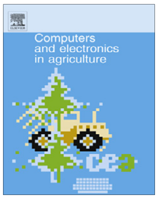
Computers and Electronics in Agriculture 109 (2014) 101–108

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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



Improving ISO 11783 file transfers into mobile farm equipments using on-the-fly data compression

Natalia Iglesias *, Pilar Bulacio, Elizabeth Tapia

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

ARTICLE INFO

Article history:
Received 28 February 2014
Received in revised form 12 August 2014
Accepted 19 September 2014

Keywords:
ISOBUS
Lossless data compression
Bus utilization
Object pool

ABSTRACT

It is considered the bandwidth bottleneck problem arising in ISO 11783 networks of mobile farm equipments when large file transfers are performed. To overcome this problem, a compression protocol called ISOBUSComp allowing the implementation of dynamic ("on the fly") data compression services for general ISO11783 file transfers is proposed. As a result, transmitting Electronic Control Units are free to choose any data compression technique they wish and receiving Electronic Control Units need not to be aware of such decisions, but just be able to process a suitable Universal Decompression Virtual Machine. Comprehensive simulation studies show that dynamic data compression services built upon the proposed protocol help to reduce bus utilization of ISO 11783 networks between a 28% and a 63%, thus speeding up the time for large file transfers.

1. Introduction

Over the last decades, a great number of technologies have been introduced to agricultural machines (Vanclay et al., 2013); from embedded systems to key networking technologies (Munack and Speckmann, 2001), more and more electronics continue to be added (Pereira et al., 2010; Wang et al., 2007; Cox, 2002). At the mobile farm equipment level, specific local agricultural networks defined by the ISO 11783 set of standards (ISO11783, 2007) are recommended (Backman et al., 2013). Taking into account that standardization and integration of information are dominant attributes in precision agriculture (Suprem et al., 2013; Aubert et al., 2012), the value of a communication standard like the ISO 11783 is increasingly recognized. It is observed, however, that ISO 11783 functionalities may require suitable extensions to deal with the increasing need of transmitting data of exponential growth (Egorova et al., 2007). For example, in mobile farm equipments supplied with different sensors able to collect large amounts of data while working (Peets et al., 2012; Nash et al., 2009; Steinberger et al., 2009), data gathering may cause undesirable bandwidth bottlenecks. Since bandwidth is not a uniformly available resource in agricultural infrastructure (Lee et al., 2012; Herpel et al., 2009), efficient, dependable and high-speed communication systems are crucial to provide almost real-time performance and thus, to avoid system malfunctioning (Mainoo, 2012).

Nowadays, the efficient management of network bandwidth is a top-of-mind in all areas of life (Passos et al., 2013; Kim and Lee, 2014). In the agricultural context, this issue was first envisioned by Stone (1994) who proposed the use of agricultural networks based on buses of higher bandwidth or multiple buses (Stone et al., 1999). It is observed, however, that these hardware solutions are not straightforward applicable to the ISO 11783 standard. In this regard, an alternative option is the development of applications to reduce the amount of data to be transferred, similar to those used for in-vehicle networks (Miucic et al., 2009; Ramteke and Mahmud, 2005), e.g., by the utilization of data compression methods.

Data compression is a well-known technique to improve network performance of low speed networks. In recent work (Iglesias et al., 2014), it was shown how data compression methods can be used to alleviate bandwidth bottleneck problems of ISO 11783 networks at initialization time. It is observed, however, that in such proposal, the provision of static data compression services is required, i.e., receiving Electronic Control Units (ECUs) have to know in advance the specific data compression method used by transmitting ECUs.

In this paper, built upon our previous work, a data compression protocol allowing the development of dynamic data compression services for ISO 11783 networks called ISOBUSComp is introduced.

---

* Corresponding author at: CIFASIS-CONICET Institute, Bv. 27 de Febrero 210 Bis, Rosario, Argentina.

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

http://dx.doi.org/10.1016/j.compag.2014.09.012
0168-1699/© 2014 Elsevier B.V. All rights reserved.

As a result, data compression methods can be independently selected at transmitting ECUs and receiving ECUs need not to be aware of such decisions. This unique feature provides a great flexibility in the design and operation of data compression services for general ISO 11783 file transfers.

This paper is organized as follows. In Section 2, a proposal for the implementation of dynamic data compression services in ISO 11783 networks is presented. In Section 3 and Section 4, the impact of such proposal in ISO 11783 bus network performance is evaluated using the CANoe.ISO11783 framework (Vector-Informatik, 2011). Finally, in Section 5, conclusions are presented.

2. On-the-fly data compression for ISO 11783 networks

Over the last decade, with the aim of boosting productivity in sustainable way (Blank et al., 2013), data management has become a topic of high interest in the field of agricultural engineering. On the other hand, the ISO 11783 standard has been originally designed just for the periodic transmission of small amounts of control data (Oksanen, 2010; Godoy et al., 2010). Hence, new methods are required to allow the effective and efficient use of ISO 11783 networks for general data sharing purposes. In this context ISOBUSComp is introduced. ISOBUSComp is a compression protocol designed to transparently enhance the current ISO 11783 standard with a common data compression service interface between applications and transport protocols.

Briefly, this proposal is inspired in the design of the Signaling Compression Protocol (SigComp) (Du et al., 2013; Wu et al., 2010; Bin et al., 2006; Jin and Mahendran, 2005; Price et al., 2003). SigComp is a widely used communication protocol originally designed by the Internet Engineering Task Force to alleviate the signaling overhead required for the provision of Voice over IP (VoIP) services in mobile networks (Rosenberg et al., 2002). SigComp essentially embodies a method for compressing text-based communication data used by VoIP signaling protocols. Similarly to SigComp, the core of ISOBUSComp is a Universal Decompression Virtual Machine (UDVM), a virtual machine[1] like the Java Virtual Machine (JVM) (Downing and Meyer, 1997) optimized for running decompression algorithms. Using a UDVM allows ISO 11783 implementers to select any compression algorithms they wish: compressed data at the transmission side will combine with a set of UDVM instructions (bytecodes) allowing original data to be extracted at the destination side.

