

The research of equipment simulation technology based on TCP

Xie Yang-zhong

Application Business Dept.
Comba Telecom Systems (China) Ltd.
Guangzhou, China
e-mail: 211xyz@163.com

Ye Bang-yan

School of Mechanical & Automotive Engineering
South China University of Technology
Guangzhou, China
e-mail: byye@scut.edu.cn

Abstract—According to the development of personnel positioning system, with the use of data storage technology, multithreading technology, information buffer technology and TCP communication technology, this simulator can not only simulate multidevice reporting to multiuser, but also realize pressure test for the upper software by adjusting the amount of reporting data, therefore, the software problems can be found in advance and solved in time.

Keywords- simulator; TCP; multithread; buffer

I. FOREWORD

Communication equipment simulator is often used to test the performance of hardware and software products by communication equipment manufacturing companies or network administration software development companies. The research and development of equipment simulator is an important part of the entire communication system. A good simulator can not only save cost for the company, but also simulate a variety of test scenes which can't use real equipment to establish them in practical work^{[1][2]}. Currently, the common practice is to develop proper simulator to simulate real equipment to test the upper software compression capability.

II. KEY TECHNOLOGY

A. Transfer Control Protocol

The development of network protocol is usually divided into different levels, and each layer is responsible for different communication functions. The TCP/IP protocol family is a combination of multiple protocols on different levels. TCP/IP is usually considered to be a four-layer system, as shown in Figure 1.

Application Layer(Telnet, FTP, HTTP, etc.)
Transport Layer(TCP and UDP)
Network Layer(IP, ICMP and IGMP)
Link Layer(Ethernet, Token Ring Network, etc.)

Figure 1. The Architecture of TCP/IP Protocol Family

The Transmission Control Protocol (TCP) provides reliable connection-oriented communication services for applications, suits for the applications which have response

requirement. TCP protocol ensures the reliability of end-to-end data communication through the following process^{[1][3]}:

(1) TCP entity divides the application into suitable data blocks, plus the TCP packet header to generate the data segment.

(2) The timer is started immediately when TCP entity sends out the data segment, if the source device still haven't received confirmation message from the destination device after the timer is cleared, the data segment will be resent. The peer TCP entity will return a confirmation message when the data is received.

(3) TCP contains an end-to-end checksum field to detect all the data change during transmission process. TCP will discard the data segment if checksum of the data received by the destination device and calculation results are different, retransmitted data segment, the source device will retransmit data segment after the above-described timer is cleared.

(4) Since the data bearer in IP packets, TCP and IP provides connectionless, unreliable service, the packets may be out of order. TCP provides a rearrangement mechanism which will reorder the received data for the application.

(5) TCP flow control. TCP has buffer window at each end of connection. Destination device only allows the source device to send data it can receive to prevent buffer overflow. The TCP supports full-duplex data transmission.

In view of the connection-oriented, orderly, reliable and full-duplex features of TCP technology, this communication mode is used between simulator and server.

B. Multithread Technology

Thread is a single sequential control flow in program, multiple threads run at the same time to perform different work in a single program called multithread. Multithread application divides program into multiple independent tasks, therefore, the performance can be improved significantly in following areas:

(1) Multithread technology enables the program faster response, because the user interface has been in active state while doing other work at the same time.

(2) The processor time can be given to other tasks when there is no processing task currently.

(3) The tasks which take up a lot of processing time can give the processor time to other tasks regularly.

(4) The task can be stopped at any time.

(5) The priority can be set for each task so as to optimize performance^[5].

In the simulator, in order to ensure concurrency of multitask running, make good use of system resources, the multithread technology is used when encountering processor-intensive tasks blocking the user interface operation or listening for external connections, and it achieves very good effect.

III. SYSTEM FRAME

This simulator system uses the Client/Server (Client/Server) architecture, as shown in Figure 2, which runs on the PC, packs the message according to the equipment communication protocol, combines with the database and uses TCP communication mode to send the information to the application server. Finally, it realizes data exchange between server and client as well as business processing. In addition, based on the framework, the distributed deployment of application server and client is achieved, the system flexibility improve markedly^[6].

