Discussion of Network Programming

Deen Chen

Computer Science Department, North China Electric Power University, Baoding 071000, China

906062523@qq.com

Abstract

Network communication is a significant technology in current Internet technology, so network programming technology has become the core of current Internet communication development. The socket, which is built on the transport layer protocol, can solve the problem of communication between networks. This paper gives a detailed elucidation about the network programming technology based on socket. Firstly, this article introduces the TCP/IP and UDP, and points out their communication process, then use this as a basis to classify sockets from an application perspective. Secondly, two kinds of Winsock programming models are described based on the C/S model.

Keywords

TCP/IP &UDP Protocol; Socket; Winsock Programming Model.

1. Introduction

The network application is at the top of the computer network architecture network hierarchy, and can be functionally divided into a communication module dedicated to network communication and a user-oriented user interface module and data processing module. Among them, communication is the problem that needs to be solved first. The data transmission on the network requires the guarantee of the network protocol. The commonly used TCP/IP protocol establishes an end-to-end connection, realizes the seamless connection of various heterogeneous networks and the cross-platform transplantation of applications, and adopts a complicated mechanism to ensure data to be complete and reliable.

A port, which uniquely identifies a process, is a communication protocol interface between an application layer and a transport layer. It can be divided into TCP port and UDP port. MFC encapsulates the Winsock API and uses the message-driven mechanism of the Windows system to receive and send data using sockets. Sockets are created on idle ports and can be divided into TCP interfaces and UDP interfaces.

This paper will receive detailed network programming based on TCP/IP protocol stack, and build a connection-oriented and connectionless programming model in Winsock programming based on sockets to deepen the communication module, socket and I/O mode.

2. TCP/IP and UDP

Heterogeneity exists in each computing node in a computer network. If mutual communication is achieved, a common protocol model needs to be followed. The International Organization for Standardization (OSI) has established a seven-layer model of the Open Systems Interconnection to address communication problems between nodes. The model is divided into application layer, presentation layer, session layer, transport layer, network layer, data link layer and physical layer. The concept is clear and the theory is complete. TCP/IP is a four-layer architecture that includes an application layer, a transport layer, an internet layer (to solve interconnection problems of different networks), and a network interface layer. Due to its simplicity and convenience, the four-layer protocol is widely used in real life.

TCP is the Transmission Control Protocol - Provides connection-oriented, reliable data transmission services, the unit of data transmission is the segment of the message, located at the transport layer.

UDP is the user datagram protocol - Provides a connectionless, best-effort data transmission service (the reliability of data transmission is not guaranteed), and its data transmission unit is user datagram, which is located at the transport layer.

IP protocol - The network layer is responsible for providing communication services for different hosts on the packet switched network. When transmitting data, the network layer encapsulates the segments or user datagrams generated by the transport layer into packets or packets.

IP address - A worldwide 32-bit unique identifier assigned to each interface of each host on the Internet.

Port - The abstract protocol port between the protocol stack layers. The end of the communication is the application, and the port can specify which process the message should be handed over to. A port number is marked with a 16-bit port number, and it has only local significance.

TCP is a connection-oriented transport layer protocol. Each TCP connection can only have two endpoints. Such a port is a socket. Three handshaking is required when establishing a connection, and four times when disconnecting.

The User Datagram Protocol UDP adds multiplexing and demultiplexing and error detection functions to the IP datagram service. It has the characteristics of no connection, best effort delivery, and message-oriented. When the transport layer receives the UDP datagram from the IP layer, it passes the UDP datagram through the corresponding port according to the destination port in the header, and hands over the last destination - the application.

3. Socket

3.1 socket definition

A socket is a basic operating unit that supports TCP/IP network communication. It is an abstract representation of an endpoint in a network communication process. It is represented by (IP address: port number) and is a handle of a communication chain. Each TCP connection is uniquely determined by two endpoints (two sockets) at both ends of the communication. Sockets use inter-network communication facilities for process communication, but do not care about communication facilities, only the communication facilities have sufficient capabilities. Before the process communicates, both parties need to establish their own sockets, otherwise they cannot connect and communicate.

3.2 Classification

The types of sockets are: streaming sockets, datagram sockets, raw sockets.

