP file from some ECU on the network. Then, this OP file is sent to *Message Identifier Filter entity*.

Activity 2 The *Message Identifier Filter entity* determines if the OP file was compressed or not by a logical `AND` operation between the identification field of message and the constant value `0x800000h`. If the result is `1`, the compressed OP file is sent to *Decompression Dispatcher entity*, otherwise the uncompressed OP file is directly sent to the VT.

Activity 3 *The Decompression Dispatcher entity* receives just the compressed OP file and instances the *Decompression Machine entity* to make the decompression. By default, just one *Decompression Machine entity* exists at the *TD* with two possible operational states: *listening* or *decompressing*. Note that multiple decompression requests (n) may occur simultaneously if multiple compressed OP files are received. In this case, the *Decompression Dispatcher entity* has to create multiple instances of *Decompression Machine entity* if the default one is already in use (state decompressing).

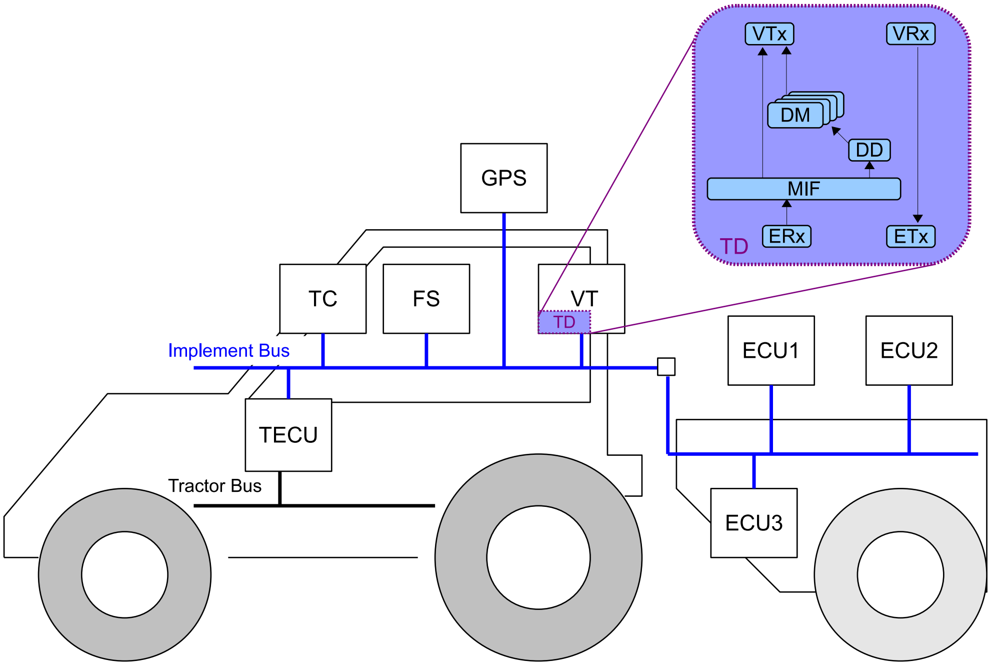


Fig. 1. Schematic form of a simplified ISO 11783 network. The TD application serves as an interface between the general VT functionalities and the ISOBUS network. The TD application is made up of several entities, an ECU Transmitter (*ETx*) and an ECU Receiver (*ERx*) pair, a VT Transmitter (*VTx*) and a VT Receiver (*VRx*) pair, a Message Identifier Filter (*MIF*), a Decompression Dispatcher (*DD*) and a Decompression Machine (*DM*).

