ISSN: 2664-9640

Research on Multi-source Data Parallel Transmission in Intelligent Factory Based on TCP/IP CNC Machine Tools

Shaobo Ren

Sichuan university of science and engineering, Yibin 644000, China

Abstract

This paper aims at the problem of multi-source data parallel transmission in the intelligent factory project of a CNC machine tool enterprise. Based on the analysis of the types of enterprise data transmission, several data transmission models are proposed. The advantages and disadvantages of each model, as well as its application, can be clarified by comparing these models in corresponding situations. By using the corresponding data transmission model, the validity and accuracy of the transmitted data are guaranteed. The results show that these models can solve the problem of message waiting and blocking in the parallel transmission of multi-source data in intelligent factory. The model in this paper has been verified in a project of a company, which can be used as a reference for solving the problem of message waiting and blocking in the parallel transmission of multi-source data in an intelligent factory.

Keywords

Intelligent factory, TCP/IP, Data transmission.

1. Introduction

Intelligent factory is developed on the basis of digital factory, so intelligent factory also inherited the technology of digital factory, such as product data management technology, virtual manufacturing technology, rapid prototyping technology, computer aided detection technology[1], digital control technology, etc., are the basic technology of intelligent factory. At the same time, the intelligent factory also added some new technologies, so that the intelligent factory can achieve data collection, analysis, judgment, scheduling planning and other functions. With these new functions in smart factory[2], new requirements are put forward for data transmission in smart factory.

With the development of intelligent factories, more and more data need to be transmitted, and the modes of transmission are becoming more and more diverse. At this time, it may face the problem of message waiting and blocking when multi-source data are transmitted in parallel. This is the focus of this article. Since most factories now use Ethernet for data transmission, and Ethernet mostly uses TCP/IP for data transmission, so this paper is based on TCP/IP for data transmission[3].

TCP/IP was originally designed to allow computers to communicate freely with other computer terminals. TCP/IP is not a protocol, but a family of protocols collectively. It includes IP protocol, IMCP protocol, TCP protocol, as well as HTTP, FTP, POP3 protocol and so on. TCP/IP includes transport layer (TCP and UDP protocol), network layer (IP protocol) and physical interface layer (for a variety of physical hardware technology)[4], but the IP layer protocol of TCP/IP can not directly provide services to users, because the socket is Windows built-in program. So choose the socket API to serve the user.

The article is a numerical control machine tool company's intelligent factory project as the object of writing. The project is based on TCP/IP, on the basis of Socket, establish a variety of

ISSN: 2664-9640

transmission models, solve the intelligent factory in the parallel transmission of multi-source data waiting and blocking problems.

2. Smart factory project overview

The intelligent factory project of a CNC machine tool enterprise is mainly based on the production of CNC machine tools, including three production lines and a logistics line. Each production line is controlled by an industrial computer, and four lines are controlled and managed by a master controller. The three production lines are three seat line, box line and logistics line. Three lines are used for processing motor seat, bearing seat and nut seat (a total of more than 20 different specifications and shapes of parts). Through matching automatic tooling and pallet library, mixed line processing can be realized on one production line. The three lines are composed of two double-station horizontal machining centers (FANUC system), two six-axis loading and unloading robots (wide number), one set of pallet exchange system, and one set of front and rear station conveying system. Box line is used for spindle box parts processing (a total of 8 different specifications and shapes of parts), through matching automatic tooling and pallet library, to achieve mixed line processing in a production line. The box line is composed of two double-station horizontal machining centers (FANUC system), one six-axis loading and unloading robot (FANUC), one set of pallet exchange system and pallet library. Large line for large parts production line is composed of bed processing unit, column processing unit, sliding saddle processing unit and other three independent units. Each processing unit is composed of two gantry composite processing centers, a moving column planer processing center, and the corresponding tooling and caching station[5]; All of them adopt FANUC system. The logistics line is to control the headstock and three logistics lines, small blank loading robot and finished product output logistics line. The IPC of a single production line is responsible for managing the equipment running status and parts processing status of the production line. The main controller is responsible for managing the IPC, distributing the processing tasks to the IPC and receiving the processing conditions of the tasks of each production line. The structure of the smart factory project is shown in the figure.

