Researches on the Analysis Framework of Application Layer Communication Protocol Based on SQLite

Wenyuan Xu[1], Hao Li[1], and Weifeng Xu[2(✉)]

[1] China Shipbuilding Industry Systems Engineering Research Institute, Beijing, China
xwy0987@sina.com
[2] School of Control and Computer Engineering, North China Electric Power University, Hebei, China
weifengxu@163.com

Abstract. Since the concept of mobile Internet, big data and cloud computing has been proposed, the format of the information undergoes a tremendous change, information gradually appear problems in large quantities, the complexity and the data form is not fixed. This is not a small impact and challenge to the traditional fixed form protocol analysis. Dynamic processing and analysis of data is very important for the data processing flexibility, easy scalability and better fault tolerance. Dynamic is mainly reflected in the dynamics of dynamic definition, dynamic analysis and dynamic processing. In the dynamic analysis of the data, you don't need to set the specific format of the data in the program, only the framework of dynamic analysis need to be constructed in the sending and receiving programs, the programs can read the format of data automatically, and get the content of data easily. This approach can make the program more suitable for the current application in size and flexibility than the traditional form of a fixed communication protocol analysis. Redundant structures does not need to add into the programs, which is no longer processing data passively. Data transmission becomes more convenient, because the data format needn't to be token care when the administrator wants to transfer the data.

Keywords: SQLite · Communication protocol · Dynamic analysis
Message buffer

1 Instruction

At present, the format of messages transmitted between application processes on different terminal systems is mostly defined in programs, and it can't be changed when users transmit messages with application processes, the flexibility of dynamically define message format is lacked [1]. Once a user modified the message format, changes to the format of messages in code layer by program developers would be required, and part of communication procedure of application would need to be retested, this process will consume a lot of manpower and material resources [5]. The format of the information undergoes a tremendous change, information gradually appear problems in large quantities, the complexity and the data form is not fixed. This is not a small impact and challenge to the traditional fixed

Q. Zu and B. Hu (Eds.): HCC 2017, LNCS 10745, pp. 150–157, 2018.
https://doi.org/10.1007/978-3-319-74521-3_17

form protocol analysis. Dynamic processing and analysis of data is very important for the data processing flexibility, easy scalability and better fault tolerance.

The analysis framework of application layer communication protocol based on SQLite [6], which is proposed in this paper, presents the format of message defined in the program in the form of database table, users can configure the table flexibly according to needs, and dynamically analyze it when in using. The whole framework contains protocol development process, dynamic object generation process, protocol analysis process, message combination sending process, message receiving process, data stored procedure, communication process. Dynamic is mainly reflected in the dynamic definition of communication protocol, dynamic analysis and dynamic processing. In the dynamic analysis of the data, you don't need to set the specific format of the data in the program, only the framework of dynamic analysis need to be constructed in the sending and receiving programs, the programs can read the format of data automatically, and get the content of data easily. This approach can make the program more suitable for the current application in size and flexibility than the traditional form of a fixed communication protocol analysis. Redundant structures does not need to add into the programs, which is no longer processing data passively. Data transmission becomes more convenient, because the data format needn't to be token care when the administrator wants to transfer the data.

In view of the SQLite memory database belongs to the lightweight database, with superiorities such as less resource consumption, portability, security, reliability, zero management costs and so on, we will use the SQLite memory data as the base carrier to implement the message format storage and high-speed information buffer in this paper.

2 SQLite Database Technology

SQLite memory database is an embedded database engine. Specifically suitable for appropriate data access on a variety of equipment with limited resources (such as mobile phones, pads and other intelligent devices) [3, 7]. It follows the ACID relational database management system, its design goal is embedded, and now has been applied in many embedded products, and its source code is abundant on the official website. SQLite memory database takes up very few resources, the memory is only occupied in the order of 100 k bytes in the embedded device [8]. It can be supported by mainstream desktop operating systems like Windows/Linux/Unix and all the mobile operating system platforms, and can be combined with plenty of programming languages, such as C#, C/C+, PHP, Java and ODBC interface, and it is faster than MySQL and PostgreSQL, which are two of the world famous open source database management systems [4]. The SQLite supports the most of the SQL92 syntax, and allows developers to use SQL statements to manipulate data in the database, and the SQLite is just a file, that doesn't need to be installed or started the server processes as databases like Oracle and MySQL. It has the following characteristics:

(1) Lightweight: The SQLite is a built-in database, all the database operations can be completed by it with a dynamic library.
(2) Portability: The SQLite can be run in a variety of environments, we can see from the source code provided by the official website, not only the desktop side, but also the mobile side is covered by the mainstream platform.