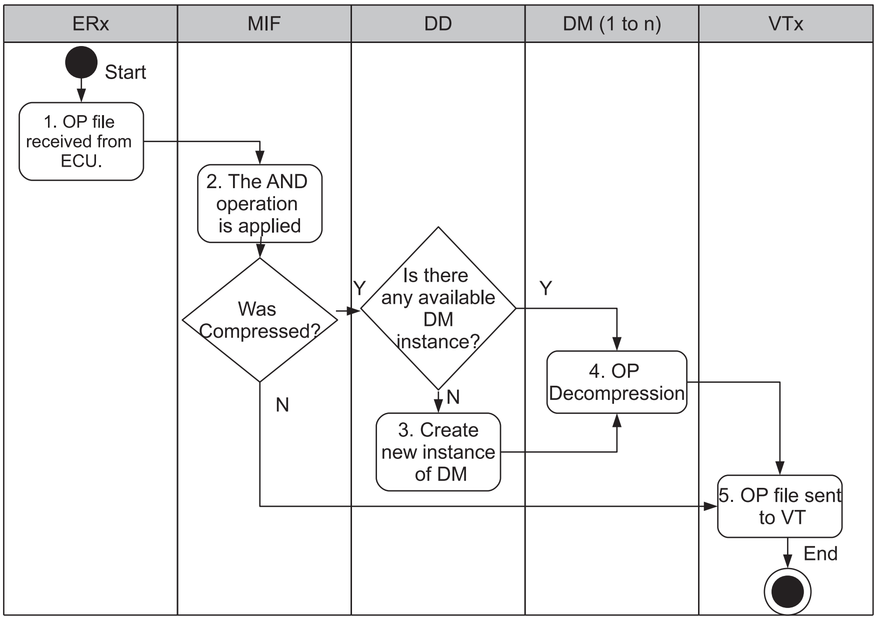

ERx
MIF
DD
DM (1 to n)
VTx
Start
1. OP file received from ECU.
2. The AND operation is applied
Was Compressed?
Y
N
Is there any available DM instance?
Y
N
3. Create new instance of DM
4. OP Decompression
5. OP file sent to VT
End

Fig. 2. *TD* application activity diagram. Several entities are involved, an ECU Receiver (*ERx*), a Message Identifier Filter (*MIF*), a Decompression Dispatcher (*DD*), a Decompression Machine (*DM*) and a VT Transmitter (*VTx*).

Activity 4 The *Decompression Machine entity* make the OP file decompression, changing from listening to compressing state during this task.

Activity 5 The uncompressed OP files are sent to the VT by the *VT Transmitter entity*.

In the next section, a comprehensive study about the performance of data compression techniques on typical OP files is performed. This study is intended for the selection of the best method for OP file compression and the implementation of the *TD* module.

3. Experiments and analysis

3.1. Experimental setup

Taking into account that ISOBUS network initialization with powerful GUIs might lead to critical bus utilization when the number of connected ECUs is increased, three ISOBUS network configuration scenarios (*T1*, *T3*, and *T5*) were considered. These scenarios were categorized by considering the number of connected ECUs (1, 3 and 5) respectively. In all these scenarios, the simultaneous transmission of OP files from the set of connected ECUs to a unique VT unit enhanced with the proposed TD module was assumed. Networking scenarios were implemented using the CANoe.ISO11783 for the analysis and simulation of complex ISO 11783 communication structures. Simulations were performed on a machine with the specifications shown in Table 1.

To gain insight into ISOBUS network utilization arising from the use of powerful GUIs, the transmission of OP files from an ad-hoc benchmark dataset was considered (see Table 2). This benchmark dataset, was constructed following the requirements of the ISOBUS Compliance Test Protocol [11] for VT certifications. For this purpose, first we select two OP files from the set of examples provided in the CANoe.ISO11783. These files contained a simple GUI for a *sprayer* implement (Sprayer.iop) and a naive GUI design with a single image (ISOBUSMask.iop). The third file, which was generated with the *vt-designer* software tool [12], contains an advanced GUI for a *planting* implement. Such GUI aims to show the state of 48 nozzles of a planting implement and the corresponding seed map in real time (map10.iop). A detailed description of IOP benchmark datasets is shown in Table 3.

To identify the most effective data compression technique for OP file, four lossless compression methods widely used in literature, were evaluated: Huffman [7], RLE [5], LZW [20] and GZIP [4]. For this purpose, each compression method was characterized by its *compression ratio C* defined as the relation between the output and input file size, and its *relative decompression time D*, defined as the relation between the decompression time using any algorithm and using a reference algorithm [15], e.g., GZIP. Note that the evaluation of this latter relative measure avoids undesirable dependencies on specific processor capacities and available resources. The criterion of the best compression algorithm selection was through the lowest *C* and *D* measures on IOP benchmark datasets.

Simulation of ISOBUS network initialization was accomplished with the set of messages shown in Table 4. Message priorities in the range of 6 to 7 were set in accordance with ISO 11783. In addition, response times were set to their default values to avoid timeout events due to the bus access latencies or messages forwarding. Hence, ECUs expecting a response must wait at least 1250 ms before giving up or retrying. In addition, ECUs when required to provide a response must do so within 200 ms.