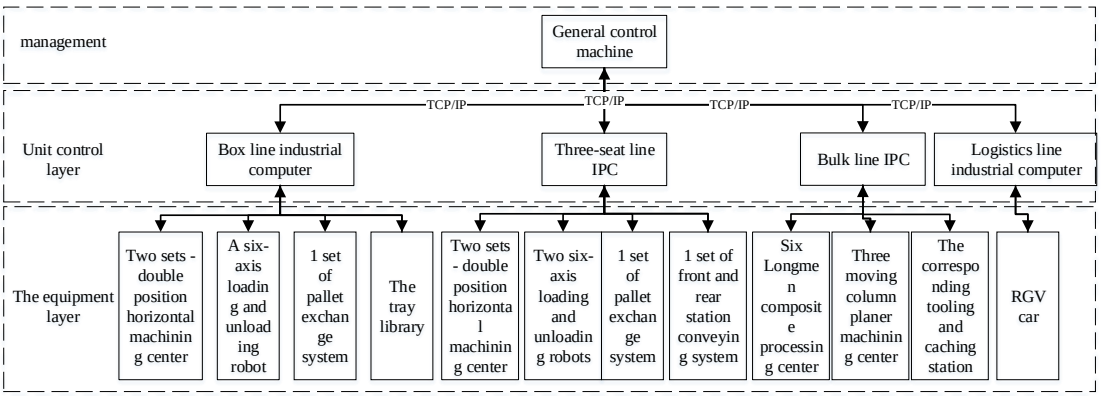

Figure 1 Overall structure diagram of intelligent factory

In the whole system, because each equipment needs to carry on the data transmission, and the way of transmission is also various. But now the data in the factory is transmitted through Ethernet, that is, data is based on TCP/IP transmission. Since the IP layer protocol of TCP/ IP cannot provide services directly to users, most of them use the socket network programming technology of Windows to provide services for users.

3. Socket network programming

Socket is designed for the client/server model and provides different Socket system calls for client and server programs. The establishment of the client is generally divided into five steps.

ISSN: 2664-9640

The first step is to establish a Socket. The second step is to call the connect function to connect with the server through the IP address and port number. The third step is to send data to the server through the send function when the connection is established with the server. The fourth step is to receive the data sent by the server through the recv function when the connection is established with the server. The fifth step is to close the Socket when the data transfer is complete or the client stops using it[6]. The Closesocket function is called at this point. The establishment of the server generally includes 7 steps. The difference between the server and the client is that the server has a binding process and a listening process. Its role is to listen to whether the client has sent a connection request. Also, instead of a connect function, there is an accept function to receive a connection from the client. In this way, the client and the server can transfer data. The data transmission structure of client and server is shown in the figure

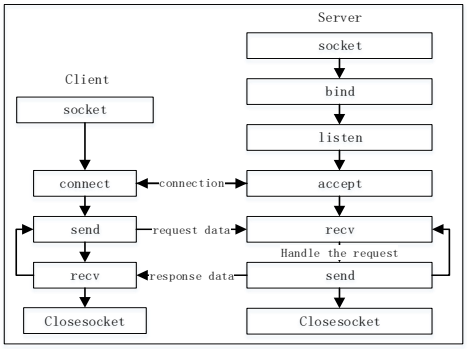

Figure 2 Client and server logic diagram

When a client accesses the server, the accept function is prone to the problem of waiting for messages. In general, the accept function processes socket functions in sequence. When there is no message for socket0 processed by the accept function, the accept function will wait until there is a message for socket0. Even if Socket2 has a message, the accept function does not process Socket2 with a message, as shown in the figure.

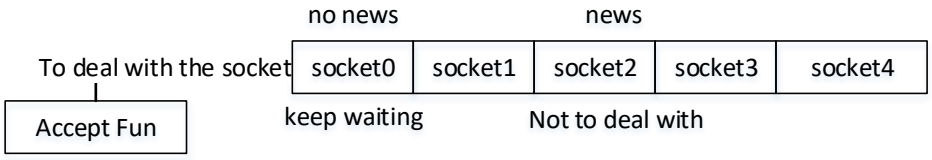

Figure 3 Accept function handling mechanism

When receiving data, the recv function of the server often copies the data into the cache area first and then receives the data from the cache area. If the amount of data is relatively large, there will be data blocking, affecting the transmission of subsequent data. As is shown in the figure.