(3) Security and Reliability: When a certain data need to be accessed by multiple processes simultaneously, data in the database can be read but cannot be written at the same time by these processes, which ensures the reliability of the data.
(4) Green software: Another feature of SQLite is green: its core engine doesn't rely on any third-party software, and installation does not be required. So a lot of trouble can be saved at the time of deployment.
(5) Easy to manage: The various data information in SQLite involve graphics, tables and other files are isolated from each other, to ensure the mutual interference would not take place, and also facilitate the operations on the database.

3 Overall Structure of the Application Layer Communication Protocol Analysis Framework

3.1 Architecture Design of Analysis Framework

Figure 1 shows the architecture design of the application layer communication protocol analysis framework. In this framework, the definition between the sender and receiver is very vague, so we do not deliberately distinguish the sender or receiver.

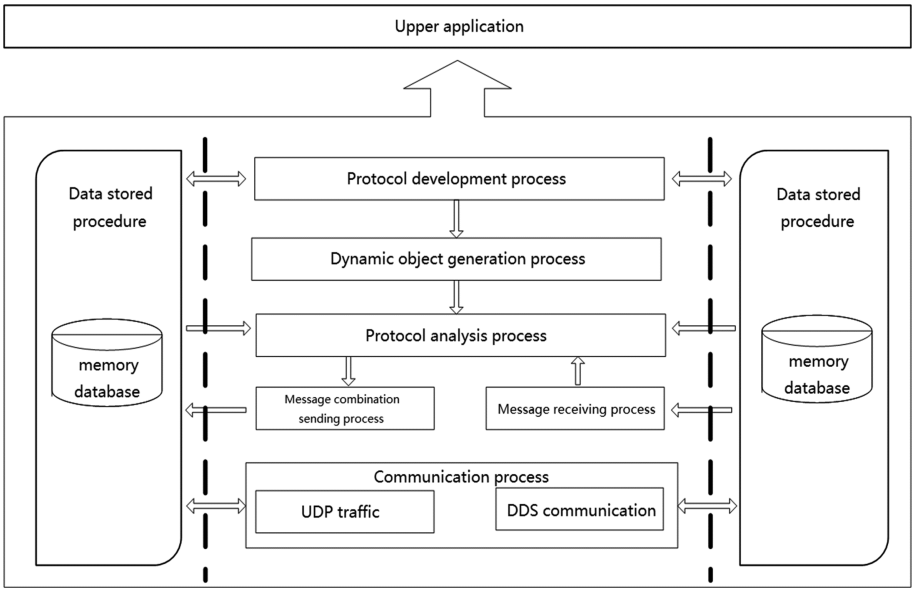

Fig. 1. Overall structure of the application layer communication protocol analysis framework

Firstly the framework provides the user with a data table that can be created and modified, and this table stipulates the format in which the protocol is created. The protocol can be analyzed by software after it has been created, that is, getting into the protocol analysis process. During the protocol analysis process, the object of each

attribute in the protocol is generated by starting the dynamic object generation process. The message combination sending process should be started if the simulation software needed to send data after the protocol has been analyzed.

During the message sending process, the UDP or DDS [2] interface in communication process will not be called directly for data transmission, but starts the data stored procedure firstly, stores the data to be sent into the memory database, and then throw a specific message to the communication process, which initiates the send function after receiving this message, retrieves the data to be sent from the memory database and sends it to the destination. If a message sent by the external software is received by the communication process of the simulation software, this message would not be analyzed but the raw string should be stored in the memory database firstly after the UDP or DDS communication interface receives the information, and then a specific message should be thrown by protocol analysis process which retrieves the raw string data from the memory database and initiates the protocol analysis function to complete the analysis of the message after receiving it. In the analysis process, message is associated with the specific agreement according to the information identification, and finally completes the analysis of information according to the protocol. The data which have been analyzed would be stored in the memory database again after the analysis process, and then a specific message should be sent to inform the business application layer to complete the corresponding business functions, such as display, calculation and so on.