2.1. ISOBUSComp architecture

ISOBUSComp protocol architecture is built upon two modules, the Compressor and Decompressor ones (Fig. 1). The Compressor Module is composed of a *Compression Dispatcher* and a *Compression Machine*; the Decompressor Module is composed of a *Decompressor Dispatcher* and a *UDVM unit*. If the application decides a compression, the Compressor Machine joint with the Compressor Dispatcher do the application message by compressing and attaching the *bytecodes* of the specific decompression algorithm. At the receiving ECU, the Decompressor Module decompresses the application message by interpreting received *bytecodes* with the resident UDVM.

For the sake of better understanding, it is consider the example where the Application is the Object Transport Protocol dealing with the transmission of the Object Pool (OP) files during ISO 11783 network initialization. The ISOBUSComp protocol starts with the requirement of an OP file transfer by the Object Transport Protocol (see Fig. 2) from a transmitting ECU to a designated receiving ECU, in this case a Virtual Terminal (VT[2]) unit.

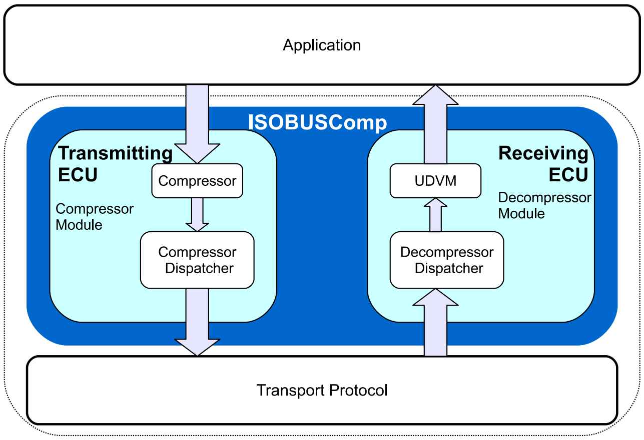


Fig. 1. The ISOBUSComp protocol architecture in ISO 11783 network. A Compressor Module is located at transmitting ECU; a Decompressor Module is located at receiving ECU. Both the Compressor and Decompressor modules provide data compression services to general ISO 11783 applications.

The following activities are made by the **Compressor Module**:

Activity 1: The compression process is initiated by the Object Transport Protocol when an OP transfer is required.
Activity 2: The OP file is compressed by the *Compressor Machine* using a compression algorithm.
Activity 3: The *Compressor Dispatcher* receives the compressed OP file from the *Compressor Machine* and builds an *Object Pool Transfer Message* with the following information: length and location of bytecodes, the bytecodes, and the compressed OP file. Note that ISO 11783 does not specify compression techniques at any layer, and thus, a reserved bit for future applications of the ISO 11783 frame (R bit) is used to signal the use of ISOBUSComp data compression services (Iglesias et al., 2014).

The following activities are made by the **Decompressor Module**:

Activity 1: The decompression process is initialized when an *Object Pool Transfer Message* from an ECU is received.
Activity 2: The *Decompressor Dispatcher* checks R bit. If $R = 0$, then OP file is directly sent to the VT. Otherwise, the message is split into *bytecodes* and compressed OP data; both parts are send to the *UDVM unit*.
Activity 3: The *UDVM* unit receives the *bytecodes* and the compressed OP data to make possible the decompression. Finally, the OP file is sent to the VT.

2.2. Implementation details

A proof of concept of the ISOBUSComp protocol was developed using an adaptation of the open source *libsigcomp*[3] library. For convenience of simplicity, the DEFLATE compression algorithm (Deutsch, 1996), already included in *libsigcomp*, was used. Since DEFLATE is closely related to the GZIP compression algorithm (Deutsch, 1996), this choice allowed to build upon (Iglesias et al.,

---

[1] The LLVM Compiler Infrastructure, http://www.llvm.org/.

[2] ISO 11783-6:2010 describes a terminal unit that can be used by both tractors and implements as universal Virtual Terminal (VT). While AEF (Agricultural Industry Electronics Foundation) call it as ISOBUS Universal Terminal (UT). For practical purposes the VT and UT terms can be used interchangeably

[3] LibSigComp: Open Source SigComp API, https://code.google.com/p/libsigcomp.

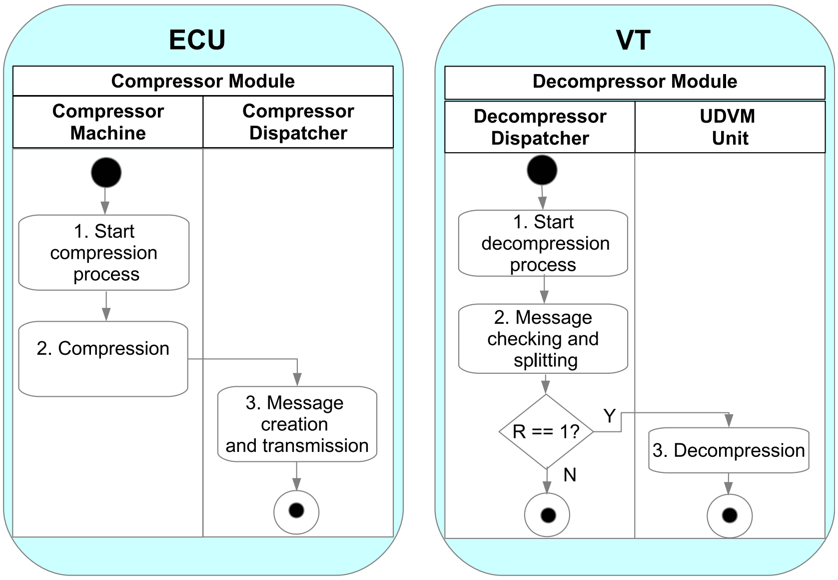


Fig. 2. A Compressor Module is composed of a *Compressor Dispatcher* and a *Compression Machine*. A Decompressor Module is composed of a *Compressor Dispatcher* and a *Compression Machine.*

2014). Bytecodes for other well-known compression algorithms like LZ77, LZSS, LZW and LZJH can be found in Surtees and West (2006).