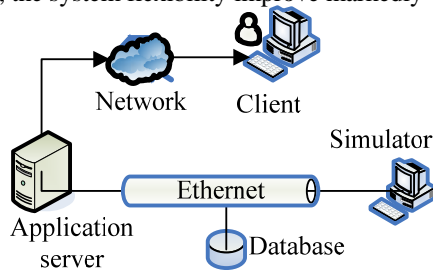


Figure 2. System architecture diagram

IV. DESIGN AND IMPLEMENTATION OF THE SIMULATOR

A. Design of the simulator

On the whole, this simulator can be divided into four modules: reporting data file record and configuration information database record module, data buffer module, protocol synthesis and analysis module and the TCP link module. Data sending periodically is achieved through using the multithread technology. As shown in Figure 3.

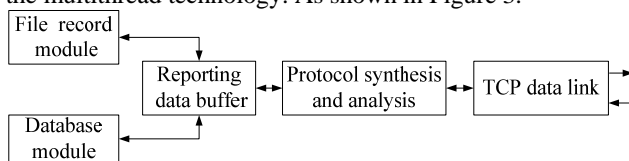


Figure 3. Simulator Architecture Diagram

The simulator adopts modular design which reduces the coupling among modules, improves module internal cohesiveness. Therefore, the system has better reusability, maintainability and extensibility. Different devices tend to use different communication protocols or different communication modes practically, which requires the simulator to have very strong flexibility and configurability in the internal modules switching. Therefore, the factory design pattern are applied in the protocol synthesis and analysis module and TCP data link module, which solves direct dependent relation in the module effectively, realizes the decoupling among modules.

B. Realization of the Simulator

As shown in Figure 4, during the simulator initialization, as the interactive medium of simulator and database, the database access class DatabaseAccess completes the loading of device configuration information while the file record class RecordFile realizes to read or save the device reporting data. Before starting to report, the TCP communication main class TcpMultiSvr will store the database information and file record data into the buffer SendBuffer first, thus in the later data reporting phase, link class RecordReport simply reads data from the buffer, which reduces the frequency of operating database or file, saves the consumption of computer resources as well as improves the reporting efficiency greatly^[4].

In order to prevent the multi-user concurrent operation of simulator public resources, such as database, log files and buffer, the singleton pattern is used in the classes of DatabaseAccess, RecordFile and SendBuffer, which achieves the mutual exclusion of resource operation. In addition, as the service end of TCP communication, the simulator needs to be started first to enter the monitoring state, waiting for the client connection. Special thread is created in the simulator^[5], which not only avoids "suspended animation" phenomenon caused by the jam effectively, but also keeps the uninterrupted monitoring of client connection, and realizes work mode of single server multiple clients in the simulator.

As shown in Figure 5, when the user starts simulator, firstly, in the simulator, the functions Read() in class RecordFile and LoadDbInfo() in class DatabaseAccess will be called to read the reporting data information and load device configuration information respectively. Secondly, through calling the function InitReport() in class TcpMultiSvr, the above data and configuration information will be synthesized and stored in the SendBuffer buffer via PushSendInfo() function, and the monitoring thread will be created to wait for the client persistent connections, thus achieving reporting initialization. Finally, at the start of data reporting, the reporting thread is also needed to be created to realize reporting. In this thread, at first, the initial report information is obtained by calling PopSendInfo() function in class SendBuffer, then according to the equipment communication protocol, the byte stream packets are synthesized with the calling of function Encode() in class Protocol, and ultimately the information transmission is completed with TCP communication mode. In addition, users can modify data reporting period or the number of message reported during the period according to requirement, thus realizes to adjust the testing pressure for upper layer software.

V. CONCLUSION

The project practice of the personnel positioning system shows that based on data reporting of simulator, the R&D personnel can predict the surviving deficiencies and potential problems for upper software, which realizes early detection and early settlement of problems, greatly improves the software robustness. Under the situations when it's difficult

to establish the environment with real equipment or it doesn't have the conditions, the simulator as an alternative is simple and efficient, project cost reduces obviously.

REFERENCE

- [1] Wang Zhen-Ti. Design and realization of communication equipment simulator based on UDP protocol[D]. NanJing: Southeast University, 2005.
- [2] Liu Shuang, Shi Guo-You, Zhang Yuan-Qiang. The design and realization of communication software based on TCP/IP protocol and multithread[J]. Computer Engineer and Design, 2010,31(7):1417-1420.
- [3] Zeng Juan, Yang Hong-Min. The design and development of network device simulator based on the B/S[J]. Modern Electronic Technology, 2008, (11):156-158.
- [4] Wang Qing-Qing. The UML2.0 Learning Guide [M]. Beijing: Tsinghua University Press, 2007.
- [5] Hao Wen-Hua, Wen Zi-Yong, Wang Hao-Qiang, etc. Windows multithreaded programming techniques and examples of [M]. Beijing: China Water Power Press, 2005.
- [6] Liu Shuang, Jia Chuan-Ying, Chen Peng. The design and realization of vehicle monitoring system based on C/S[J]. Computer Engineering and Design, 2005,26 (12) :3320-3323.