To ensure the correct transmission of either uncompressed or compressed OP files, a bus sniffer available in the CANoe.ISO11783 was used to monitor and collect ISOBUS messages during network initialization. The bus sniffer allowed us to check the following protocol issues: the address claim procedure of each node, the technical data

Table 1
Setup details of the resources environment.

Resources	Details
Operating system	Microsoft Windows XP Professional SP 3
CPU	Intel Core(™)2 Duo E6750 CPU @2.66 GHz
Hard disk	Hitachi HDS721616PLA320 150 GB
RAM	2 GB
C compiler	Microsoft Visual C++ 2010 Express version 10.0.30319.1
Simulation environment	Soft: CANoe.ISO11783 version 7.6.91 (SP5) Hard: CANcaseXL

Table 2
The IOP benchmark datasets.

Name	Size [byte]	Description
Sprayer.iop	3037	Simple GUI for a sprayer implement
ISOBUSMask.iop	16,895	Naive GUI design with a single image
map10.iop	165,793	Advanced GUI for a planting implement

Table 3
The structure of IOP benchmark datasets in terms of the number of objects required for their definition. Only objects found in at least one benchmark file are shown. Objects are grouped by classes in agreement with ISO 11783:6.

Object	Sprayer	ISOBUSMask	map10
Working set	1	1	1
Data mask	2	1	2
Alarm mask	2	–	–
Container	10	–	23
Soft key mask	3	–	2
Key	4	–	5
Input number field	1	–	–
Input list field	1	–	–
Output string field	18	1	24
Output number field	12	–	8
Output line shape	3	–	2
Output rectangle shape	2	–	729
Output ellipse shape	–	–	2
Output meter graphic	–	–	1
Output liner bar graphic	2	–	49
Output graphic context	–	–	2
Picture graphic	4	2	6
Number variable	14	–	51
Font attribute	4	1	5
Line attribute	3	–	2
Fill attribute	2	–	729
Aux control function type 1	4	–	–

messages used to request the characteristics of the VT before transmission of uncompressed or compressed OP files, the normal handshaking, error checking, and retransmission procedures. The bus sniffer was then used to measure OP file compression benefits on bus utilization, time between messages transmitted *T* and overall ECUs initialization times *I*. Regarding bus utilization, its empirical value extitU, defined as the ratio of the actual to the maximum number of bits allowed in the bus during a given period of time, was measured. For practical purposes, observation periods were set to multiples of 1s.

Finally, the benefits of data compression under increasing bus loads *L* were also evaluated. For this purpose, relative OP file transmission times T^* were measured. Briefly, T^* is defined as the relation between the (compressed or uncompressed) OP file transmission time on a bus with any bus load *L* and the uncompressed OP file transmission time over a dedicated bus. For the sake of simplicity, only the transmission of the largest IOP benchmark datasets, i.e. the map10.iop file, in the *T1* networking scenario was considered. *L* values in the range from 0 to 95% of the maximum of 250 Kbps were considered. Note that bus loads *L* was normalized with respect to fully occupied situation, i.e., $L = 60\%$ means a bus transmitting just 60% of the time. To manipulate *L*, a dummy ISOBUS message was first generated using the generator block tool of the CANoe.ISO11783 environment. Thus, by changing the generation period of such dummy message, different *L* settings were simulated. For example, for a message *m* carrying $^{S}{}_{m} = 150$ bits with a bit duration $\tau = 0.004$ ms, a message generation period $p_m = 1$ ms leads to $L = \frac{s_m \cdot \tau}{p_m} = 0.6\ (60\%)$.

Table 4
ISOBUS messages used during the initialization. The first column is the Parameter Group Number (PGN) of each message as defined in the ISO 11783 standard. The second column shows the acronym name of each message. The third column shows message priority levels, being 0 the highest priority level and 7 the lowest one. Finally, the last column is the Data Length Code (DLC) of each message, i.e., the number of data bytes contained in each message.