3.2 Communication Process Based on Framework

The basic communication process of the software based on framework implementation has been shown in Fig. 2. As can be seen from the figure, both the data sending and

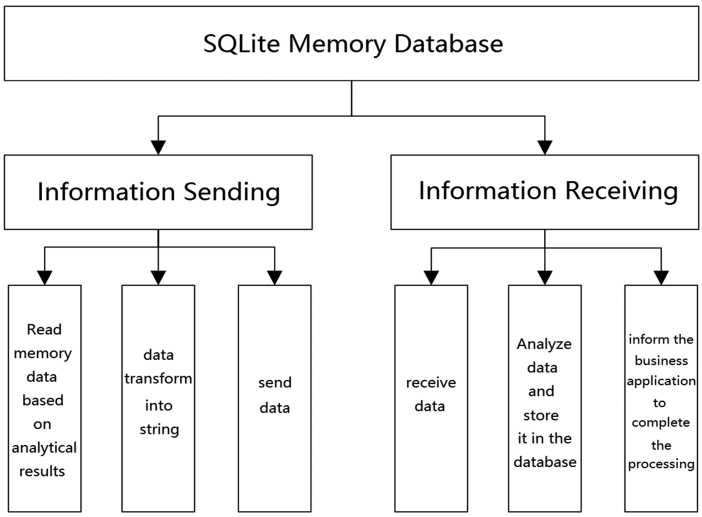

Fig. 2. Information sending and receiving process

receiving is centered on SQLite memory database. Taking memory data as the center is an another important feature of the framework proposed in this paper.

4 Analysis Method of Application Layer Communication Protocol

4.1 Protocol Configuration Process

In the analysis method of application layer communication protocol, information is configured according to the information format, and the received information is identified to accomplish the analysis of the received information and the formatted storage of the information, and the analysis method has the function of calling the business application in real time through the service manage center. Users configured by the communication protocol can configure through the configuration files dynamically, the real-time information and business application scheduling relationship users can also configure dynamically through the configuration file.

The purpose of configuring the information protocol is that the information protocol can be dynamically configured by the software developer. The framework can analyze the received information according to the identity of the configuration pair and the message after the information protocol is configured in database. The configuration of the information protocol is configured by the developers of the simulation software. The configuration is as follows:

(1) Information index configuration
Database name: Config.db
Database path: main_project_directory\Bin\Win32\Config
Database table: MessageDictionary

When configuring the information protocol, the basic information in the information index table should be configured at first. As shown in Table 1, RecNo in the configuration table is the information number automatically generated by database; ID is the protocol number (not repeatable); Name is the information protocol name; and ReMark is the information protocol description.

(2) Information protocol configuration
Database name: StructMessage.db
Database path: main_project_directory\Bin\Win32\Config
Database table: MessageDictionary

Table 1. An example of the protocol configuration

ID	EName	Srartindex	Byte count	Type
1	InFoN	0	16	Char[]
2	InFoId	16	1	Char[]
3	ShipGauge	24	4	FLOAT
4	WindSpeek	28	4	FLOAT
5	WindCourse	32	4	DOUBLE

Each information establishes a protocol configuration table which is named by the name of the information protocol (name of the Name item) in the MessageDictionary table.

As shown in Table 1, RecNo in the configuration table is the protocol field number automatically generated by database; ID is the field number in the protocol (not repeatable); EnName is the English name of field; startIndex is the number of the first byte of the field in the entire protocol, the byte of protocol is numbered from "0"; ByteCount is a count of bytes occupied by the field; Type is the field type; Rule is an additional processing instructions of the field; and ReMark is the field description.

4.2 Protocol Analysis Implementation

(1) Dynamic generation of the object:

```
ClassCreator::ClassCreator(constchar *cName, CreateClass cc)
{
  static std::map<std::string, CreateClass> s_classMap;
  pMap = &s_classMap;
  map<std::string, CreateClass>::iterator it = pMap->(className);
  if (it == pMap->end())
  {(*pMap)[className] = cc;}
}
```

This code is a dynamic generation of the object, and the object needs to be generated before dynamically analyzing in order to call the analysis process.

(2) Process of dynamic analysis

```
bool CStructMessage::ParseMessage(unsigned char* pBuf)
  {
    for(int i = 0; i < m_StructMessageVec. size(); i++)
    {m_StructMessageVec[i]->SetValue(pBuf);}
    return true;
  }
```

Through using a for-loop on valid data in the database, this code analyzes each segment of data into an available value, and combines this values into a valid string for transmission in the next step.