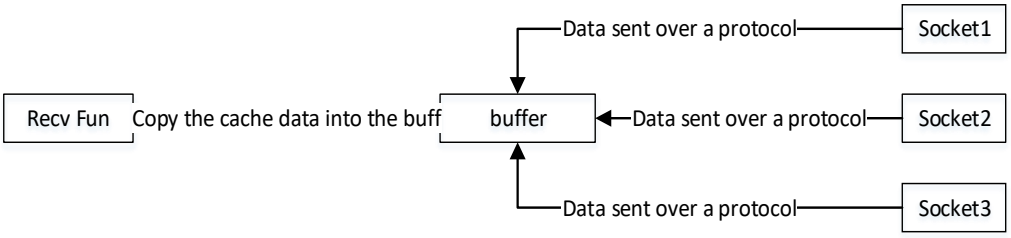

Figure 4 Recv function acceptance mechanism

To solve the problem of message waiting and blocking in socket, we propose several solutions.

ISSN: 2664-9640

4. Solutions

After analysis, we propose the following solutions to the problem of message waiting and blocking during the parallel transmission of socket multi-source data.

4.1. Overlapped IO Model

Overlapped IO model is a mechanism provided by Windows to read and write files asynchronously, and the essence of socket is to operate on files. In the overlapping IO model, faced with the problem of message waiting and blocking during the parallel transmission of multi-source data between the Accept function and the Recv function, the overlapping IO model delivers the messages with responses in the Accept function to the message queue, and the server takes out the messages from the message queue, while the main thread of the server is also processing the data at this time. So basically both the server and the operating system are processing the data at the same time. The message queue mechanism is FIFO, which saves a lot of time and improves the efficiency of the server to process the message. At the same time, the WSARecv function also processes the socket response message. The WSARecv function delivers the socket response message to the system, and the system creates a separate thread to copy the message to the cache, while the WSARecv function processes other messages, which greatly saves the time of message processing. Improve the efficiency of data transmission. The message transfer process of the overlapping IO model is shown in the figure.

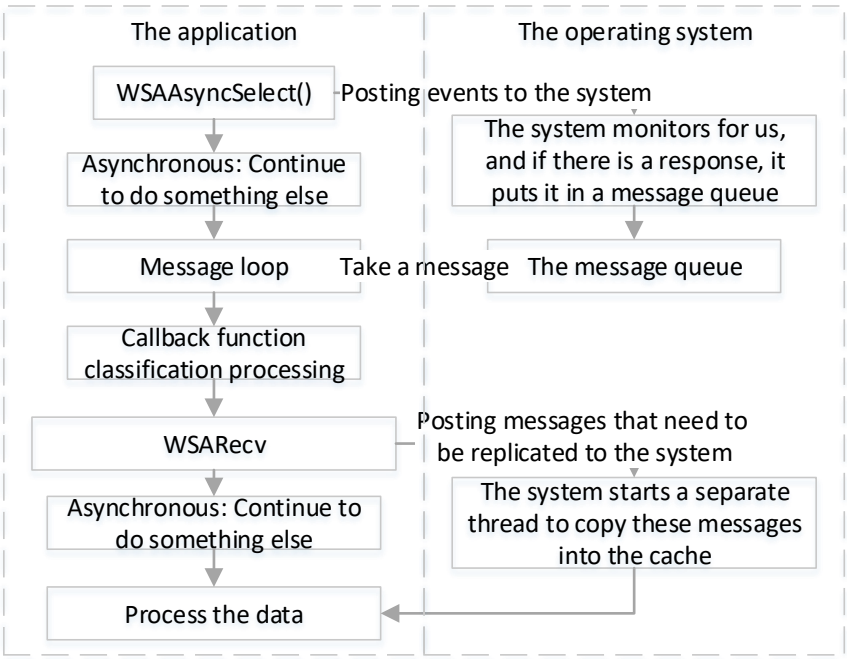

Figure 5 Overlapped IO Model Diagram

4.2. Multi-threading

Multiple threads are programs that run independently. This concept is applied to the project, that is, in the case of a server with multiple clients, a thread is established for each client connected by the accept function in the server, so that the recv function and send function only handle the message of a socket, and the data transmitted by each client can be independent. The logical relationship is shown in the figure.