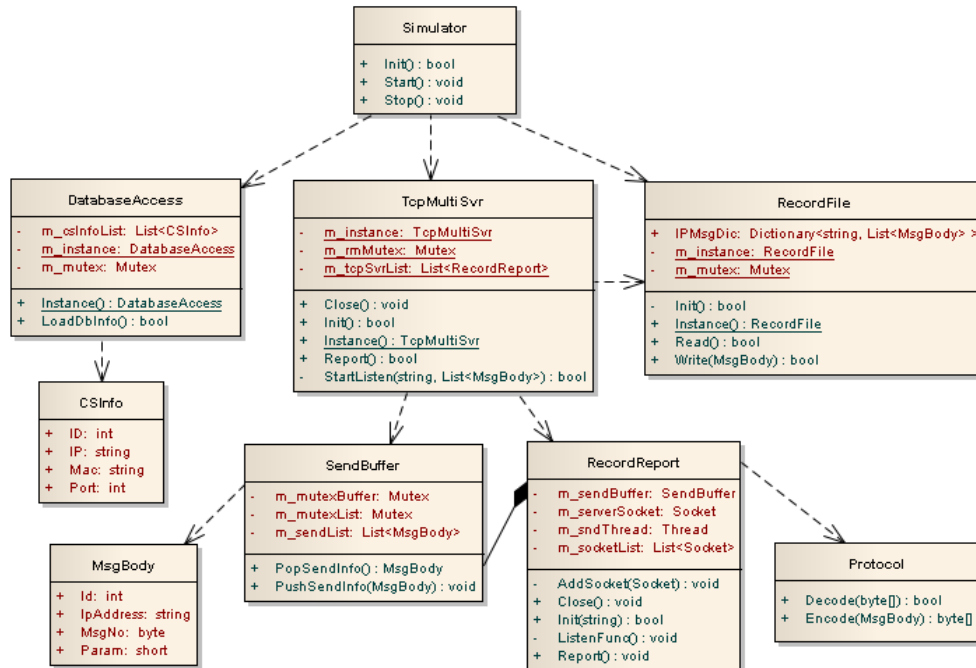


Figure 4. Simulator Class Diagram

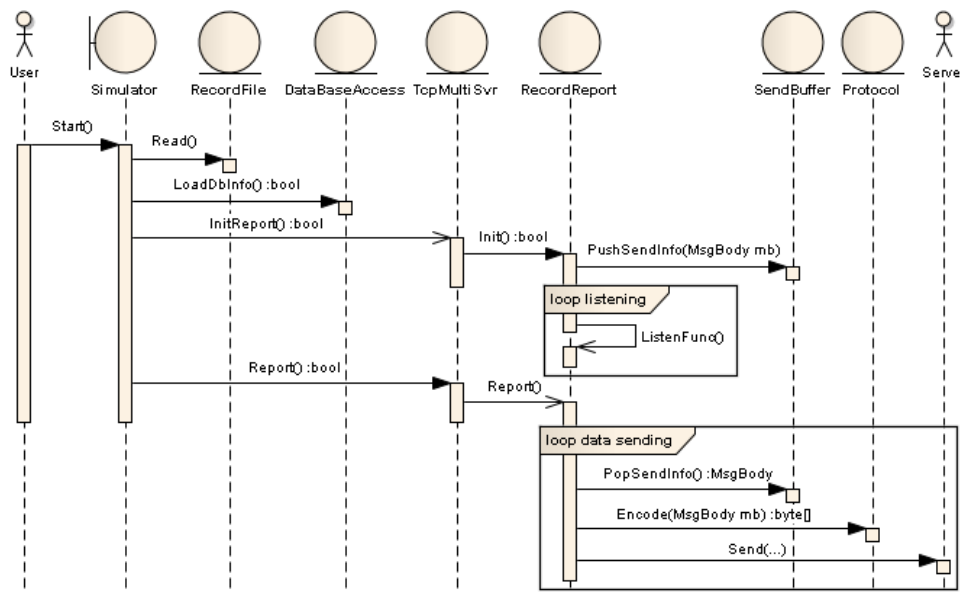


Figure 5. Simulator Sequence Diagram