PGN	Message	Priority	DLC (Bytes)
EE00	ACL	6	8
EA00	RQST	6	3
FE0D	WSMSTR	7	8
FE0F	LC	6	8
E600	VT to ECU	7	8
E700	ECU to VT	7	8
EC00	TP.CM	7	8
EB00	TP.DT	7	8
C800	ETP.CM	7	8
C700	ETP.DT	7	8

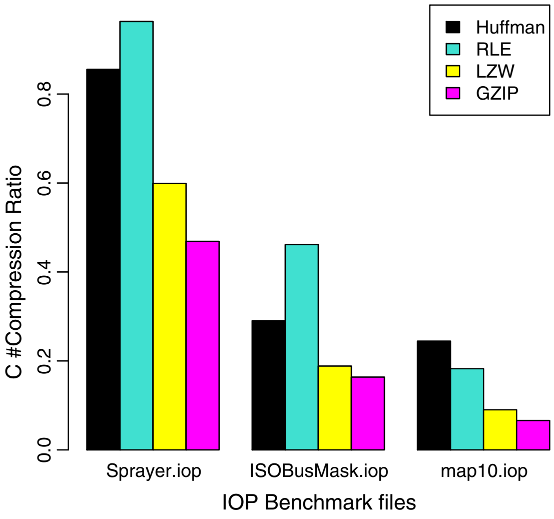


Fig. 3. Compression ratios *C*. Four lossless compression methods (Huffman, RLE, LZW, and GZIP) using IOP benchmark datasets are compared. GZIP exhibits the best compression performance.

3.2. Results analysis

GZIP data compression exhibited the lowest compression ratios *C* on IOP benchmark datasets allowing the reduction of OP file sizes by a factor from 2 to 15 with *C* in the range from 0.066 to 0.46 (see Fig. 3). Similarly, for the largest OP file in the IOP benchmark datasets, the map10.iop file, remarkable higher *D* values were observed for all other candidate compression methods (see Fig. 4). Hence, GZIP was selected to measure the benefits on OP file compression on the empirical bus utilization parameter *U* when increasingly complex networking scenarios *T1*, *T3* and *T5* considered.

As expected, empirical bus utilization *U* was remarkably reduced by the use of OP file compression techniques and the TD application. Average *U* reduction factors ranging from 2 to 15 were observed in the set of IOP benchmark datasets when considering increasingly complex networking scenarios (see Figs. 5 to 7).

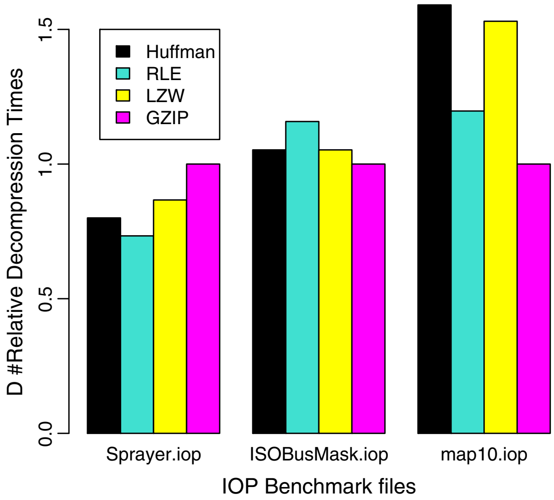


Fig. 4. *Relative Decompression time D*. The absolutes decompression times are normalized with respect to GZIP. Thus, four lossless compression methods (Huffman, RLE, LZW, and GZIP) using IOP benchmark datasets are compared. The results show that, e.g. in the case of the map10.iop file compressed with the Huffman method, *D* has a value of 1.59 this means that it takes 1.59 times longer than using GZIP.

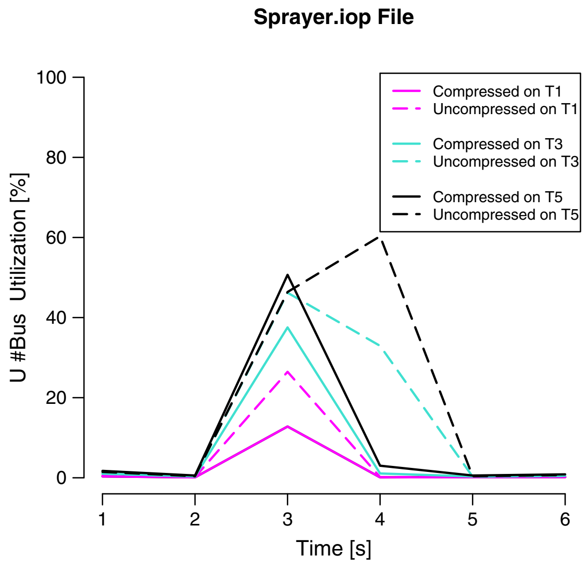


Fig. 5. Empirical bus utilization *U* with (—) and without (— — —) data compression for the Sprayer.iop file on increasingly complex networking scenarios *T1*, *T3* and *T5*. *U* without data compression is twice that with data compression.