Compression of large ISO 11783 application files by *libsigcomp* requires some adaptations, since this library has been originally designed to work with small files of a few thousand of bytes. This is reflected on the original setting of the maximum data buffer size (`MaxBuffz`) and the UDVM memory size (`UDVMmz`) parameters. The `MaxBuffz` parameter, which limits the size of uncompressed ISO 11783 application files manageable at ECU side, is set to 65,520 bytes. The `UDVMmz` parameter, which limits the size of admissible compressed ISO 11783 application files at the receiving side, is set to 8192 bytes. To match these constraints, a transparent Partition and Reassembly Mechanism (PaRM) was introduced at the Compressor and Decompressor modules respectively. Note that the choice of the partition size has two bounds, an upper one limited just by the size constraints previously mentioned and a lower one limited by the traffic overhead caused by bytecodes transmission (each piece of compressed data is transmitted with its bytecodes). In PaRM implementation, ISO 11783 application files are divided into pieces of equal `UDVMmz` size.

A flowchart of PaRM operation at the Compressor Module is shown in Fig. 3 (left). First, PaRM checks the size (`OPsize`) of the ISO 11783 application file. If the file size is less than `MaxBuffz`, then the file is compressed. Otherwise, the file is partitioned into smaller pieces of `UDVMmz` size. However, if the size of the compressed file is greater than `UDVMmz`, then the compressed file is discarded and the original uncompressed file is taken. After the original uncompressed file is partitioned into smaller pieces of `UDVMmz` size, each file piece of size `UDVMmz` is compressed and *bytecodes* are added. Similarly, a flowchart of PaRM operation at the Decompressor Module is shown in Fig. 3 (right). First, PaRM separates compressed pieces of the ISO 11783 application file according to the identification header. These pieces are individually decompressed at the *UDVM* unit and then reassembled into the original ISO 11783 application file at the *Decompressor Dispatcher.*

3. Experimental setup

Simulation experiments were made with the CANoe.ISO11783 framework, a simulation environment for modeling and testing ISO 11783 networks. A network consisting of an implement bus and a number of connected ECUs was considered (see Fig. 4). Connected ECUs can be a Tractor ECU, a VT, a Task Controller, a sprayer implement, a smart sensor, etc. ECUs within the CANoe.ISO11783 framework were programmed using the CAPL language (CANtech, 2004).

Regarding the experimental work on data compression, the following ECU functionalities were programmed: *initialization, address claim, data compression and transmission, data reception and data decompression*. In particular, the compression and decompression functions were implemented as CAPL DLLs (Dynamic Link Library) using C++ language; a CAPL DLL is a library of routines loaded and linked into applications at run time by a CAPL program. All simulations were performed using the computational environment described in Table 1.

Three types of ISOBUSComp data compression services were considered. In the first type, called Uncompressed (UC), ISO 11783 files are transferred without data compression. In the second type, called Static Compression (SC), ISO 11783 files are compressed with a fixed compression algorithm which is known in advance by the receiving ECU which holds bytecodes for decompression resident in its memory. Finally, in the third type, called Dynamic Compression with bytecodes (DC), ISO 11783 files are compressed and bytecodes for decompression are added before transmission.

To gain insight into the effectiveness and efficiency of DEFLATE data compression for general ISO 11783 files, an extension of the benchmark dataset used in Iglesias et al. (2014) was prepared. A detailed description of this extended dataset, which includes three IOP (ISOBUS Object Pool) files,[4] three BIN (Binary) files and three XML (eXtensible Markup Language) files with sizes varying from 3 KB to 162 KB, is shown in Table 2. These files were used to measure a number of data compression metrics: the *absolute size W* of transmitted files; the *relative size* W_r of transmitted files defined as the ratio between the compressed and the uncompressed file sizes; the *compression rate* R_c defined as the ratio between the uncompressed file size and the elapsed time during the compression process; and the *decompression rate* R_d defined as the ratio between the uncompressed file size and the elapsed time during the decompression process. Since both R_c and R_d are related to the specific data format and available machine resources, special care was taken to

---

[4] The IOP file is used for the Object Pool description.

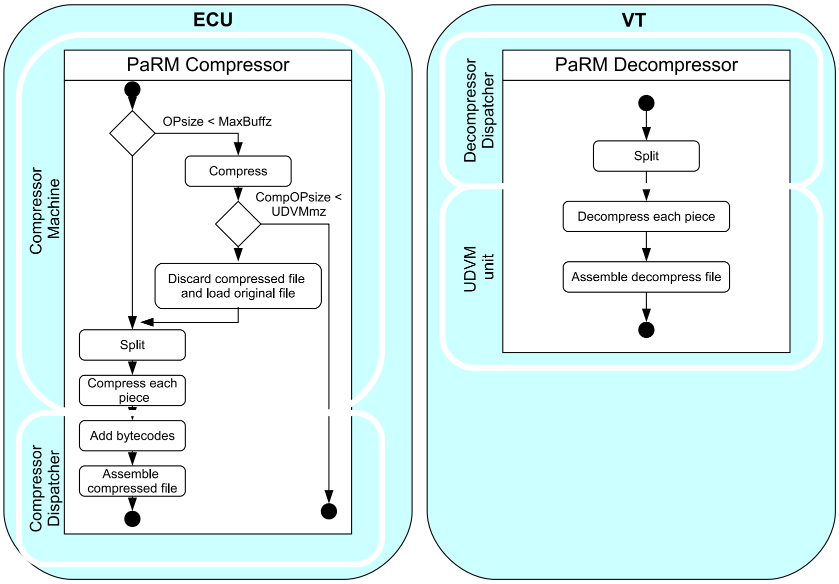


Fig. 3. Flowchart of the PaRM operation at the Compressor (left) and Decompressor Modules (right).

