Research on C/S Protocol and P2P Protocol in Wireless Communication

Geng Yue
School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Haidian District, Beijing, 100876, China
GengYue@bupt.edu.cn

Abstract: This paper mainly studies how the two communication protocols c/s protocol and p2p protocol are implemented and applied in the communication system. The traditional c/s protocol and p2p protocol are two widely used communication protocols. This paper studies how these two communication systems are implemented, and imagines how the two protocols work together. This paper first studies the development and characteristics of the two protocols, and then studies the structure of the two communication systems separately, how to implement the two communication systems separately and explore how to apply them in practice.

1. Introduction
The traditional C/S architecture, is the Client-Server (C/S) architecture. The C/S structure usually takes a two-layer structure. The server is responsible for the management of the data, and the client is responsible for completing the interaction tasks with the user. The client connects to the server through the local area network, accepts the user's request, and makes a request to the server through the network to operate the database. The server accepts the client's request, submits the data to the client, and the client computes the data and presents the result to the user [1]. In the C/S structure, the application is divided into two parts: the server part and the client part. The server part is information and functions shared by multiple users, executing background services, such as controlling the operation of the shared database; the client part is proprietary to the user, responsible for executing the foreground function, and has powerful functions in error prompts, online help, etc. Function and freely switch between subroutines [3].

Peer-to-peer (P2P), is an Internet system that exchanges information without a central server and peers. Its role is to reduce the network. Nodes in transit to reduce the risk of data loss. Different from the central network system with a central server, each client of the peer-to-peer network is both a node and a server. Any node cannot directly find other nodes, and must rely on its household group for information exchange.

An important feature of P2P networks is that each Peer can communicate directly with each other, at least. P2P builds a P2P-compliant network on existing networks, but is not limited to existing network protocols. An important goal of a P2P network is to provide resources to all clients, including bandwidth, storage, and computing power. Therefore, when a node joins and requests for the system increase, the capacity of the entire system also increases [10]. This is not possible with a Client-Server architecture with a fixed set of servers, because in this configuration, the increase in client means slower data transfer for all users. The distribution characteristics of P2P networks also increase the robustness of fail-safe by replicating data on multiple nodes, and in pure P2P networks, nodes do not
need to rely on a central index server to discover data. In the latter case, the system will not experience a single point of collapse [9].

The c/s architecture and the p2p architecture have their own characteristics. The advantage of the c/s structure is that it can fully utilize the processing power of the client pc. A lot of work can be submitted to the server after processing by the client. The corresponding advantage is that the client response speed is fast. Specifically, the application server runs with lighter data load and the data storage management function is more transparent [2]. However, in this model, customer service depends on the server. As the information grows and the number of customers increases, the pressure on the server becomes larger. When the entire network reaches a certain scale, the server will cause resource bottlenecks and bandwidth to the entire network. The bottleneck limits the expansion of the network [3].

The P2P protocol allows each Peer to communicate directly with each other without going through the server. In the P2P network environment, each node is both a server and a client, which reduces the computing power and storage capacity requirements of the traditional C/S structure server. At the same time, because the resources are distributed in multiple nodes, the load of the entire network is better realized. Equilibrium [8]. But P2P communication technology will bring a "full distribution" network of services. The traffic will show greater arbitrariness, and the direct data exchange between users will be more frequent, which has higher requirements for network management [7]. The P2P architecture [5] can be implemented more conveniently using the JXTA protocol. JXTA is a network computing platform developed by Sun for P2P applications. JXTA is not an application, but a framework with a set of standards that support P2P applications. This framework contains a set of protocols that provide services and foundations for P2P applications that are independent of programming languages, system platforms, and network platforms, making it easy to construct P2P applications simply and easily [4]. JXTA provides the basic components for developing P2P-based applications that are operating system-independent, language-independent, and run on any device, thus enabling development on different platforms. JXTA is also network-independent, and the JXTA protocol can be transported using TCP/IP, HTTP, Bluetooth, Wi-Fi, home networking, etc. [5]. Peers on different networks can easily communicate using the standard JXTA protocol. The underlying layer of JXTA is also the Socket communication mode.

This paper envisages that when most machines need to work together, the information of each machine can be stored on the server side through the c/s structure, such as the IP address and port number of each machine, for distribution to each machine. Each machine can request information from other servers to connect to the server, and then communicate with each other through the p2p mode, thereby improving the efficiency of mutual communication and cooperative work. The following describes how the two communication protocols are implemented and how they are applied.

2. Implementation of communication system based on c/s architecture

2.1 Socket communication

A socket is an abstraction layer through which an application can send or receive data, and can be opened, read, written, and closed as if it were a file. Sockets allow applications to plug I/O into the network and communicate with other applications on the network. A network socket is a combination of an IP address and a port. Socket abstracts the complex operations of the TCP/IP layer into several simple interface supply layer calls to implement the process communication in the network. Socket is an implementation of the "open-read/write-close" mode, the server and the client respectively Maintain a "file". After the connection is established, you can write the content to your own file for the other party to read or read the other party’s content, and close the file when the communication ends.