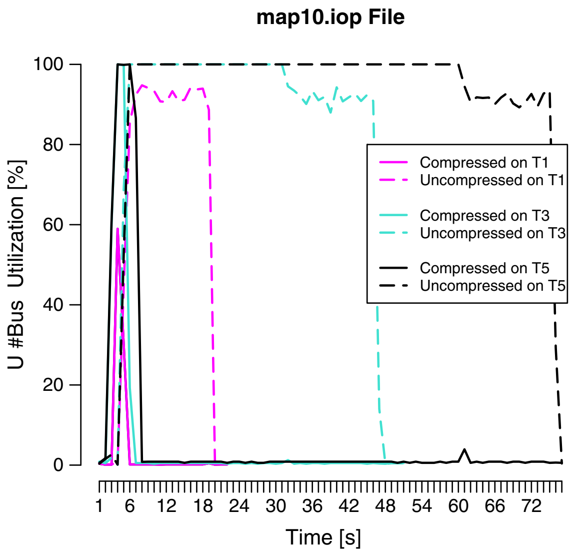


Fig. 7. Empirical bus utilization *U* with (—) and without (– – –) data compression for the map10.iop file on increasingly complex networking scenarios *T1*, *T3* and *T5*. *U* without data compression is 15 times that with data compression.

Finally, roughly invariant time between messages transmitted *T* (see Table 5) and initialization times *I* (see Fig. 8) were observed when using GZIP data compression and the TD application on increasingly complex networking scenarios. On the other hand, both *T* and *I* dramatically increased when no data compression was used and increasingly larger IOP benchmark datasets or increasingly complex networking scenarios were considered. In particular, *T* values in the range from 0.56 to 0.58 ms with a standard deviation in the range from 0.13 to 0.19 ms were observed when data compression was used. On the other hand, average *T* values were roughly doubled and their standard deviation remarkably augmented when no data compression method was used.

Increasing bus loads on the T1 networking scenario leads to relative OP file transmission times of exponential growth (see Fig. 9). Under a typical bus load $L = 60\%$, the T^* accomplished for the uncompressed transmission of the map10.iop file was 2.5. In other words, OP file transmission time without data compression at $L = 60\%$ was 2.5 times those over a dedicated bus. On the other hand, the T^* accomplished for the compressed map10.iop file was 0.15, i.e., OP file transmission time with data compression at $L = 60\%$ was roughly 6.6 times lower than on a dedicated bus without data compression.

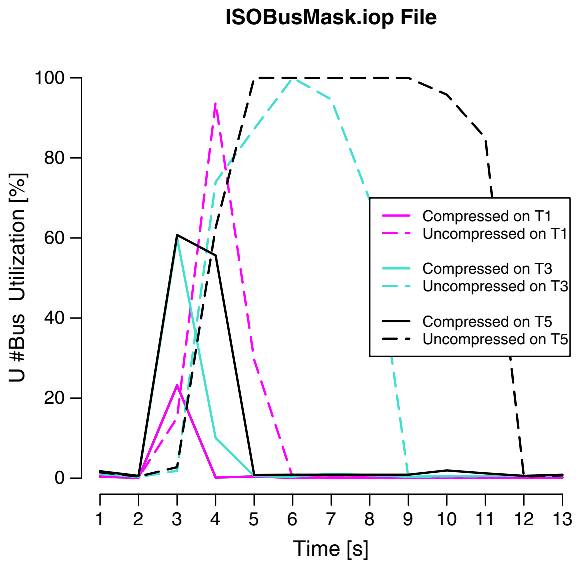


Fig. 6. Empirical bus utilization *U* with (—) and without (– – –) data compression for the ISOBUSMask.iop file on increasingly complex networking scenarios *T1*, *T3* and *T5*. *U* without data compression is 6 times that with data compression.

Regarding initialization times *I*, we note that since we are assuming the *T1* networking scenario, *I* can be easily estimated from *U* temporal profiles by observing the width of *U* pulses (see Fig. 10). Hence, at $L = 60\%$, the *I* for the map10.iop file with data compression turns to be roughly 15 times that without it. Remarkably, roughly the same relative improvement factor on *I* attributable to OP file compression was observed for the map10.iop file when transmitting over dedicated bus. Overall, these results suggest that provided *L* remains moderate, OP file compression benefits on ISOBUS network initialization times *I* remain valid.