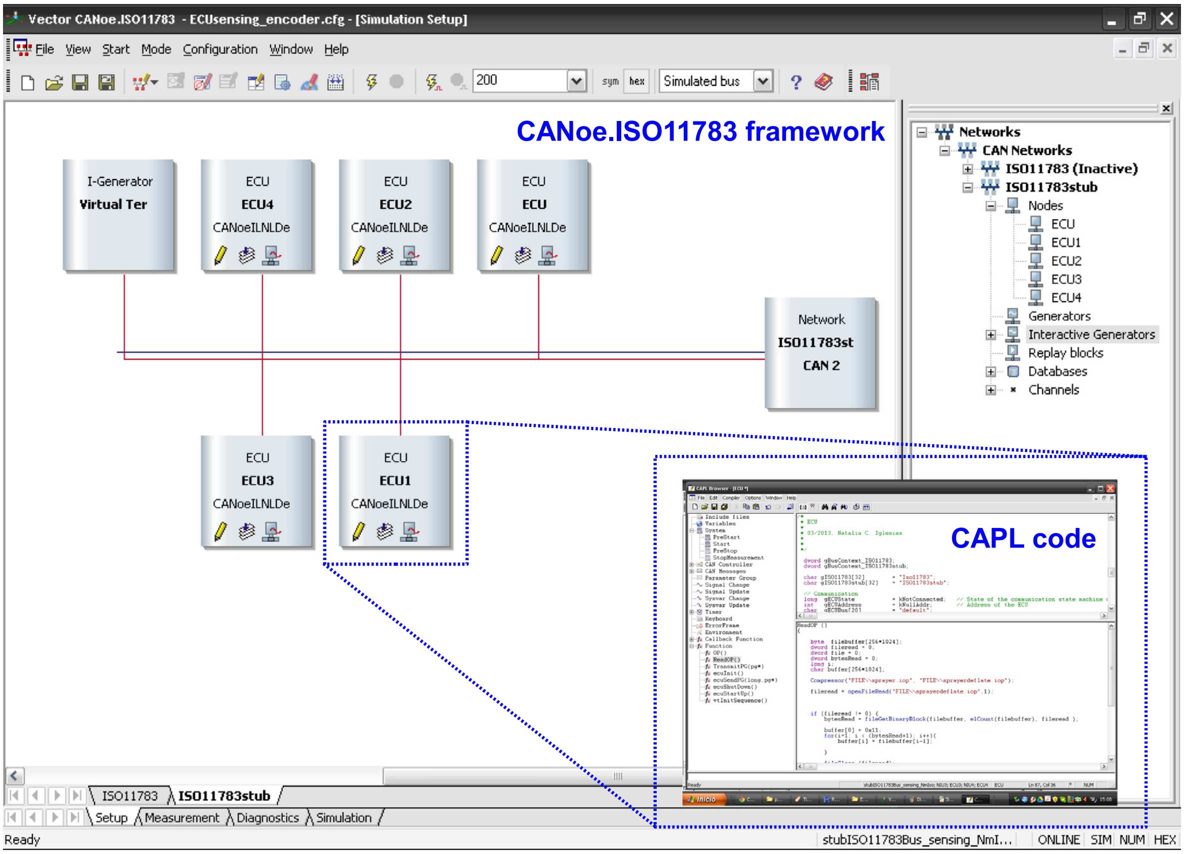


Fig. 4. The CANoe.ISO11783 simulation environment developed for the evaluation of *UC* (Uncompressed), *SC* (Static Compression), and *DC* (Dynamic Compression) data compression services based on the ISOBUSComp protocol.

ensure the same computational resources across all simulation experiments.

Finally, to evaluate the actual impact of ISOBUSComp on ISO 11783 network performance, an operation scenario where multiple ECUs aim to simultaneously transmit IOP files to a designated ECU was considered. Practically, the designated ECU can be a VT, a Task Controller or a File Server. For the sake of simplicity, the operation scenario is restricted to the transmission of IOP benchmark files to a designated VT unit. Furthermore, taking into account that ISOBUS network initialization involving the transmission of large OP files may lead to critical bus utilization when the number of connected ECUs is increased (Iglesias et al., 2014), three ISO 11783 network configuration scenarios comprising one (*N*1), three (*N*3) and five (*N*5) connected ECUs were considered. In all these network

Table 1
Setup details of the computational environment.

Resources	Details
Operating system	Microsoft Windows XP Professional SP 3
CPU	Intel Core(TM)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

configurations, the enhancement of all devices with the proposed ISOBUSComp data compression service was assumed. At each network configuration, the following network performance variables were then monitored across 10 simulation experiments: the *bus utilization* U defined as the ratio between the actual bus utilization and the maximum number of bits allowed in the bus during a period of time of the 1 s; and the *file transfer duration* T defined as the time for a file transmission process (see Fig. 5). Both U and T were characterized by their average values, $\overline{U}$ and $\overline{T}$ respectively. In addition, U was further characterized by its average peak value $\overline{U_p}$. Network measurements were performed using real traffic traces generated with the set of messages described in Table 3.

