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).

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).

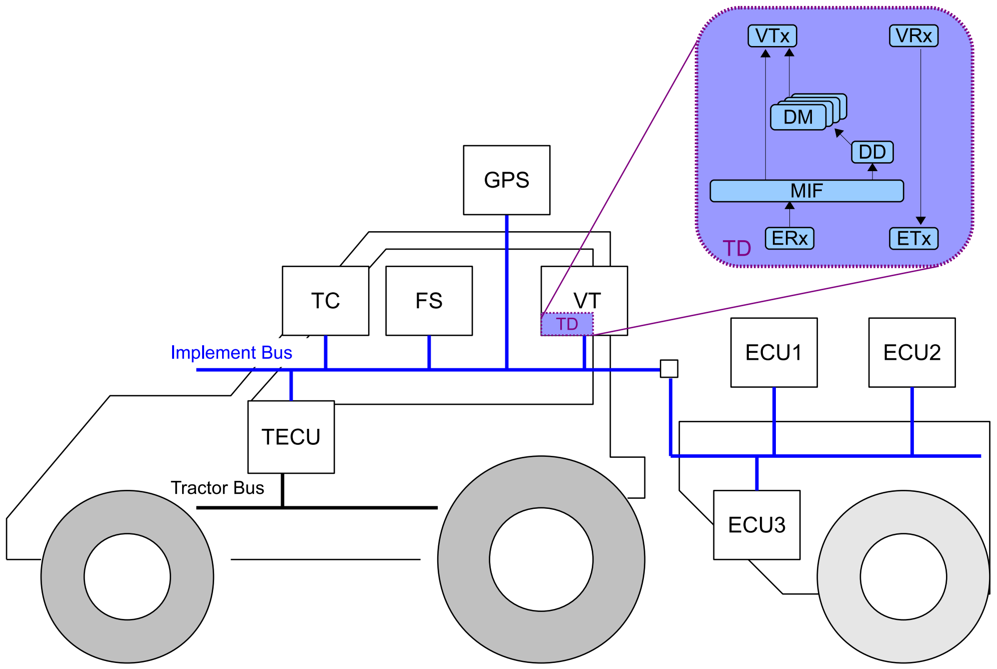

Fig. 1. Schematic form of a simplified ISO 11783 network. The TD application serves as an interface between the general VT functionalities and the ISOBUS network. The TD application is made up of several entities, an ECU Transmitter (*ETx*) and an ECU Receiver (*ERx*) pair, a VT Transmitter (*VTx*) and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