4. Conclusions

In this paper, a proposal for enabling the use of powerful GUIs in ISOBUS networks was presented. The proposal allows a remarkable reduction of bus utilization and initialization times by the straightforward compression of large OP files. Among popular compression techniques, the GZIP was identified as the best choice for compressing OP files containing powerful GUIs. Practically, the proposed compression system can be deployed as a simple decompression module application resident at the designated VT unit. The decompression module can be designed to ensure compatibility with ISOBUS legacy systems by introducing a simple compression signaling protocol based on the utilization of reserved R bit of ISOBUS messages.

Although in this paper we have restricted ourselves to the static selection of compression techniques, dynamic selection approaches

Table 5
The mean and standard deviation (sd) of time between message transmission *T* accomplished when transmitting IOP benchmark datasets with and without data compression on increasingly complex networking scenarios *T1*, *T3* and *T5*.

File	Processing	T on T1		T on T3		T on T5	
		Mean	sd	Mean	sd	Mean	sd
Sprayer.iop	Compressed	0.57	0.13	0.57	0.13	0.58	0.15
	Uncompressed	0.60	0.29	1.00	4.48	1.19	7.36
ISOBUSMask.iop	Compressed	0.56	0.14	0.56	0.14	0.56	0.15
	Uncompressed	0.60	0.42	0.95	3.74	1.64	26.30
map10.iop	Compressed	0.56	0.14	0.56	0.19	0.56	0.15
	Uncompressed	0.61	1.27	1.32	34.87	1.37	29.27

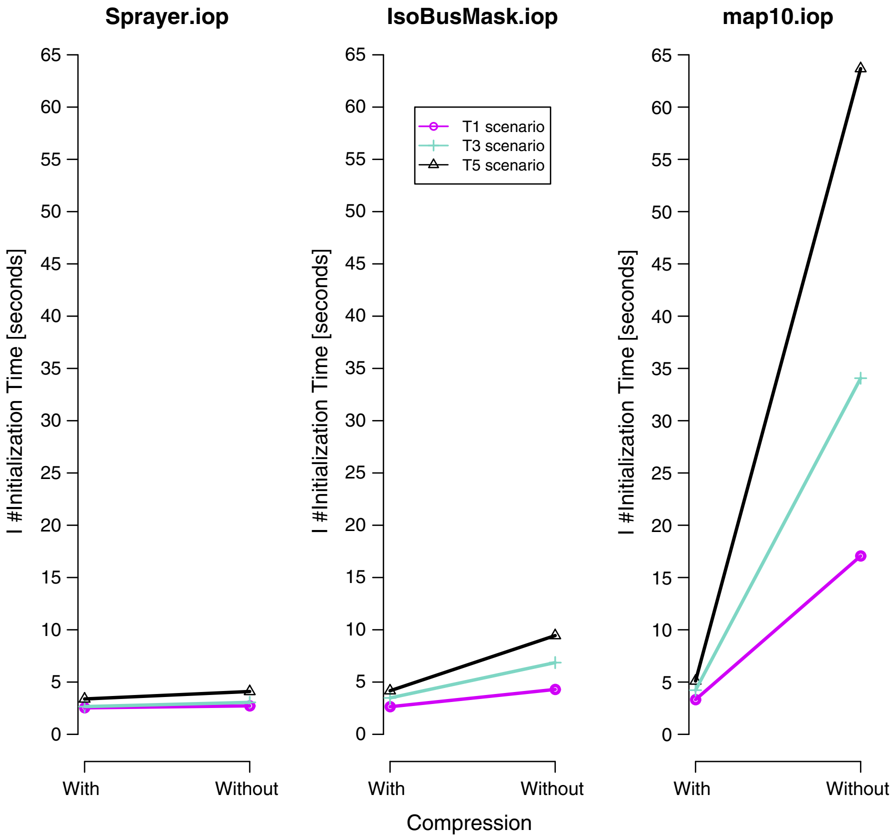


Fig. 8. Initialization times *I* with and without data compression for IOP benchmark datasets on increasingly complex networking scenarios. *I* is remarkably reduced when data compression is used.

