A Distributed Instant Messaging System Model

Hao Zhang, Zhongkui Sun, He Li

Qinggong College, North China University of Science and Technology, Tangshan 063000, Hebei, China

Jianmin Li

Institutes of Electronics, Chinese Academy of Sciences, Beijing 100000, China

Abstract

Most existing instant messaging system is built on "centralized server" mode, which is designed by using C/S or P2P framework to require client must log on a remote server, or on a centralized server farms. There is a problem about this instant messaging system: If the LAN firewall or proxy server restrictions instant messaging services, or when outside the LAN connection is disconnected, the user cannot even in the same local area network to communicate with each other. To solve this problem, design a multi-server model of distributed instant messaging system by using the C/S and P2P hybrid structure. When the remote server with limited or disconnected from a network failure, the client can log on to the server on the LAN, so you can ensure the same in a LAN client continue instant messaging. This model not only can the user's important information for centralized and easy to manage, but also to improve the quality and efficiency of communication between the customers; also enables users on the same LAN without passing through the external network can communicate with each other needs.

Key words: INSTANT MESSAGING; CLIENT/SERVER; PEER TO PEER

1. Introduction

IM (Instant Messaging) is a LAN-based or real-time interactive Internet applications, instant rapid development of communications are dramatically changing the way people communicate, collaborate and play [1]. More simply, functional instant messaging system is the user real-time communication. Instant messaging system to organize the exchange of large virtual communities, where users can find like-minded, have a common language, willing to share the Friends (Friends), for on-line (Online) friends can communicate in real-time text, voice, video, documents, etc.. For example, we are most familiar Tencent QQ in China, QQ users to communicate with the general pattern of use is as follows: First, the user must register a QQ account, communicate with friends, log on QQ client, and connect to the QQ server. Next, look at your friends are online; if friends are online, you and your friends to send or receive messages to each other; sometimes to find online users, to choose one request add to friends, instant communications; friends Offline can to a friend mail; a friend can transfer files to each other.

The used mode Tencent term used to describe (in order): on-line; Friends List; messaging; add friends; online; send the file. Communication model for other instant messaging systems and Tencent QQ is similar.

In RFC2778, the instant messaging system is defined as: A presence and messaging system allows users to subscribe to each other and be notified of changes between each user in state, and for users to send each other short instant messages [2, 3]. Popular speaking, instant messaging refers to the communication both sides on the Internet, very quickly sends and receive messages to each other's activities. Receivers send and receive messages and information publisher information delay is very small, almost at the same time meet the requirements of transfer. Online communication tools such as we often use: QQ, MSN, Fetion mobile communications systems tools [4, 5]. In July 1996, four young Israelis set up a company Mirabilis, ICQ and released the initial version of the same year in November, as the world's first instant messaging software, quickly swept the world, then instant messaging software abound. Currently the best-known instant messaging system ICQ, MSN Messenger, Google talk and Yahoo Messenger, the most popular Tencent QQ, Netease bubble and flying letters. The instant messaging system provides real-time communication on the basis of the basic functions, which has its own characteristics. Flying letters can send free text messages to friends' mobile phones; Google talk chats in the form of e-mail stored in Google mail mailboxes inside.

It said instant messaging system based on the "hub server" mode, using C/S or P2P mode design requires the client application must be logged in to the remote server, or on a centralized server farms. If C/S model, this model is divided into formal logic, business logic, data logic and data storage from the structure, all the forms and business logic reside on the client, and data storage logic and data reside on the server side. The advantage is that the user mode some important information can be centralized and easy to manage, but with the number of users increases, the system is running and when communication between customers, increase the burden on the server, reducing the quality and efficiency of communication among customers [6, 7]. In P2P communication mode, each node also has information on most consumers' information providers and information and communication aspects of the function. The rights and obligations of P2P networks have each node are peers [8]. This approach has powerful processing capability and save bandwidth performance, P2P technology weaken the role of the server, or even abolish the server, any two computers to each other servers, but it is also a client. The disadvantage is inconvenient to manage.