5 Message Buffer Design

The whole system is divided into three modules by function in order to implement the message buffer: first, Socket communication module, the module is mainly used to accept and transmit messages which require to be buffered, such as string, structure, etc.; second, SQLite message access module, the module is to access messages, which need to be buffered, in the SQLite database; third, the forwarding module of Windows message mechanism, the module generates a message reminder in the process and then distributes it; The overall structure is shown in Fig. 3.

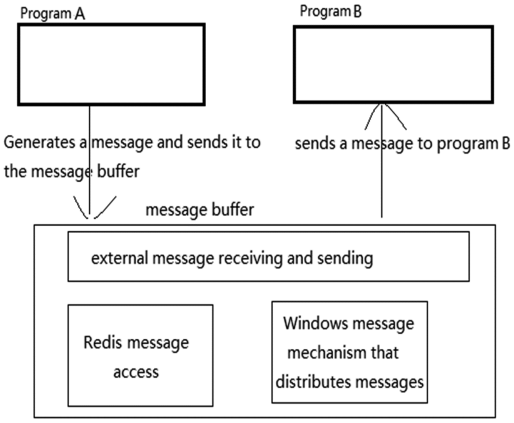

Fig. 3. Structure of the message buffer

In the implementation of the message buffer, communicating between programs is needed, so the Socket network communication technique has been used. Messages are transmitted through the sockets between the program A and the message buffer. Sockets programming can be used in three ways, streaming socket (SOCK_STREAM), datagram socket (SOCK_DGRAM), raw socket (SOCK_RAW); this program is designed based on UDP socket programming uses streaming sockets. The following describes the specific implementation of the program: The programing steps of the sender of program A: 1, Load the socket library, and create the socket (WSAStartup()/socket()); 2, send the connect request to the receiver of buffer (connect()); 3, communicate to the receiver of the message buffer (send()/recv()); 4, close the socket, and close the loaded socket library (closesocket()/WSACleanup()).

When the message has been received by the message buffer through the socket, it will be stored in the SQLite database according to the type, the stored messages has different types, so the SQLite message storage operations are not same, too. Step 1: load/release the Winsock library; Step 2: connect to the SQLite; Step 3: different storage operations are performed according to different message types.

When the message has been stored in the SQLite database, the system will send message reminder to different windows based on the different types of messages, which involves the windows message mechanism and dialog window design with MFC, and this will be introduced in next two sections. Messages will be processed by MFC window after the window receives a message reminder, but due to the limitation of design, the message processing in this experiment is simply simulated, which selects messages from SQLite database and displays them to the MFC Edit controls.

6 Conclusions

The analysis framework of application layer communication protocol based on SQLite, which is proposed in this paper, presents the format of message defined in the program

in the form of database table, users can configure the table flexibly according to needs, and dynamically analyze it when in using.

The framework that is proposed in this paper can make the program more suitable for the current application in size and flexibility than the traditional form of a fixed communication protocol analysis. Based on the framework we proposed, redundant structures does not need to add into the programs, which is no longer processing data passively. Data transmission becomes more convenient, because the data format needn't to be token care when the administrator wants to transfer the data.

However, this paper only uses one type of database, SQLite, and only implements one kind of communication mode, UDP, so that the application scope of this framework is still limited. In the future work, increasing the type of database and expanding the communication model will be an important way to develop this framework.

References

1. Harrison, T., Pyarali, I.: An object behavioral pattern for demultiplexing and dispatching handlers for asynchronous events. In: 4th Pattern Languages of Programming Conference in Allerton Park, Illinois, 2–5 September 1997 (1997)
2. An, K., Gokhale, A.: A cloud-enabled coordination service for internet-scale OMG DDS applications. In: Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, New York (2014)
3. Hunt, P., Konar, M., Junqueira, F.P., Reed, B.: ZooKeeper: wait-free coordination for internet-scalesystems. In: Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, vol. 8, p. 11 (2010)
4. Li, M., Ye, F., Kim, M., Chen, H., Lei, H.: A scalable and elastic publish/subscribe service. In: 2011 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 1254–1265 (2011)
5. Mei, H., Shen, J.-R.: Progress of research on software architecture. J. Softw. **17**(06), 1257–1275 (2006)
6. Lin, M.: Design and Implementation of a Personal Address Book Management System Based on SQLite. Jilin University (2015)
7. Tong, S.: Study on Application of Mobile Meter Reading Technology Based on SQLite. Jilin University (2015)
8. Yang, X.: Design and Implementation of Beidou Navigation System Based on SQLite Database. Lanzhou University (2015)