Table 3
ISO 11783 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 messages 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

Regarding the creation of such messages within the CANoe.ISO11783 framework, the CANdb++ editor was used; a typical traffic trace is shown in Fig. 5.

4. Results

Results regarding the effectiveness and efficiency of DEFLATE in the context of ISOBUSComp data compression services are shown

Table 2
The extended benchmark dataset.

Name	Size (Byte)	Description	Format	Source
Sprayer	3037	Simple GUI for a sprayer implement	.iop	Vector-Infomartik[a]
ISOBUSMask	16895	Naive GUI design with a single image	.iop	Vector-Infomartik
map10	165793	Advanced GUI for a planting implement	.iop	Proprietary
TASKDATA01	3874	Coding data and number of tasks of a generic implement	.xml	AEF[b]
TASKDATA02	65904	Coding data and number of tasks of a generic implement	.xml	AEF
TASKDATA03	70162	Coding data and number of tasks of a generic implement	.xml	AEF
GRD00001	3445	Binary Grid-file	.bin	AEF
TLG00002	7276	Binary Timelog data file	.bin	AEF
GRD00003	11520	Binary Grid-file	.bin	AEF

[a] http://vector.com/.
[b] http://www.aef-online.org/.

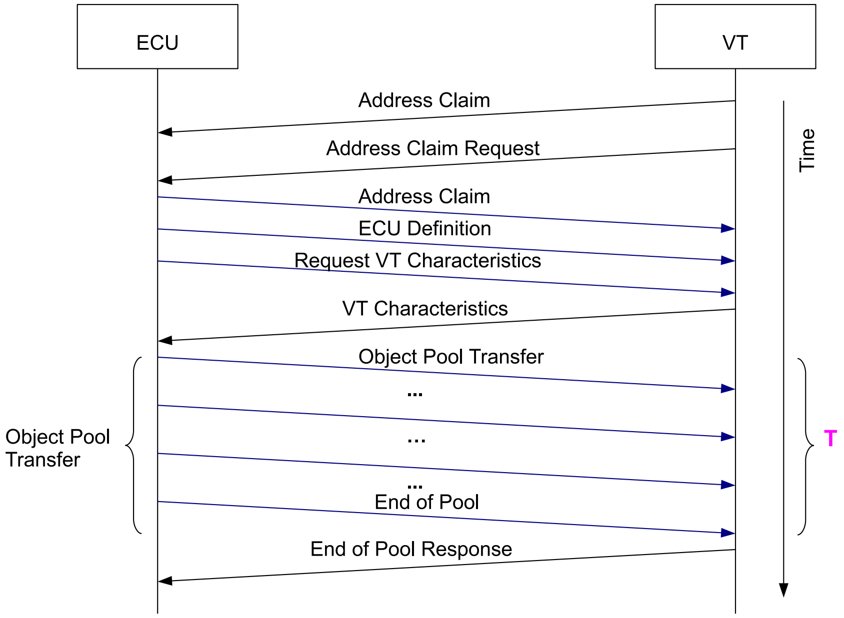


Fig. 5. A typical exchange of ISO 11783 messages when transmitting OP data. The extent of measurements for the duration of files transfers T is shown.

Table 4
The effectiveness and efficiency of DEFLATE data compression on general ISO 11783 files. The absolute and relative file sizes W and W_r respectively are shown with the mean rate of data compression R_c and the mean rate of data decompression R_d. Standard deviations for both R_c and R_d are presented in parentheses. Best results are shown in bold.

File	Service	W (kB)	W_r	R_c (kbps)	R_d (kbps)
Sprayer.iop	UC	2.9677	1.0000	–	–
	SC	1.7080	**0.5755**	108.6947 (6.3368)	155.2947 (12.3346)
	DC	2.0107	**0.6775**	160.7520 (11.9840)	178.9735 (8.9529)
ISOBusMask.iop	UC	16.4990	1.0000	–	–
	SC	3.1318	**0.1898**	1028.0723 (100.8226)	1198.8973 (84.6296)
	DC	3.4345	**0.2081**	1576.8667 (96.9393)	1148.6570 (94.3600)
map10.iop	UC	161.9072	1.0000	–	–
	SC	13.7060	**0.0846**	2720.5931 (125.3474)	265.3029 (4.6003)
	DC	20.0634	**0.1239**	4869.8695 (245.4848)	266.9067 (4.7973)
TASKDATA01.xml	UC	3.7832	1.0000	–	–
	SC	0.6748	**0.1783**	162.3263 (4.8293)	113.5123 (9.5985)
	DC	0.9775	**0.2583**	214.1395 (12.8727)	103.2428 (7.7324)
TASKDATA02.xml	UC	64.3593	1.0000	–	–
	SC	17.2236	**0.2676**	1768.8626 (135.0682)	188.3372 (4.1153)
	DC	22.9023	**0.3558**	4095.5086 (275.9507)	191.1909 (3.3710)
TASKDATA03.xml	UC	68.5175	1.0000	–	–
	SC	19.0468	**0.2779**	1961.533 (177.0213)	333.1645 (58.1451)
	DC	21.7714	**0.3177**	2957.3891 (141.96394)	194.5778 (2.9961)
GRD00001.bin	UC	3.3642	1.0000	–	–
	SC	0.2285	**0.0679**	189.1440 (25.4494)	125.4026 (6.0597)
	DC	0.5312	**0.1579**	257.5012 (36.1526)	79.7973 (4.9378)
TLG00002.bin	UC	7.1054	1.0000	–	–
	SC	3.8359	**0.5398**	390.1406 (43.5097)	24.7591 (0.1163)
	DC	4.1386	**0.5835**	599.0224 (77.4908)	24.6961 (0.1315)
GRD00003.bin	UC	11.2500	1.0000	–	–
	SC	0.3925	**0.0348**	797.3377 (66.1761)	201.2988 (5.9344)
	DC	0.6953	**0.0618**	857.5287 (89.1361)	190.5323 (6.8524)