might be alternatively considered. In such case, the specific compression technique might be dynamically negotiated between the VT and the ECUs before effective data transmission. Furthermore, the specific machine code for the selected data decompression algorithm might be provided and transmitted by the ECU intending the transmission of compressed OP files. A candidate protocol for implementing the dynamic selection of data compression techniques for the compression of OP files is the Signaling Compression (SigComp) [13] developed by the IETF. The interesting point of SigComp is that the compression algorithm is not specified. This means that the choice is left to the implementer. This is possible because the core of SigComp is the Universal Decompressor Virtual Machine (UDVM) which receives by messages the bytecode and the data dictionary of the decompression algorithm used from the sender. Since there is no need to manually check

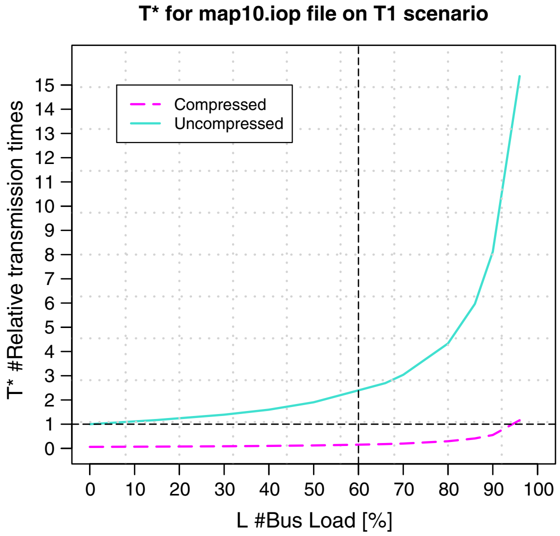


Fig. 9. Relative OP file transmission times T^* with and without data compression for the map10.iop file on the *T1* networking scenario for variable bus loads *L*. Increasing *L* settings lead to relative transmission times T^* of exponential growth. The vertical dashed lines indicate a $L = 60\%$. The horizontal dashed line indicates the reference value 1 used in our analysis (i.e. absolute transmission time normalized with respect to transmission time of the uncompressed file on a dedicated bus).

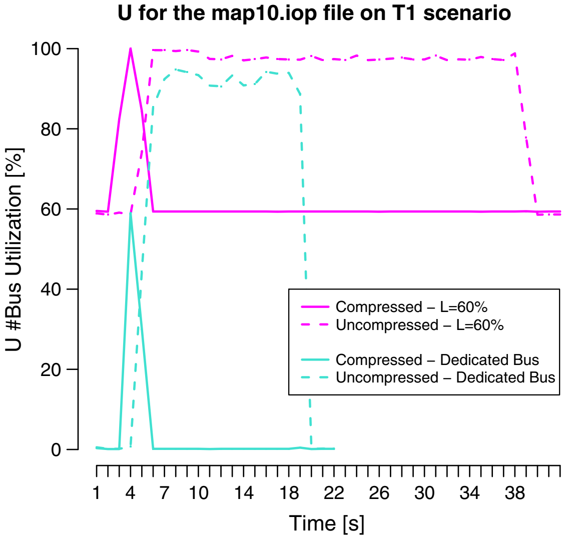


Fig. 10. Bus utilization *U* with (—) and without (– – –) compression for the map10.iop file on the *T1* networking scenario. Transmissions over a dedicated bus and a bus loaded already loaded with $L = 60\%$ are considered. *U* without data compression turns to be roughly 15 times that with it.

compression and decompression compatibility issues, dynamic selection of data compression techniques appears to be the ideal solution for the flexible compression of OP files.

The proposed philosophy of data compression for OP files can be easily extended to other types of files used in the ISO 11783, including device description XML files containing all information for a device to perform a task or the definition of the task itself to be performed by a device. We thus conclude that generalized data compression of key ISO 11783 data files might be of a great value for sustaining current technology innovation in tractors and agricultural implements.

Acknowledgment

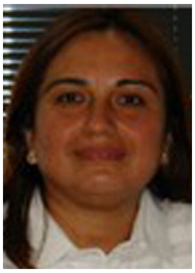
This work was supported in part by the Agencia Nacional de Promoción Científica y Tecnológica under Project PICT PRH No 0253 and by the CIFASIS-CONICET, Argentina.