Therefore, instant messaging system research worldwide will accelerate the development of the emergence of a trend, but both centralized and easy to manage important information about the user, and can effectively reduce server load and improve the quality and efficiency of customer communications among instant messaging system model is the direction of current research.

2. IM systems and communication process model

2.1 Centralized server mode and communication process

Now the communication mode most instant messaging system is primarily built on "centralized server." Between the user of this mode of communication, must be first log on to a remote server or a centralized server farms, establish a connection, communication is instantaneous. Its network model shown in Figure 1:

Assuming a single-server IM is shown in figure 1 to establish a buddy relationship between the user A, B, C, and D. In a single-server mode communication process between users as follows: Initially assumed that users are offline, the user A logged into the local server, and informs its IM server online, the local server record online user A's. When between user A and user C under the same server need to communicate, IM server advertised, IM buddy list server detected, notify the user A, user C's current online status. In this case, if the user is offline C, IM, server feedback to the user A, user C current offline, therefore, when the user A can only be in a wait state. If the user C already is online, then IM server notifies the user A and user C both in-line and can communicate.

When a user logs on the server, his IP address and port number of the report prepared for the future communication to the IM server. When the new user logs on, the servers is responsible for the other user status notification new login user, and notify all their online friends of IP addresses and communication ports. Therefore, by each other's IP addresses and communication ports to establish P2P connection between friends, then you can point to point between the two instant messaging. In the communication process, it is assumed between user A and user C can properly establish TCP / IP connections, instant messages can be transmitted through ordinary Sockets interconnected between them. But if there is a firewall blocking or other reasons cannot establish normal TCP / IP connection in both the

communication process, the instant messages between the two still need to be forwarded through the server.

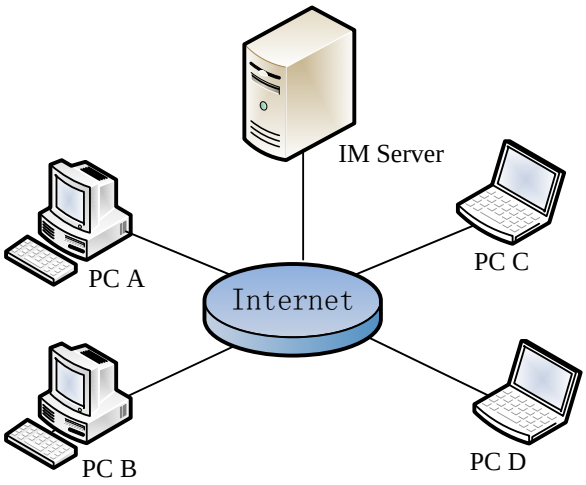

Figure 1. Model of centralized server IM

The advantage of this mode of communication is beneficial to the system server maintenance and upgrades. But its shortcomings are obvious: a first, the network speed is not fast enough and widely distributed users, this model increased the burden on the server, it is difficult to provide high quality service for all users. Second, the "central" server system on the server hardware made very high demand, the cost of purchase and construction of such servers tend to be very expensive, and maintenance cost is also expensive.

In order to ease the pressure on the server, there are now "focus" IM server farms. This uses multiple servers to complete a single server functions, and batch data for each user on multiple servers. This IM server farm to some extent reduces the server load independent. But the "centralized" type server model prevalent a problem: If the LAN firewall or proxy server restrictions instant messaging services, or when outside the LAN connection is broken, even within the same local area network users cannot communicate with each other.

2.2 Distributed multi-server instant messaging system mode and communication process

In order to solve the "central" server mode problems in a multi-server distributed instant communication model. The system model with three and four client terminal server, for example, the system model shown in Figure 2:

In this model system, the local server is Ⅰ, users within the LAN A, B, etc. to provide services within the local server is Ⅱ, LAN users C, D, etc. to provide services, the server Ⅲ remote server, a multi-a LAN client terminal to provide services.