in Table 4. Important reductions in the absolute file size W of transmitted files are observed when using either *SC* or *DC* services. Although these reductions are highly data-dependent, they are not substantially affected by the specific use of *DC* or *SC* services. As shown by W_r results, the additional overhead introduced by transmision of bytecodes in *DC* services is minimum. In addition, it is observed that data compression rates R_c are generally fast compared to the recommended 250 kbps transmission rate of ISO 11783 networks. On the other hand, decompression rates R_d are generally in the order. Thus, it is possible to say that using *DC* services allows important reductions in the size of transmitted files with the advantage of having flexibility to choose the data compression method to use on the fly.

Similarly, simulation results for the evaluation of the bus utilization U are shown in Table 5 and Fig. 6. Important reductions in the average bus utilization $\overline{U}$ are observed when using either *DC* or *SC* services compared to *UC* ones. Furthermore, an important reductions in the duration of average bus utilization peaks $\overline{U_p}$ are also observed, recalling that a bounded duration of bus utilization peaks is a critical factor to prevent network congestion; the reduction of peaks duration following from reductions in accompanying $\overline{U}$. These results become even more relevant when either the number of connected ECUs is increased or increasingly larger files are transmitted. Thus, it is possible to say that the flexibility gained through the use of *DC* services does not occur at the expense of network performance degradation.

Table 5
Bus utilization U on the $N1$, $N3$ and $N5$ networking scenarios when using *UC*, *SC* and *DC* services for the transmission of IOP files. The average bus utilization $\overline{U}$ and the average bus utilization peak $\overline{U_p}$ are reported.

File	Service	U on $N1$ (%)		U on $N3$ (%)		U on $N5$ (%)	
		$\overline{U}$	$\overline{U_p}$	$\overline{U}$	$\overline{U_p}$	$\overline{U}$	$\overline{U_p}$
Sprayer.iop	UC	6.84	26.46	15.47	52.84	25.54	73.90
	SC	4.08	15.35	11.86	45.83	15.28	55.89
	DC	4.66	17.70	10.03	46.78	15.78	57.36
ISOBusMask.iop	UC	23.24	83.07	48.02	100.00	61.73	100.00
	SC	6.98	26.97	15.41	53.64	23.01	76.08
	DC	7.59	29.41	15.21	57.98	25.81	88.16
Map10.iop	UC	70.17	95.45	87.12	100.00	91.10	100.00
	SC	20.27	64.41	42.75	100.00	56.57	100.00
	DC	25.42	92.23	53.65	100.00	65.63	100.00

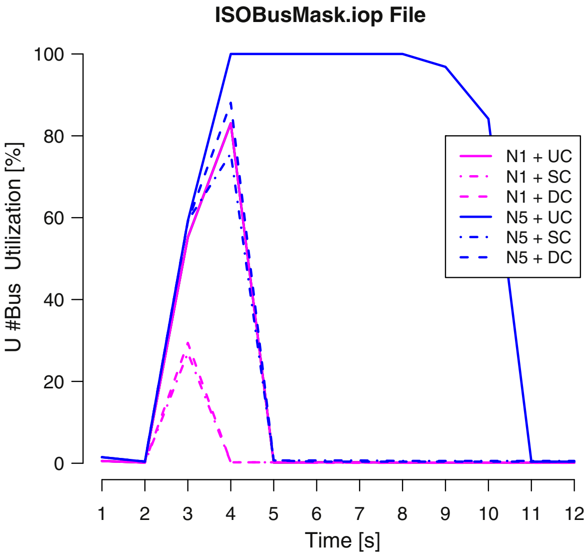


Fig. 6. Bus utilization (U) when transmitting the ISOBusMask.iop file with *UC*, *DC* and *SC* services over the extreme $N1$ and $N5$ networking scenarios. Average bus utilization $\overline{U}$ with *DC* services is roughly 3 times lower than with *UC* ones.

Table 6
The transfer file duration T on the $N1$, $N3$ and $N5$ networking scenarios when using *UC*, *SC* and *DC* services for the transmission of IOP files. The average file transfer duration $\overline{T}$ is reported; standard deviations for T are presented in parentheses.

File	Service	T on $N1$ (s)	T on $N3$ (s)	T on $N5$ (s)
Sprayer.iop	UC	0.27429 (0.00055)	0.62824 (0.01208)	0.90604 (0.02776)
	SC	0.18259 (0.00094)	0.36615 (0.00423)	0.55452 (0.00778)
	DC	0.19130 (0.00049)	0.41745 (0.01154)	0.60929 (0.01246)
ISOBusMask.iop	UC	1.48799 (0.00446)	3.31582 (0.01008)	5.08408 (0.29358)
	SC	0.28727 (0.00109)	0.65291 (0.00513)	0.93853 (0.01310)
	DC	0.31458 (0.00072)	0.69338 (0.00997)	0.99532 (0.01968)
Map10.iop	UC	14.67519 (0.64304)	32.36589 (0.18118)	47.16936 (0.21213)
	SC	1.24048 (0.00761)	2.74392 (0.01114)	4.12845 (0.03570)
	DC	1.82879 (0.02124)	3.41335 (1.00377)	6.00440 (0.01570)

Table 7
Proposed definition for Parameter Group Number (PGN). The following abbreviations are used: DP data page (1 bit), PF PDU format (8 bits), DA destination address (8 bits), PS PDU specific (8 bits), and PGN parameter group number (3 bytes).

