

A Data Sharing Scheme among Industrial Process Controllers Based on UDP

Liu Liu, Jun Du, Yandong Zhang and GuanJun Qin

NR Electric Co., Ltd. Nanjing, 211102, China

Abstract—With the development of industrial process control, more and more data need to be exchanged among industrial process controllers. Based on analysis of advantages & shortcomings of existing data sharing schemes among industrial process controllers, a new high-efficiency scheme is provided in this paper. Compared to other schemes, the network load is reduced and system reliability is enhanced, and the configuration is simpler. Moreover, dual network mechanism is designed to minimize data packet loss, and UDP is adopted to resend lost data while broadcasting the next message. Repeated messages can be identified and discarded. This scheme has been applied in various projects and proven effective.

Keywords – industrial process controller; data sharing; user datagram protocol; object linking and embedding for process control

I. INTRODUCTION

Generally speaking, industrial process control system consists of engineering workstation, operator workstation, industrial process controller (or PLC: Programmable Logic Controller) and fabrication facility^[1]. With the development of industrial process control, more and more data need to be exchanged among industrial process controllers.^{[2][3]}

Due to expansion & retrofit of production line, industrial process control systems of various manufacturers may be involved in 1 production line: Siemens, ABB, Schneider & GE. Generally speaking, OPC (Object Linking and Embedding for Process Control) integration technology is adopted to achieve data sharing among industrial process controllers of various manufacturers.^{[4][5]}

Manufacturers of industrial process controllers can provide data sharing schemes for their own controllers, e.g.: Siemens S7-300 adopts GD (Global data) scheme to realize data sharing. Low communication rate, complex configuration method, limited message length, unencrypted plain text, simple data check, lack of redundant transmission and other problems exist in these schemes.^[6]

In this paper, a new scheme is provided to solve all the above problems. The source controller adopts a data format of “Source controller No. + Configuration page No. + Resource No. +Data type + Data quality”, and then broadcast data to the network via UDP (User Datagram protocol).^[7] The target controller monitors & receives all the network broadcast messages, and then analyzes them to get available data.

II. METHODOLOGY

A. Principle of the Data Sharing Scheme

In this paper, a new scheme based on UDP protocol is designed to achieve the following features:

- 1) Convenient data configuration;
- 2) Online data analysis;
- 3) High transmission rate;
- 4) Reliable & secure data transmission.

Hereinafter, data broadcasted by the source controller via UDP protocol is abbreviated as “Source data”, data received by the target controller via UDP protocol is abbreviated as “Target data”.

By implementing the configuration tool, “Source data” can be configured in the source controller, and “Target data” can be configured in the target controller. Configuration files need to be downloaded to the source controller & target controller, and then the target controller can parse the configuration files automatically.

Please refer to *FIGURE 1* for the principle of this data sharing scheme. In this scheme:

C674X series double kernel CPU manufactured by TI (Texas Instruments in USA) is adopted.

ARM (Advanced Reduced instruction set computing Machine) adopts Linux OS, and provides TCP/TP protocol stack.

DSP (Digital Signal Processor) can parse received messages and execute the algorithm.

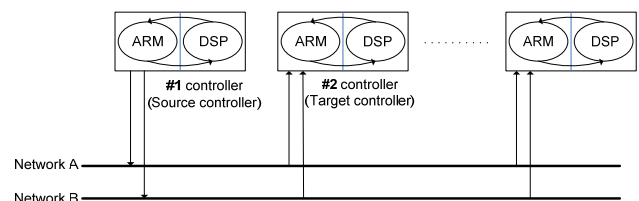


FIGURE 1. PRINCIPLE OF THIS DATA SHARING SCHEME

DSP of source controller transmits “Source data” to the ARM via shared memory, and then the ARM broadcasts “Source data” to the network via UDP protocol.

Target controller monitors new data on the network continuously. “Target data” will be received by the target controller immediately, and then the target controller will parse the target data, only required data will be sent to the shared queue of ARM, and then the ARM will transmit the required target data to DSP.

B. Principle of the Configuration Software

Configuration rule is defined in the configuration tool. “Source data” & “Target data” and their relation can be configured in the configuration tool.

E.g.: in *FIGURE 1*, “Source data” can be configured on the source controller (i.e.: #1 controller), “Target data” can be configured on the target controller (i.e.: #2 controller ~ #n controller).

Each controller can be configured as the source controller or the target controller by the configuration tool.

C. Principle of the Firmware

With continuous development of computer technology, the software development becomes a complex system project. [8] A big project can be divided to into many small parts, and then the software can be revised & expanded conveniently. [9]

In this paper, firmware programs of the controller can be divided into 6 parts as shown in *FIGURE II*.

1) Reception module:

Reception module of the target controller monitors the network broadcast messages, parses the broadcasted data in the shared memory, and then judges whether to save it or not.

2) Transmission module:

Transmission module of the source controller collects the source data, arranges them into packets and then broadcasts them via network.