The C/S communication structure can be realized by using Socket communication. This paper studies the socket communication using TCP protocol communication, its interaction process: TCP protocol: three-way handshake protocol (server end accept, customer service connect), four waved (client close, server close) server end accept blocked, waiting The link of the (multiple) client
terminals; the general link and communication flow is as follows: read data from the Socket, open the file input stream from the Socket, and read the data from the input stream. If the Socket has data to read the data, the Socket has no data. Read blocking.

The specific implementation process is discussed below.

2.2 **C/S structure flow based on Socket communication:**

2.2.1 **server side** Socket communication requires the server to start first, and provides services to the client according to the request. The workflow is as follows:

1. opening a communication channel and informing the local host that it is willing to receive a client request at a port on a recognized address;
2. waiting for a client request to reach the port;
3. When receiving a service request from the client, the request is processed and a response signal is sent. Upon receiving a concurrent service request, a new process is activated to process the client request. The new process processes this client request and does not need to respond to other requests. After the service is completed, the communication link between the new process and the customer is closed and terminated.
4. Return to step (2) and wait for another customer request.
5. Shut down the server

The process can be represented by Figure 1:

![Figure 1 Socket communication server-side process](image1)

2.2.2 **client** The client needs to initialize a Socket and then connect to the server. If the connection is successful, then the connection between the client and the server is established, and then the two parties can transfer data to each other. The specific process is as follows:

1. Open a communication channel and connect to a specific port of the host where the server is located;
2. Send a service request message to the server, wait for and receive the response; continue to make the request.
3. After the connection is successful, the information is exchanged with the server.
4. After the request ends, the communication channel is closed and terminated.

The process can be represented by Figure 2:

![Figure 2 Socket communication client process](image2)

2.2.3 **Overall structure realization process** We can use a lot of functions to create a Socket communication system. The specific implementation process is shown in Figure 3:
2.3 c/s structure application Considering the characteristics of the C/S structure, when many machines work together, the C/S structure is suitable for storing data on the server side for each client to access and use. The interconnection between the client and the server and the exchange of information can be realized through the underlying Socket communication. Enable each client to understand each other's information.

3. Implementation of Communication System Based on P2P Protocol

3.1 JXTA protocol
The traditional C/S architecture is suitable for situations where each node needs to request resources from the same server. However, when each node needs to transmit information to each other, the P2P architecture is more suitable for mutual communication between nodes. The Peer to Peer (P2P) network is referred to as the third-generation network, which uses a point-to-point decentralized network shelf. It does not require server transit, making the connection between the user and the user more convenient and direct. The P2P architecture communication network can be implemented more conveniently using the JXTA protocol and the API functions provided by it.

The structure of the JXTA system studied in this paper is shown in Figure 4:
JXTA is an open p2p network platform that builds an underlying application architecture for the operation and development of p2p applications. The good independence and scalability of the JXTA architecture also enables p2p technology to be better applied to a variety of portable digital devices. JXTA can make p2p communication easier to implement, and it can make each node more convenient to realize information transmission.

### 3.2 Implement p2p structure communication based on JXTA protocol

This paper studies the P2P communication network based on the JXTA protocol and Wi-Fi as the communication medium. The system needs to be initialized first, and then functions such as communication and file sharing can be realized. After opening the application, the system first checks if it is connected to the Wi-Fi network. If it is not connected, it will remind the user to access the network. After connecting to Wi-Fi, the JXTA network is added. The system searches for the network rendezvous point RDV and reads the peer list above. Communication services are implemented with bundled pipe services. When sharing files, the user node needs to request the RDV for a list of resources of nearby nodes. RDV is similar to a server and is a more powerful node in the JXTA network. After the RDV receives the request information, it feeds back the resource list in XML format. At this time, the user node sends a file transfer request to the relevant node according to the list information, and the file can be transmitted after the handshake.

Although JXTA is platform- and protocol-independent, the JXTA community has introduced
Java-implemented APIs. Since Java is also the development language of Android, you can call the Java API directly, which makes it easier to implement the system. The specific steps are as follows:

(1) Initialization section: After the application is launched, the system checks the current network configuration. First create a network connection management object ConnectivityManager, determine whether there is a network connection, use get Active Network Info() function to obtain network information. Then use the Network Info class to determine if the network is available. If the Wi-Fi network is not available, the user is prompted to reconfigure the network. If the user is connected to the Wi-Fi network, the system will complete the initialization operations such as checking the cache status, starting the JXTA network connection, searching the JXTA rendezvous point RDV, and reading the peer list on the rendezvous point.