PGN	MP	FP	PS	Description
116736	1	200	DA	Extended Transport Protocol – Connection Mgt with Data Compression
125952	1	236	DA	Transport Protocol – Connection Mgt with Data Compression

Finally, simulation results for the evaluation of the files transfer duration T are shown in Table 6. Important reductions in $\overline{T}$ are observed when using either *DC* or *SC* services compared to *UC* ones. For example, for the ISOBusMask.iop file in the $N1$ networking scenario, the $\overline{T}$ with *SC* or *DC* services is roughly 5 times lower than with *UC* ones. As with bus utilization U, positive T effects become more visible when the number of connected ECUs is increased and increasingly larger files are transmitted.

5. Discussion

The aim of this work is to discuss how data dynamic data compression methods could be realized on ISO 11783 networks. The main focus is on the provision of a data compression service for the transference of the three type of files used in the ISO 11783 standard, namely, IOP (ISOBUS Object Pool), XML (Extensible Markup Language) and BIN (Binary) data files. IOP files are used to describe the Object Pool, XML files are used to describe devices, tasks and data storage, and BIN files are used to describe work areas and data storage. The use of the proposed data compression service is intended to be optional and thus, it's only required that all specific ECUs (VT, TC, FS, ECU Tractor) defined by the ISO 11783 standard are aware of it by providing suitable mechanisms for its acceptation or rejection. For this purpose, multiple operational options are possible. A first option is the generation of new Technical Data Messages at the application level for each specific ECU (VT, TC, FS, ECU Tractor). A second option is the generation of two new standard connection management messages with data compression at the transport level, e.g., TP.CM_RTS.DC and ETP.CM_RTS.DC, one for each transport protocol respectively using reserved 18_{10} and 24_{10} control bytes of the ISO 11783 standard. Finally, a third option is the use of the R bit as proposed in (Misbahuddin et al., 2001) for the data compression signaling on J1939 networks. Taking into account that the introduction of new data compression services within the ISO 11783 standard should be as transparent and simple as possible to prevent messages loses when bus utilization is close to 100%, the third option involving the use of the R bit is proposed. When bus utilization is close to 100%, messages must be processed fast enough to avoid messages loses due to messages overrun. Signaling data compression services at the application or transport levels involves a processing overhead which is remarkably higher than that arising from the use of the R bit at the data link level (ISO11783-3, 2007). It may argued that the R bit is a precious resource to use it just for data compression signaling purposes. In this regard, an intermediate option for data compression signaling is be the definition of new PGN values (ISO11783-1, 2007). For example, PGN 116736 and PGN 125952 values in relation to PF (PDU Format) values definided by the ISO 11783 standard for transport protocol messages in conjunction with the DP (Data Page) bit (Table 7).

6. Conclusions

A compression protocol, called ISOBUSComp, suitable for the implementation of dynamic data compression services regarding the efficient transmission of large files in ISO 11783 networks has been presented. Experimental results show that the proposed protocol allows the efficient transmission of bulky ISO 11783 files in a remarkable flexible way: compression solutions can be independently selected at each transmitting ECU without the need that receiving ECUs are aware of such decisions. This unique feature is built upon an adaptation of the SigComp protocol traditionally used for the efficient implementation of VoIP services. A key element of the ISOBUSComp protocol is the requirement of a Universal Decompression Virtual Machine resident at receiving ECUs. This virtual machine is used to decompress any compressed ISO 11783 file provided suitable bytecodes for decompression have been sent in advance. Since the deployment of dynamic data compression services based on ISOBUSComp does not require changes to the current ISO 11783 standard, their implementation on ISO 11783 compliant devices should be relatively easy.

Acknowledgement

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

References

Vanclay, F.M., Russell, A.W., Kimber, J., 2013. Enhancing innovation in agriculture at the policy level: the potential contribution of technology assessment. Land Use Policy 31 (0), 406–411.

Munack, A., Speckmann, H., 2001. Communication technology is the backbone of precision agriculture. Agric. Eng. Int.: CIGR J. Sci. Res. Develop. III.

Pereira, R.R.D., Godoy, E.P., Sakai, R.M.R., Porto, A.J.V., Inamasu, R.Y., 2010. Implementation and evaluation of a CAN-based distributed control system for variable rate technology in agricultural machinery. In: ABCM Symposium Series in Mechatronics, vol. 4, pp. 460–469.

Wang, N., Zhang, N., Wei, J., Stoll, Q., Peterson, D., 2007. A real-time, embedded, weed-detection system for use in wheat fields. Biosyst. Eng. 98 (3), 276–285.

Cox, S., 2002. Information technology: the global key to precision agriculture and sustainability. Comput. Electron. Agric. 36 (2-3), 93–111. http://dx.doi.org/10.1016/S0168-16990200095-9.

ISO11783, 2007. Tractors and Machinery for Agriculture and Forestry – Serial Control and Communications Data Network.

Backman, J., Oksanen, T., Visala, A., 2013. Applicability of the ISO 11783 network in a distributed combined guidance system for agricultural machines. Biosyst. Eng. 114 (3), 306–317.

Suprem, A., Mahalik, N., Kim, K., 2013. A review on application of technology systems, standards and interfaces for agriculture and food sector. Comput. Stand. Interfaces 35 (4), 355–364.

Aubert, B.A., Schroeder, A., Grimaudo, J., 2012. IT as enabler of sustainable farming: an empirical analysis of farmers' adoption decision of precision agriculture technology. Decis. Support Syst. 54 (1), 510–520.

Egorova, R., Borst, S., Zwart, B., 2007. Bandwidth-sharing networks in overload. Perform. Eval. 64 (9–12), 978–993, 26th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation.

