ISA Transactions 59 (2015) 457–470

Contents lists available at ScienceDirect

ISA Transactions

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

Protocol description and optimization scheduling for multi-fieldbus integration system

Chongquan Zhong, Chen Chen *

Faculty of Electronic Information and Electrical Engineering, Dalian Univ. of Technology, Dalian, Liaoning 116024, China

A R T I C L E I N F O

Article history:
Received 2 November 2014
Received in revised form
16 September 2015
Accepted 11 October 2015
Available online 26 October 2015

Keywords:
Multi-fieldbus integration
Protocol description
Packet encapsulation
Network load optimization
Communication scheduling

A B S T R A C T

Device integration technique is applied to integrate different fieldbus devices into one control system. At present mature integration techniques use appropriative software to support corresponding protocols. New software must be developed when a new fieldbus is integrated. In this research, a universal protocol description method is proposed. It focuses on the packets encapsulation description, and different protocol messages can be encapsulated and parsed by the interpreter in a unified way. Moreover, in order to ensure the communication efficiency and QoS of different kinds of messages, packets encapsulated via protocol description are optimized and scheduled before transmission inside the interpreter. The approaches have been applied in the prototype of a software product and verified in a power monitoring project.

1. Introduction

With the maturity of control network and fieldbus technologies, there is quite a large demand for the fieldbus based control systems. In today's automation applications, field devices from different suppliers with different protocols have to be integrated into one system. It is necessary to provide effective methods to make the appropriate device information available for different users [1], which is best met with an open device integration solution. Currently there are three standard device integration techniques: Electronic Device Description Language (EDDL), Field Device Tool (FDT) and Field Device Integration (FDI).

EDDL is text-based and it is simple and flexible. It defines device parameters, variables, functions, and protocols via a series of basic label elements. It depends on the interpreter to realize fieldbus driver. Appropriative interpreters are needed for HART, FF and Profibus protocols [2]. FDT is based on Windows OS and Microsoft Component Object Model (COM). It uses appropriative communication DTM components to support different communication protocols [3]. It is powerful but the development of DTM requires specialized skills, hence to be relatively difficult for users. FDI Server communicates with fieldbus devices via communication hardware it natively supports or FDI Communication Server through OPC UA [4]. Each solution has the pros and cons, however all of them use appropriative software programmes or components to support corresponding protocols. It will be a hard job for the end users to develop it provided that the appropriate products are not available or the protocol is self-defined or atypical.

The mode of EDDL to achieve device integration is more convenient and lower cost. However the interpreter is protocol dependent, appropriative interpreters are needed for corresponding protocols. To extend the compatibility of EDDL, a growing number of experts and scholars began to use XML for devices description. The fieldbus control system management structure based on XML was proposed [5] and the process data and device information was described for FF, EPA and CAN fieldbuses [6–10]. These researches extend the supported protocol types. However, the interpreter is still protocol dependent and the compatibility problem is not resolved essentially. Fabio Baroncelli etc. proposed a concept of XMPL (XML-based Multi-Protocol Language), a XML-based language for protocol description. Although no details of the language have been introduced, they pointed out the objects which should be descripted: protocol logic, protocol message set, and protocol data structures, and demonstrated the structure of the software engine which is able to run as a specific protocol application leveraging the language [11]. Michael A. and Gordon B. introduced the PP (Packet Parsing) language, a simple high-level language for describing packet header parsing algorithms. The format of packet header was defined by an ordered fields list. FPGA was used to analyze these structures and realize efficient message classification and retransmission [12]. Gatan N. C. came up with a solution which described the capabilities of devices based on

* Corresponding author. Tel.: +86 13998569945.
E-mail addresses: zhongcq@dlut.edu.cn (C. Zhong), chenchendcce@163.com (C. Chen).

http://dx.doi.org/10.1016/j.isatra.2015.10.009
0019-0578/

DCON ASCII protocol. The solution is based on the EDS (Electronic Data Sheet) files that were extended with the mission to describe the commands, and the answers used to retrieve information for each device [13].These researches provided a new vision for device integration. We can describe the packet structure and encapsulation method of fieldbus packets through protocol description. Different protocol messages can be packaged and analyzed by the interpreter in a unified way. Suppose that a new type of fieldbus needs to be integrated, only a new protocol description file is required and the interpreter is not necessarily required to be redeveloped.

Furthermore, although fieldbuses have respective facilities for message scheduling and prioritizing at Data Link Layer [14,15], in practical applications we noticed that sometimes the communication efficiency [16] and QoS of different priority messages are still unsatisfactory. For example the interpreter reads variables with address in a row one by one or reads a whole register area just for a few variables, which reduces the communication efficiency and brings negative impact on the network load. Moreover, a real time message is likely to be blocked by a packet inside the interpreter. It may be a message with lower priority or one of a higher priority but with remaining sufficient timeout period. So packets encapsulated via protocol description should be optimized and scheduled before transmission inside the interpreter.