3) Redundancy module:

Redundancy module of the source controller can transmit the broadcast messages to both 2 networks (i.e.: “Network A” & “Network B” in *FIGURE 1*).

Redundancy module of the target controller can receive the broadcast messages from both 2 networks and parse them. Repeated messages will be discarded.

4) Configuration file parsing module:

Configuration file parsing module is designed to parse the configuration file downloaded by the configuration tool.

5) Security module:

Security module is designed to check, rectify, encrypt & decrypt the messages.

6) Diagnosis module:

Diagnosis module can check data quality of the received data. Moreover, whether the source controller is offline can be detected by the diagnosis module.

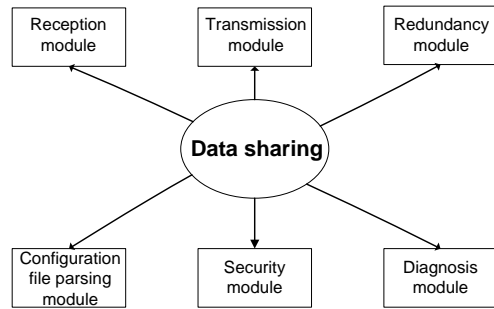


FIGURE II. FIRMWARE MODULES

D. Principle of the Source Controller DSP

Source controller DSP executes the algorithm page periodically. When the algorithm page execution is started, a global variable “g_AlgorithmTaskRunFlag” will be set as “1” to indicate the source controller algorithm page execution is in progress. If there is a new source data point, it will be appended to the source data queue.

When the algorithm page execution is completed, the global variable “g_AlgorithmTaskRunFlag” will be set as “0”, and then all the data points in the source data queue will be sent to the ARM via shared memory.

Because cyclic source data queue is adopted, when a data point is added/deleted in the next execution period, the data change will be sent to the ARM immediately.

E. Principle of the Source Controller ARM

Cyclic source data queue is also adopted in the source controller ARM.

When the source controller DSP sends source data to the shared memory, the data points will be appended to the cyclic queue. These data points need to be extracted from the cyclic queue before they are broadcasted via UDP protocol.

To ensure data integrity, when source controller ARM begins to receive data from DSP, a global variable “g_CpuTransOutDataProcessing” will be set as “1”. When all the data have been received, the global variable “g_CpuTransOutDataProcessing” will be set as “0”.

The global variable “g_CpuTransOutDataProcessing” is monitored continuously, when it is set as “0”, the source controller ARM will broadcast the source data via network A & network B.

F. Principle of the Target Controller ARM

It is very important for the target controller ARM to determine which data need to be received.

The target controller DSP will send the required data list of this controller to the target controller ARM.

The target controller ARM monitors & receives broadcasted messages via network A & network B. Received broadcasted data will be compared with the required data list in the target controller ARM. Only required data will be stored and transmitted to the target controller DSP.

G. Principle of the Target Controller DSP

The target controller DSP receives required data from the target controller ARM via shared memory, and then parses these data and stores them.

III. COMMUNICATION MECHANISM

User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite. UDP uses a simple connectionless transmission model with a minimum of protocol mechanism. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram. [6]

TABLE I.

TABLE I. SOLUTIONS TO AVOID COMMUNICATION ERRORS

No.	Message errors	Solutions			
		CRC32 check	Add message No.	Add timestamp	Encrypt
1	Damaged	Effective	/	/	/
2	Repeated	/	Effective	/	/
3	Disordered	/	Effective	/	/
4	Lost	/	Effective	/	/
5	Delayed	/	/	Effective	/
6	Intercepted	/	/	/	Effective

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Environment

Experimental environment of this scheme is listed in FIGURE III.

PCS-9150 process controllers manufactured by NR is applied as the actual controllers (i.e.: #1~#8 controller). The data sharing function mentioned above is provided by PCS-9150 controller.

Moreover, virtual controllers simulated by software are configured in the, i.e.: #9~#20 controller in this project. These virtual controllers can provide the same functions as actual PCS-9150 process controllers.

On the “Engineering workstation”, all the data changes can be monitored, and the configuration files can be edited to update the algorithm page in the controllers.

All the devices & computers are connected to dual networks. Numbers of source data & target data are listed in FIGURE III:

(S): Indicates Source data (i.e.: the controller is used as a source controller.

The lack of retransmission delays, datagram fragmenting & packing provides high-efficiency transmission. Moreover,

communication bandwidth utilization of UDP is higher than Transmission Control Protocol (TCP). [8][9][10]

Because the industrial process controller adopts embedded operating system, system resources are limited, so UDP is adopted for data transmission in this scheme.

UDP has no handshaking dialogues, so the transmission process is not reliable, extra treatment is necessary to ensure reliability & security of UDP data transmission. In our scheme, the messages are encrypted to enhance reliability & security.

In this paper, dual network mechanism is adopted to minimize data packet loss. Moreover, a cyclic redundancy check mechanism named as CRC32 [11] is implemented to ensure integrity & reliability of data transmission.