(2) Protocol (PeerDiscovery Protocol) implementation. Advertisement is the basic unit of information exchange between peers. It is the meta-document used by JXTA protocol to describe resources. JXTA Advertisement is usually represented by XML document. The problem of discovering other Peer and its resources is converted into the problem of discovering the advertisement describing each resource. As long as the corresponding Advertisement is found, it is equivalent to finding the resource. The PDP defines a protocol for discovering other Peers and resources. The protocol includes two aspects, one for requesting access to other Peer's Advertisements and the other for responding to other Peer requests. After each node has obtained the information of the remaining nodes, it can select the correct Peer node to connect.

(3) Information communication: The connection of information communication is established directly from end to end, so the communication module naturally has the nature of P2P. The communication module uses JXTA bidirectional pipes for message transmission. After the user logs in, start a thread, create a JxtaServerPipe in the thread, keep polling, and wait for messages from other users. When user A wants to chat with user B, it is only necessary to establish a JxtaBiDiPipe to user B, and the JxtaBiDiPipe and user B’s JxtaServerPipe are connected.

JxtaBiDiPipe and JxtaServerPipe are very similar to Socket and ServerSocket in Java JDK. The implementation steps are similar to those in Socket communication. The JxtaServerPipe class is used to listen for port connections. JxtaBiDiPipe is used to generate pipes for communication.

After the two users establish a connection, the message encapsulated in the xml format is transmitted to the other end through the JXTA pipeline. After parsing, the message is displayed on the screen. The message contains different information such as the version of the message, the name of the sender of the message, and the like. The receiving end parses the received message and displays it in the chat dialog interface.

(4) File Transfer: The file transfer implementation is a program written in P2PSocket that can transfer files peer-to-peer between any peers. Files are streamed and can support ASCII files and binary files. The system simplifies the search based on file content. Therefore, the file resource is only broadcasted in the JXTA network. If the response is received, the resource request confirmation is performed. After the confirmation is completed, the current file resource is sent by calling the this.socketService. fileTransfer() function. PipeMsgListener is implemented by PipeMsgEvent, and events are caught each time a message is sent to the pipeline. fileRequest is used to process requests for shared file lists.

The file transfer process can be represented by Figure 5:
4. Conclusion
The C/S structure is mature in technology. Its main features are strong interactivity, secure access mode, fast response, and good processing of large amounts of data. It can be implemented with the most basic Socket communication. However, in the C/S architecture, the client needs to rely on the server, which is not conducive to more clients working at the same time. The nodes in the P2P structure can implement direct communication and improve the utilization of resources. This structure can be implemented on the basis of the JXTA protocol. However, the P2P system is relatively open and has high requirements for network management. Therefore, in a system that requires a large number of machines to work at the same time, the C/S protocol is used to implement server management and information distribution for each machine, and then the P2P protocol is used to implement point-to-point communication between the machines, and the simultaneous application of the two protocols can be performed. Improve resource utilization and make the system work more efficiently.

References
[1] James F. Kurose, Keith W.Ross (2018) COMPUTER NETWORKING A Top-Down Approach (Fifth Edition)[M] China Machine press
[2] Rolou Lyn R. Maata,Ronald Cordova,Balaji Sudramurthy and Alrence Halibas Design and Implementation of Client-Server Based Application Using Socket Programming in a Distributed Computing Environment[C] 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),2017
[3] Wang Ruiqing Research on Client/Server Technology[J] Computer Development & Applications, vol.25, no.6, sept.2012: 23-25
[4] Li Gong, Scott Oaks, Bernard Traversat. JXTA IN A NUTSHELL[M]. 0’Reilly&Associates, Inc. 2002.
[5] Joseph D. Gradecki. Mastering JXTA: Building Java Peer-to-peer Applications[M]. John Wiley&Sons.Inc.2002.
[6] JXTA JavaTM Standard Edition v2.5: Programers Guide[M].Sun Microsystems, Inc.2007.
[7] Liu Hongqing; Liu YanFile Transmission and Download System Based on P2P[C] Proceedings of 2018 5th International Conference on Electrical & Electronics Engineering and Computer Science(ICEEECS 2018)

[8] Wang Xiaohua, WangPeng, MuAiqian Study of P2P Computing Based on JXTA Platform[C] Control Engineering and Communication Technology (ICCECT), 2012 International Conference on

[9] A. Berl?H. de Meer,and T. Hossfeld. Peer-to-peer solutions for cellular networks [J]. First International Conference on Advances in P2P Systems, 2009:54-59.

[10] W. Song and W. Zhuang. Multi-service load sharing for resource management in the cellular/WLAN integrated network[C]. IEEE Trans. Wireles Comm., vol. 8, no. 2, feb. 2009:725-735.