Peets, S., Mouazen, A.M., Blackburn, K., Kuang, B., Wiebensohn, J., 2012. Methods and procedures for automatic collection and management of data acquired from on-the-go sensors with application to on-the-go soil sensors. Comput. Electron. Agric. 81, 104–112.

Nash, E., Dreger, F., Schwarz, J., Bill, R., Werner, A., 2009. Development of a model of data-flows for precision agriculture based on a collaborative research project. Comput. Electron. Agric. 66 (1), 25–37.

Steinberger, G., Rothmund, M., Auernhammer, H., 2009. Mobile farm equipment as a data source in an agricultural service architecture. Comput. Electron. Agric. 65 (2), 238–246.

Lee, D.-H., Lee, K.-S., Moon, J.-M., Park, S.-J., Kim, C.-S., Kim, M.-H., Cho, Y.-J., Kim, S.-M., 2012. Evaluation of the implementation of ISO 11783 for 250 kbps transmission rate of tractor electronic control unit. J. Biosyst. Eng. 37 (4), 225–232.

Herpel, T., Hielscher, K.-S., Klehmet, U., German, R., 2009. Stochastic and deterministic performance evaluation of automotive CAN communication. Comput. Netw. 53 (8), 1171–1185.

Mainoo, J., May 2012. A Study of the Effects of Fieldbus Network Induced Delays on Control Systems, Ph.D. Thesis, College of Technology. Indiana State University, Terre Haute, Indiana.

Passos, D., Carrano, R., Albuquerque, C., 2013. On the decrease in frame reception probability under heavy transmission loads in IEEE 802.11 networks. Comput. Stand. Interfaces 35 (4), 374–379.

Kim, J.C., Lee, Y., 2014. An end-to-end measurement and monitoring technique for the bottleneck link capacity and its available bandwidth. Comput. Netw. 58, 158–179.

Stone, M., 1994. High Speed Networking in Construction and Agricultural Equipment. Tech. Rep.

Stone, M., McKee, K., Formwalt, C., Benneweis, R., 1999. In: ISO 11783: An Electronic Communications Protocol for Agricultural Equipment, Agricultural Equipment Technology Conference, ASAE, Louisville, Kentucky.

Miucic, R., Mahmud, S., Popovic, Z., 2009. An enhanced data-reduction algorithm for event-triggered networks. IEEE Trans. Veh. Technol. 58 (6), 2663–2678.

Ramteke, P.R., Mahmud, S.M., 2005. An Adaptive Data-reduction Protocol for the Future in-vehicle Networks. Tech. Rep., SAE.

Iglesias, N., Bulacio, P., Tapia, E., 2014. Enabling powerful GUIs in ISOBUS networks by transparent data compression. Comput. Stand. Interfaces 36 (0), 801–807.

Vector-Informatik, 2011. CANoe.ISO11783 ver 7.6.

Blank, S., Bartolein, C., Meyer, A., Ostermeier, R., Rostanin, O., 2013. iGreen: a ubiquitous dynamic network to enable manufacturer independent data exchange in future precision farming. Comput. Electron. Agric. 98 (0), 109–116. http://dx.doi.org/10.1016/j.compag.2013.08.001.

Oksanen, T., 2010. Closed loop control over ISO 11783 network-challenges of plug-and-play. In: Proceedings of the Agricontrol 2010, Agricontrol 2010, Kyoto, Japan, December 6–8, 2010.

Godoy, E.P., Tabile, R.A., Pereira, R.R.D., Tangerino, G.T., Porto, A.J.V., Inamasu, R.Y., 2010. Design and implementation of an electronic architecture for an agricultural mobile robot. Rev. Bras. Eng. Agric. Ambient. 14, 1240–1247.

Du, D., Liu, J., Li, H., Peng, Z., 2013. Research of SIP compression based on SigComp. Res. J. Appl. Sci., Eng. Technol. 5 (22), 5320–5324.

Wu, Y., Liu, Y., Li, M., 2010. Performance evaluation of SigComp in MANETs. In: 2010 International Conference on Multimedia Information Networking and Security (MINES), pp. 119–122.

Bin, D., FuRong, W., Jinshui, K., 2006. Performance analysis of signaling using sigcomp scheme in narrowband system. In: Consumer Communications and Networking Conference, 2006. 3rd IEEE, vol. 1, pp. 376–379.

Jin, H., Mahendran, A.C., 2005. Using SigComp to compress SIP/SDP messages. In: 2005 IEEE International Conference Communications, 2005. ICC 2005, vol. 5, pp. 3107–3111.

Price, R., Bormann, C., Christoffersson, J., Hannu, H., Liu, Z., Rosenberg, J., January 2003. Signaling Compression (SigComp), RFC 3320.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E., June 2002. SIP: Session Initiation Protocol, RFC 3261.

Downing, T., Meyer, J., 1997. Java Virtual Machine, Java Series, O'Reilly Media.

Deutsch, P., May 1996. DEFLATE Compressed Data Format Specification Version 1.3, RFC 1951.

Deutsch, P., May 1996. GZIP File Format Specification Version 4.3, RFC 1952.

Surtees, A., West, M., May 2006. Signaling Compression (SigComp) Users' Guide, RFC 4464.

CANtech, V., December 2004. Programming with CAPL, first ed.

Misbahuddin, S., Mahmud, S., Al-Holou, N., 2001. Development and performance analysis of a data-reduction algorithm for automotive multiplexing. IEEE Trans. Veh. Technol. 50 (1), 162–169. http://dx.doi.org/10.1109/25.917911.

ISO11783-3, 2007. Tractors and Machinery for Agriculture and Forestry – Serial Control and Communications Data Network. Part 3: Data Link Layer.

ISO11783-1, 2007. Tractors and Machinery for Agriculture and Forestry – Serial Control and Communications Data Network. Part 1: General Standard for Mobile Data Communication.