So far, the optimization and scheduling research of fieldbus mainly focuses on the following three aspects: Data link layer schedule research according to communication delay, network load and packet loss rate for specific fieldbuses [17–20]; Backlog and schedule delay analysis based on queuing theory and network calculus algorithms for network transmission nodes [21–26]; Queue scheduling strategy to ensure QoS of different priority messages [27–29]. The schedule research, backlog and delay analysis corresponds to Data Link Layer, and they are closely related to the MAC strategy of respective fieldbuses. Queue schedule strategies, such as PQ, EDF, WFQ [27] and LLQ [29] are helpful to improve the QoS of the communication messages. However the upper bound of schedule delay is hard to be guaranteed. In this research, the interpreter works at application layer. Characteristics of application data should be analyzed for period message optimization. In addition, the queuing delay of different kinds of message inside the interpreter should be estimated for transmission scheduling.

The paper is organized as follows. Section 2 introduces the principle, structure and content of the proposed description method. Section 3 illustrates the packets encapsulation and optimization methods. Section 4 analyzes the schedule delay of different kinds of messages and puts forward a priority scheduling algorithm with transmission control strategy. The simulation and application are illustrated in Section 5. Finally some conclusions and future works are drawn in Section 6.

2. Protocol description

2.1. Fieldbus hierarchy

Nineteen kinds of fieldbuses are recorded in the fourth edition of IEC61158 standard. According to ISO/OSI model, IEC61158 standard simplifies the classic seven layers to physical, data link and application layers [30].

Among three layers, application layer is designed to support the conveyance of application requests and responses among fieldbus nodes. It uses application protocol to implement data exchange, device management and other application services. Application protocol is a set of rules that governs the format and meaning of the information exchanged between application layers.

Table 1
Parameters in SM area of EPA protocol

Address	Parameter	Description
–	–	–
SM13–16	Serial no.	Device unique serial number
SM17–20	IP address	IP address of the fieldbus device
SM22–28	AI type	Sensor type of no.0–6 analog input channels
SM29–30	SerialPort1 status	Transfer mode of serial port 1
SM31–32	SerialPort2 status	Transfer mode of serial port 2
SM33	ADC mode	AD converter control word
SM34	DI Stabilizer	DI converter stabilization time
–	–	–

Device parameters and variables are main objects to be managed. Parameters reflect and affect device configuration and running status; variables restore device input/output values or calculation result of control program. Parameters and variables are stored in device registers or memorizer areas. Table 1 illustrates parts of the SM (State Memorizer) Area, which is a parameter collection of an EPA fieldbus device.

Parameter setting, variable access and other operations are executed through communication commands. Commands are used to operated fieldbus devices and provide corresponding services for upper applications. Typical commands of EPA, Modbus, FF, Profibus fieldbuses are shown in Fig. 1.

As shown in Fig. 1, there are two kinds of commands: one is for device management, which is used to operate device parameters; the other is for data exchange, which is used to operate device variables. Different commands have different communication modes. It demonstrates how the software to interact with a fieldbus device for command execution. There are three kinds of communication modes [31]: Request/response mode is used to exchange information between two stations, such as peer to peer operation, client/server or Master/Slave paradigm; subscriber/publisher mode allows multiple stations to subscribe data from one data source, such as variable distribute service of EPA or subscriber–publisher VCR (virtual communication relationship) in FF; in report distribution mode, provider distributes data to all devices when trigger-event is activated.

Commands are executed through communication packets. A packet is an ordered sequence of bits exchanged between fieldbus nodes for command execution. A typical application layer packet is usually composed by packet header, data segment and check code. And each part is the combination of one or more packet blocks. Each block has attributes such as length, position, and value, as shown in Fig. 2.

In engineering applications, workstations should support the three-layer model to realize the device integration for different fieldbuses. In physical and data link layers, fieldbus access and MAC strategies can be achieved through standard appropriative hardware, such as network interface cards, adaptors, network bridges etc.; for application layer, software should support different application protocols. Through above analysis, in order to make the interpreter communicate with multi fieldbus devices, protocol description should at least support communication commands, packets, parameters and memorizer areas description.

2.2. The realization of protocol description

Protocol description can be realized by XML. The structure of protocol description is shown in Fig.3. *ProtocolDesc* is the root element. It includes basic attributes of the description file (as shown in Fig. 4(a)), and the list of *command*, *PacketBlock*, *Area* and *Para* elements.