Solutions to avoid communication errors in this scheme are listed in

(T): Indicates Target data (i.e.: the controller is used as a target controller.

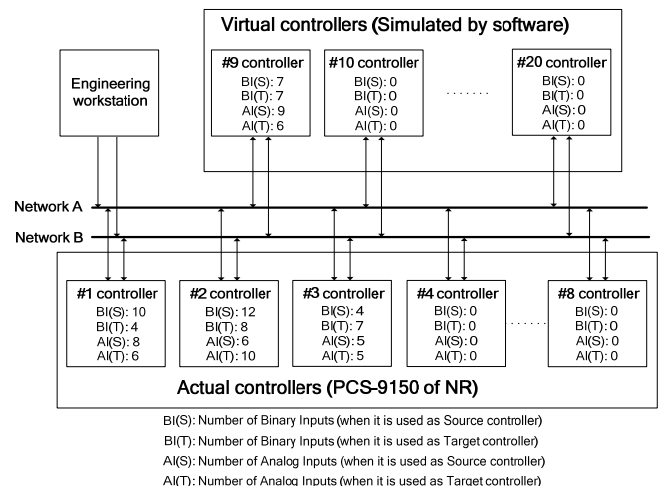


FIGURE III. EXPERIMENTAL ENVIRONMENT

B. Experimental Results and Analysis

By capturing messages with MMS Ethereal tool, we can see all the source data of a controller will be packed into 1 frame. Only the source controllers will broadcast messages: the controllers with no source data will not broadcast any messages. Therefore, the network load will not increase greatly even if there are many source data.

On the contrary, in the schemes of other manufacturers, each source data will be retrieved independently. E.g.: if there are 128 source data points, there will be 128 retrievals, so that the network load will increase greatly.

But in our scheme, there is only 1 message broadcasting & 1 message reception, the network load increment can be ignored.

Experimental results are listed in the following table.

TABLE II. EXPERIMENTAL RESULT

No.	Item	Result	Description
1	CPU load	Increase 5%~8% (Compared to the CPU load before experiment)	Checked by executing the command "uptime" in Linux OS
2	Network load	Increase 2%~3% (Compared to the network load before experiment)	
3	Delayed message	Discarded	Checked by viewing the printed information
4	Repeated message	Discarded	
5	Disconnect 1 network cable	Data OK	Checked by the configuration tool on the engineering workstation
6	Disconnect both 2 network cables	Data quality is abnormal, Data remain unchanged	
7	Network recovery	Data OK	

V. CONCLUSION AND FUTURE WORK

Based on analysis of advantages & shortcomings of existing data sharing schemes among industrial process controllers, our scheme is designed. Compared to other schemes, the network load is reduced and system reliability is enhanced, and the configuration is simpler.

In our scheme, dual network mechanism is designed to minimize data packet loss. Moreover, UDP protocol is adopted, so that the lost data can be sent again while broadcasting the next message. Repeated messages can be identified and discarded.

In the future, we will emphasis on the message encryption algorithm, so as to improve data security.

This scheme has been applied in various projects and proven effective.

REFERENCES

- [1] Hao Jiuqing, Xiao Li, "The Reliability Design of PLC Control System", *Process Automation Instrumentation*, vol.26, No.11, pp. 21-24, November 2005.
- [2] Cheng Jinliang, "An Instance for Communication Between Different PLCs", *Baosteel Technology*, pp. 40-42, 2005(s1).
- [3] Cai Fangwei, Wang Jue, Zhang Yumin, Wang Wujun, "Controlnet/Devicenet Fieldbus and its Application", *Electric Drive*, 2002, 32(1):35-39.
- [4] Zeng Rong, Feng Jiyong, Ding Mingliang, Xiang Xian Bing, "Comparison of Integration Scheme of Domestic DCS and PLC and Its Applications in Power Plant", *Process Automation Instrumentation*, vol.30, No.12, pp. 46-48, December 2009.
- [5] Zhang Yi, Han Bin, "Research on Application of Fieldbus in Power Plant Unit Control", *Process Automation Instrumentation*, 2008, 29(10):18-21.
- [6] ZhaoYan Ling, "Design and Implementation about Non-real-time Data Sharing among PLCs", *Process Automation Instrumentation*, vol.37, No.3, pp. 9-12, March 2016.
- [7] Postel J., "User Datagram Protocol"[S].RFC768.1980.

- [8] Zhao Fei, Ye Zhen, "Contrast Analysis of UDP and TCP and Improvement of UDP in Reliability", *Computer Technology and Development*, 16(9): 219-221, 2006.
- [9] Postel J., "Transmission Control Protocol"[S].RFC793, DARPA. 1981.
- [10] Velten D. Reliable Data Protocol[S]. RFC908.1984.
- [11] Kou Kenan, Fu Xiao, Sun Zhiyue, Ren Guanghui, "Research on Paralell CRC32 Algorithm Based on SATA interface", *Techniques of Automation and Applications*, 2010,29(8):40~43