Assume that the user A has been in the LAN and LAN Ⅰ Ⅱ C was established within the user buddy relationship. In this communication mode in the LAN Ⅰ, if the user A and user B communication, the communication process is similar to the "central" server mode, no further explanation. If users in the LAN Ⅰ A to user C Ⅱ communication within the LAN and, following the communication process between them(Figure 1).

Initially assumed that all users are also offline, when user A logs on the local server Ⅰ, notify the local server Ⅰ their online status. Ⅰ note of the local server on-line status of the user A, and then check the buddy list of the user A, user C is not found in the user's local server Ⅰ. A local server user exits Ⅰ, re-log into the remote server, and report on their IP address and prepare for future communications port number. Write down the remote server online user A and user C's responsibility to inform the local server Ⅱ, asking them to inform the user C, user A's on-line status. If user C is not online, due to the local server solely responsible for recording the local user's online status, so that the local server Ⅱ A discard message online status of users.

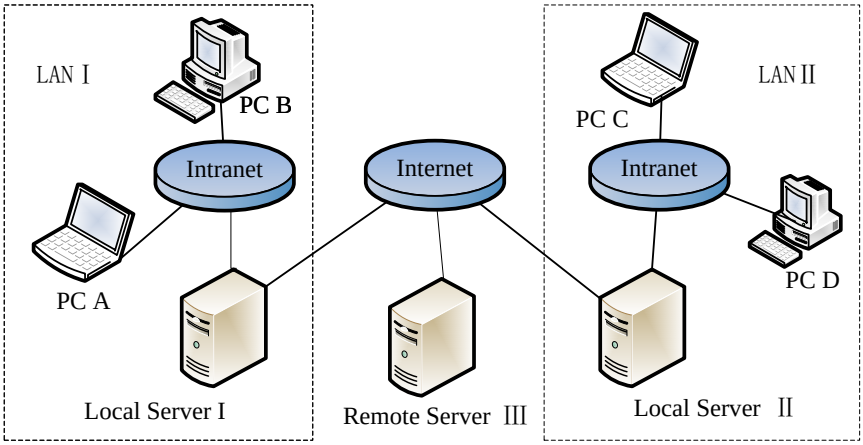

Figure 2. Model of distributed multi-server IM

When the user C on the line, it will notify the local server Ⅱ their online status. Local server Ⅱ check their friends list, find the user A is not a local server users, C quit the Local login, login remote server. Similarly, the remote server to record the user A's online status, user C report their IP address as well as preparing the future for instant communication for listening TCP / UDP port number. As has been informed of the other's IP address and TCP / UDP port numbers between user A-line, so that users A and C, will be able to direct the instant communication.

When the user C on the line, it will notify the local server Ⅱ their online status. Local server Ⅱ check their friends list, find the user A is not a local server users, C quit the Local login, login remote server. Similarly, the remote server to record the user A's online status, user C report their IP address as well as preparing the future for instant communication for listening TCP / UDP port number. As has been informed of the other's IP address and TCP / UDP port numbers between user A-line, so that users A and C, will be able to direct the instant communication.

3. Design on IM system model

3.1 IM system architecture model

Due to the distributed multi-server IM systems, each server access to the Internet in accordance with their own way and with their clients can establish an independent instant messaging system; therefore, the topology of the entire system is the network structure. Which can be established between any two nodes of the instant messaging server, eliminating the forwarding process information, thereby improving the efficiency of instant communication, reducing the burden on the server; will not result in a server hang caused by impaired users communicate. System structure shown in Figure 3:

When the users in a local area network need for communication, they will send a request to the local IM server. In some cases, the local server gives a response. After the user and the local server connection are successful, the local server creates a new thread for the user. Data processing between the user and the server is done by a thread. IM client and server-side data are to be encapsulated by calling the information processor. If the user needs to communicate between the LAN, by the local server in the corresponding thread request to the remote server. Given the appropriate remote server to establish a connection with the remote server, and ultimately the communication between the user LAN.