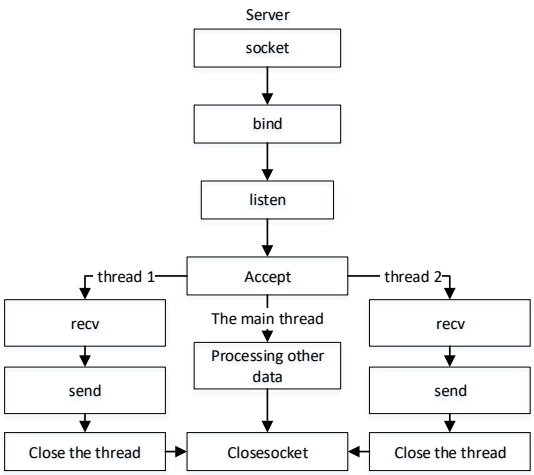

Figure 6 Multithreading Logic Diagram

5. Verification of scheme

In the project, the message transmission between the main control computer and 4 industrial computers is taken as an example. The main control computer serves as the server, and the 4 industrial computers serve as the clients. When the main control computer receives the processing task, the main control machine will send the processing task to each industrial computer, and the equipment under the control of each industrial computer will complete their tasks, and the completion of the task will be fed back to the main control computer. During this period, the data between the main control computer and each industrial computer is transmitted through Ethernet based on TCP/IP.

5.1. Overlapped IO Model

In the case of 1 server and 4 clients, the overlapping IO model can meet the needs of the project. But because a message queue is a container in which messages are held temporarily while they are in transit, the length of a message queue is limited. Moreover, the overlapping IO model not only delivers messages to message queues, but also creates new threads in the computer[7], so the consumption of computer resources is relatively large. When the amount of data is large and the client is more, the situation of stalling will occur.

5.2. Multi-threading

In the case of 1 server and 4 clients, multi-thread data transmission can ensure that the data transmitted by each client is independent of each other. And if a thread stops transmitting messages, it is suspended until a message is available. This can improve the utilization of computer resources, but also can be a good solution to the large amount of data, the client for many times the phenomenon of lag. Therefore, the multi-threaded solution is used in the project.

6. Summary

This paper studies the problem of multi-source data parallel transmission in the intelligent factory project of a CNC machine tool enterprise, and analyzes the causes of message waiting and blocking in the parallel transmission of multi-source data by means of Socket network programming. Two solutions are proposed to solve the problem of message waiting and blocking in the parallel transmission of multi-source data. Both of these solutions have been proven in this project. The results show that these two solutions can solve the problem of message waiting and blocking when multi-source data is transmitted in parallel. It is of

reference significance to solve the problem of message waiting and blocking when multi-source data are transmitted in parallel in intelligent factory.

References

[1] Li Xukan. Research and application of Modbus/TCP protocol [J]. Science and Technology Information,2019,17(02):121-123+125.
[2] Zhang Zhanyuan, Wen Xiaobin, Zhang Chao. Application and Fault Analysis of Modbus Field Bus [J]. Chemical Engineering Design Communication,2019,45(06):160-161.
[3] Sun Zhijuan. Research on Multi-robot Cooperative Communication Technology [D]. Guangxi University of Science and Technology,2019.
[4] LUO X. Research and Implementation of Security Technology of Industrial Fieldbus Modbus Protocol [D]. Jiangsu University of Science and Technology,2019.
[5] Gao Yang. Design and Implementation of Modbus-RTU Communication Method in AI Cluster Building [J]. Journal of Qinghai University,2020,38(02):58-65+80.
[6] Zhu Aman. Design of Warehouse Environment Monitoring System Based on Modbus/TCP Communication [D]. Central China Normal University,2019.
[7] Shao Xianlai, Zhao Jia. Application of Modbus protocol in smart home [J]. Integrated Circuit Applications,2019,36(02):58-59.