Streamed sockets are a type of socket represented by the TCP protocol that provides data flow services in both bidirectional, ordered, non-repetitive, and unrecorded boundaries. Before exchanging data, first establish a transmission link, and the transmitted data will be transmitted along the determined path to ensure the orderly arrival of the data. This socket relies on calculations to ensure the correctness of the data.

Datagram sockets are represented by the UDP protocol, which supports bidirectional data streams but does not guarantee reliable, orderly, and non-repetitive data arrival. An important feature is the ability to preserve record boundaries. The so-called reserved record boundary means that the transport protocol transmits the message as an independent message. Datagram sockets have a certain limit on the length of the data, which is specified to be a maximum of 64KB.

The raw socket runs access to the underlying protocol, and TCP/UDP type sockets only have access to the transport layer and data above the transport layer. The main uses of the original socket are:

sending a custom IP datagram; sending an ICMP datagram; listening mode of the network card, listening for data packets on the network; disguising the IP address; implementing a custom protocol.

4. Winsock programming model

4.1 C/S model

The C/S model is the client/server model, a common way to build in distributed applications. The service program listens for requests for services at a fixed address. When a client makes a connection request for the address of the service, the service program is "awakened" and responds appropriately to the client's request.

Winsock programming is generally based on a client/server model implementation, divided into a connection-oriented programming model and a connectionless programming model.

4.2 Connection-oriented programming model

In the connection-oriented programming model, the connection should first be established before communication. The server should create two sockets, a listen socket and a connection socket. The communication process is: (1) Create a socket. The service process always starts before the client process. The server first uses the socket function to create a listening socket. (2) Address binding. Call the bind function to bind the local address to the socket. (3) Enter the interception. Call the listen function to listen and prepare the accept function to handle the connection request at any time. (4) The client creates its own socket. (5) The client uses the socket to call the connect function to issue a connection request. (6) The server receives the connection request, and the blocked accept function generates a connection socket to connect with the client. (7) The client receives the receiving signal from the server. At this point, the connection process is completed and both parties can send a message. (8) Send information using the send function. (9) Receive messages using the receive function. (10) Close the socket. When the data is sent and received, the close socket function can be called to close the socket and release the resources occupied by the socket.

4.3 Connectionless programming model

In the connectionless programming model, the client and server roles are interchangeable in this mode because of no connection. Its transmission efficiency is high but it is not responsible for the order and correctness of the data.

The connectionless programming model communication process is: (1) use the socket function to create a socket. (2) Both parties call the bind function to bind their respective addresses, and bind the socket to the specified network address. (3) Send data using the sendto() function. This call needs to determine the two endpoints of the communication, that is, a fully related quintuple. (4) Call the recvfrom() function to receive data from the datagram socket. The first datagram will be fetched from the receive buffer queue of the socket and placed in the buffer buf of the user process. (5) Close the socket. Call the closesocket() function to close both sockets and release resources.

5. Conclusion

This article takes the TCP/IP and UDP protocols as the starting point, and then introduces the socket types and socket programming models for different protocols. Network programming is an important way for the application network to play its value, and mastering the SOCKET network programming technology can achieve the effectiveness of network communication. Sockets are the basis for communication and are the basic interface for supporting network protocol data communication. Using socket programming can make a good communication connection. Its emergence also makes it extremely convenient for programmers to simply call functions when writing web applications.

References

[1] Liu Wei.Application of Winsock Technology in Network Communication System[J].Journal of Southwest University of Science and Technology,2013,28(02):88-91.
[2] Yan Hui. Java-based socket programming [J]. Computer Knowledge and Technology, 2016, 12 (20): 104-105.
[3] Song Zerui. Network Communication Based on SOCKET Programming Interface[J]. China New Communications, 2017, 19(05): 29.
[4] Zhang Xuekun. Design and implementation of network chat program based on Socket [J]. Computer programming skills and maintenance, 2018 (04): 16-17+24.
[5] Yan Qian, Yang Yong. Network programming tcp / ip protocol and socket discussion [J]. Electronic World, 2016 (08): 68 + 70.
[6] Chi Ying. Application of Select Model to Implement TCP Concurrent Server [J]. Science and Technology Plaza, 2006 (04): 30-32.