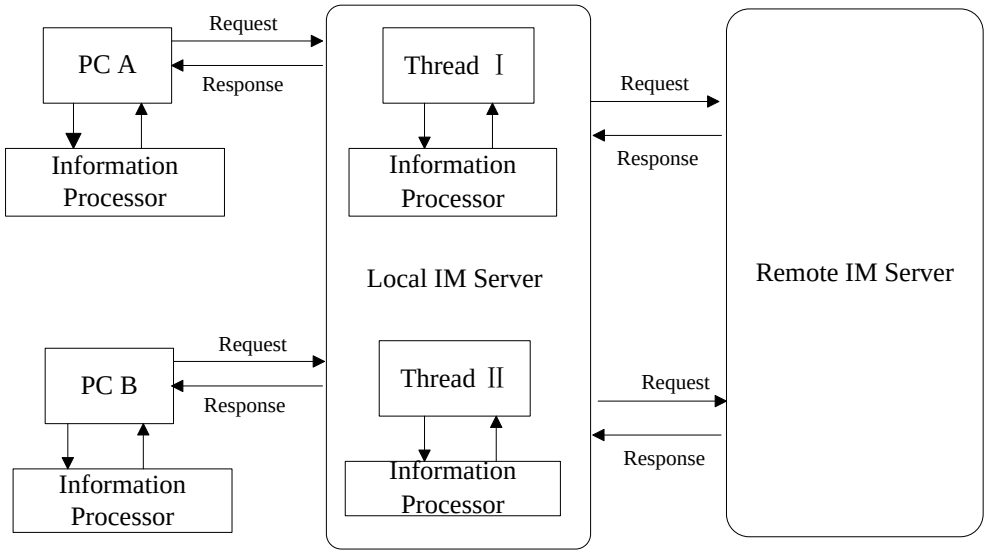

Figure 3. The system structure chart

3.2 Rules of the system user ID

In the current instant messaging system user is mainly in two ways. In ICQ, QQ and other systems such as the number of employed is way, by assigning to each user a unique digital serial number of ways a user ID. In AOL and Yahoo through instant messaging systems like you take your own way registered user identification name. Both methods have their advantages and disadvantages: the digital way is not easy to remember, but network name duplication problems that may arise between different user registration solved; logins ways much freedom to the user, but the same name problems cannot be solved, only to

Tom1, Tom2 difference in such a manner.

In this design, IM system, the use of multi-server architecture, the number of users on each server is reduced, and therefore, it is recommended to take away the user login name for user identification. Meanwhile, in order to identify the client from which server, the system user identification reference Jabber system, similar to the Email address form to be realized. For example, user Tom has an account on the server Serverl.com: Tom, then the user ID can use "Tom @ Serverl.com" to represent.

3.3 Instant Message Format of System

Between the client and the server, data transfer between clients and servers between XML structure in accordance with certain. <present /> element is used to determine the user's status, <message /> element is used to transmit messages between the two clients, <iq /> element is completed between the client and the server XML queries and responses.

Such as instant messaging data sent to Tom @ Serverl.com Jerry@Server2.com as follows:

<message
to = 'Jerry@Scrver2.com '
from = 'Tom @ Serverl.com ' type = ' chat '>
< body＞Where are you , Jerry?＜/body＞
</message>

In the XML data stream, the identification information purposes, source, type, content and so on. Server1 so you can transfer issue has logged on to the customer Server2 under Jerry, Jerry know the source of the information upon receipt of reply to an instant message to be received based on XML data.

3.4 Web services model IM systems

Since the data transmission between the LAN instant messaging systems may need a firewall, but in the case of Internet connection is normal, there may not be to transfer data between networks. Web services for data transmission via HTTP protocol. Even with the presence of a firewall, HTTP protocol can be properly adopted. Therefore, the data transmission between servers in two ways, first in the form of direct transfer of data packets, the other one is, when not directly transmit data, send data to each other through a local server tries to call a remote Web server. That is when two different instant messaging network client ends cannot communicate directly; it can be forwarded via its Web services, data transmission.