References

[1] T. Bell, K. Pawlikowski, The effect of data compression on packet sizes in data communication systems. Teletraffic and traffic, 1991. 551/556.
[2] R. Bosch, CAN Specification Version 2.0, 1991.
[3] Cox, Information technology: the global key to precision agriculture and sustainability, Comput. Electron. Agric. 36 (2002) 93/111.
[4] P. Deutsch, GZIP File Format Specification Version 4.3. RFC 1952 (Informational), 1996.
[5] S. Golomb, Run-length encodings, IEEE Trans. Inf. Theory 12 (1966) 399–401.
[6] T. Herpel, K.S. Hielscher, U. Klehmet, R. German, Stochastic and deterministic performance evaluation of automotive can communication, Comput. Netw. 53 (2009) 1171–1185, http://dx.doi.org/10.1016/j.comnet.2009.02.008(performance Modeling of Computer Networks: Special Issue in Memory of Dr. Gunter Bolch).
[7] D. Huffman, A method for the construction of minimum-redundancy codes, Proc. IRE 40 (1952) 1098–1101, http://dx.doi.org/10.1109/JRPROC.1952.273898.
[8] ISO11783, Tractors and Machinery for Agriculture and Forestry — Serial Control and Communications Data Network, 2007.
[9] ISO7498, Open Systems Interconnection Basic Reference Model: The basic Model, International Organization for Standardization, 1994.
[10] S. Misbahuddin, S. Mahmud, N. Al-Holou, Development and performance analysis of a data-reduction algorithm for automotive multiplexing, IEEE Trans. Veh. Technol. 50 (2001) 162–169, http://dx.doi.org/10.1109/25.917911.
[11] NAIITF, ISOBUS Compliance Test Protocol. Version 1.0, 2005.
[12] OSB-AG, VT-Designer: Outperforming Mask Design. Version 1.4, 2012.
[13] R. Price, C. Bormann, J. Christoffersson, H. Hannu, Z. Liu, J. Rosenberg, Signaling Compression (SigComp). RFC 3320 (Proposed Standard). Updated by RFC 4896, 2003.
[14] SAE, SAE J1939 (1998), Recommended Practice for a Serial Control and Communication Vehicle Network, 1998.
[15] S. Sakr, Xml compression techniques: a survey and comparison, J. Comput. Syst. Sci. 75 (2009) 303–322.
[16] H. Speckmann, G. Jahns, Development and application of an agricultural bus for data transfer, Comput. Electron. Agric. 23 (1999) 219/237.
[17] M. Stone, K. McKee, C. Formwalt, R. Benneweis, Iso 11783: An Electronic Communications Protocol for Agricultural Equipment, Agricultural Equipment Technology Conference, ASAE, Louisville, Kentucky, 1999, pp. 1–17.
[18] K. Tindell, A. Burns, Guaranteed Message Latencies for Distributed Safety-critical Hard Real-time Networks, Technical report, University of York, Department of Computer Science, 1994.
[19] Vector-Informatik, CANoe.ISO11783 Ver 7.6, 2011.
[20] T. Welch, A technique for high-performance data compression, Comput. 17 (1984) 8–19, http://dx.doi.org/10.1109/MC.1984.1659158.

Natalia Iglesias is a Ph.D. student in the University National of Rosario in Argentina. She received an Electronic Engineering degree in 2007 from the same university. Since 2009 she is working as scholarship holder at the Research Institute CIFASIS of Rosario. Her main research interests are ISO 11783 networks, and sensor networks in precision agriculture.

Pilar Bulacio received an Electronic Engineering degree in 1998 from the University National of Rosario in Argentina, and a Ph.D. degree in Telematics in 2006 from the Technical University of Madrid (UPM), Spain. She worked as an engineer at SIEMENS from 1998 to 2001. Since 2007 she is working as a research and Associate Professor at National University of Rosario, and at CIFASIS. Her main research interest is the selection of variables with joint measures for biological data processing and spectroscopy.



Elizabeth Tapia is an Associate Professor at the University National of Rosario. She received an Electronic Engineering degree in 1990 from the University National of Rosario in Argentina, and a Ph.D. degree in Telematics in 2001, from the Technical University of Madrid (UPM), Spain. Currently, she is the Head of the Information Sciences Division at CIFASIS. Her current work focuses on i) Communications: LDPC coding; ii) Computational Learning: Multi-classifiers and combining classifiers; iii) Bioinformatics: Sequence analysis, classification of microarray data, barcodes for multiplexing NGS; and iv) AgroInformatic: Sensor Networks in Precision Agriculture.