Implementation calls the Web service to pass data between the server programs, there are two: one is the IM server that contains the Web service. Such programs need to implement server at the same time, we must implement Web services. Another is to set up separate IM server and Web services, which is conducive to system design division and different versions in the open unity of the agreement, installation and implementation. Figure 4 is a graph:

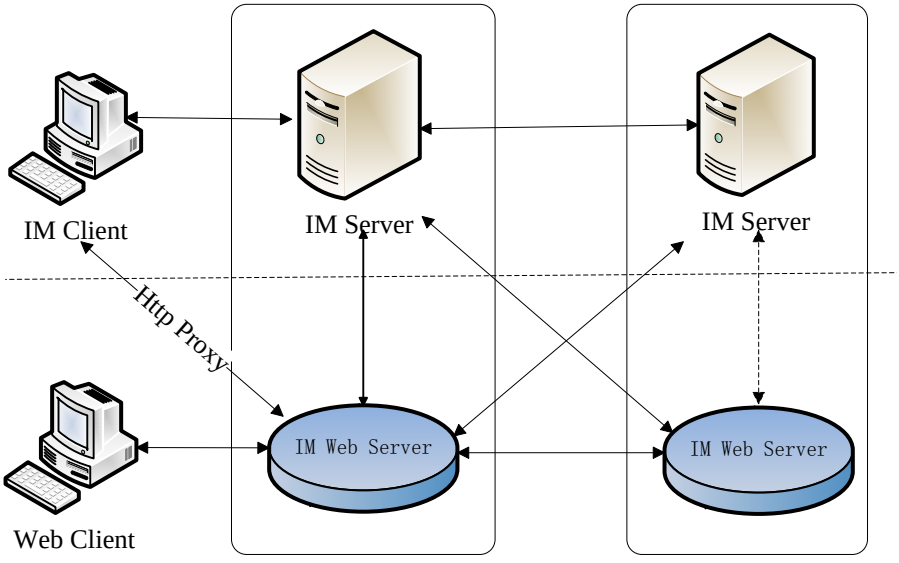

Figure 4. Web service model of IM

Part of the figure when the case of instant messaging system Web service is not open when the dotted line above; part of Web Services case opened under the dotted line. Web Services IM client can call communicate via HTTP proxy. In addition Web service can also call the Web browser client to call via the web access, you can communicate between clients. Server system provides the following Web services:

Login/Logout Service: It is responsible for providing user login authentication;

Register Service: It is responsible for providing

user login authentication;

FriendList Service : After the user logs in charge of generating the user's buddy list;

AddFriend Service : For the user to specify the user to the buddy list;

DelFriend Service : Provide a user deletes their buddy list specified item;

SendMessage Service : Implementing user sends a message to a specified friend;

Logout Service: Complete user logs off.

These Web services, users can instantly communicate through firewalls and other IM users.

4. UML system modeling

4.1 UML class diagram model establishment

System main object class to "chat interface class," "Friends informational received", "send a message to a friend category," "send offline messages to the server category," "Send to a Friend receiving server IP address and TCP/UDP communication port class "," server class. "As the "Friends of information received class" and "Receive Server Info" is to receive information classes, and receive information with a class common, allowing them to receive information class inherit the super class of common, similar due to the "send a message to a friend category "and" send a message to the server class "is a message class and common class having transmission information, you can also send a message classes inherit the super class of common, class diagram shown in Figure 5:

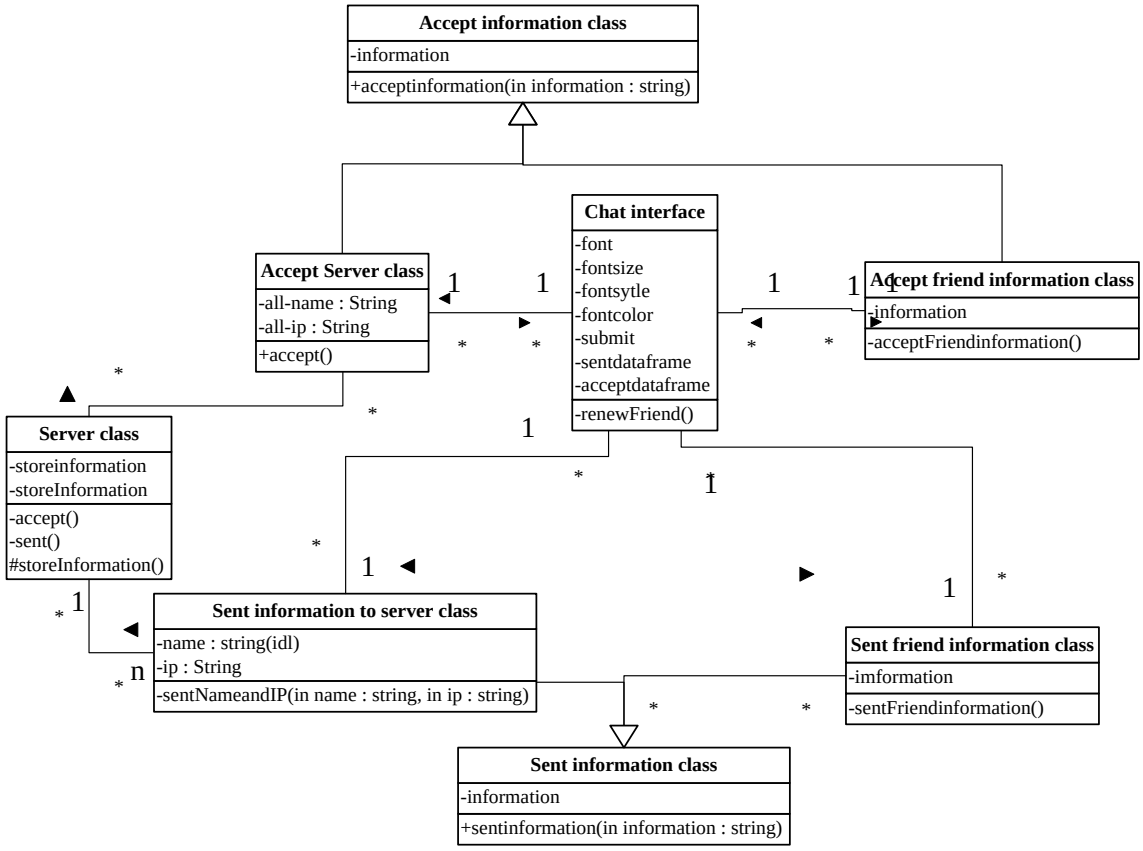

Figure 5. Class diagram

4.2 UML to build dynamic model

Purpose is to write a script that lists the user and system interaction, writing scripts.

Normally script:

1) A user opens a login screen, set the server network parameters, the connection to the server was successful.

2) User open a communication interface, the user invokes the "send a message to the server class." "Send a message to the server class" will own IP address and communication port to send to the server, and the server waiting to receive the information from.

3) When the server detects when the user requires a connection, the user's friends to send the user information.

4) The user knows the IP address of a friend and a communication port can communicate with your friends. Call the "send a message to a friend category," will send out the information to be transmitted, and friends waiting to receive the reply message.

Exceptions script:

1) A user opens a login screen, set the

server network parameters, and server connection errors.

2) "Send a message to the server class" does not own IP address and communications port to send to the server, waiting for information from the connected server.

3) The server is not listening to user requirements for connection, not the user's friends to send information to the user.

4) The user knows the IP address of a friend and a communication port can communicate with your friends. But call the "send a message to a friend category," information to be transmitted cannot be sent out; we did not receive a reply message.

Based on the above analysis of the script, draw the following diagram describes the sequence of events for each interaction system.

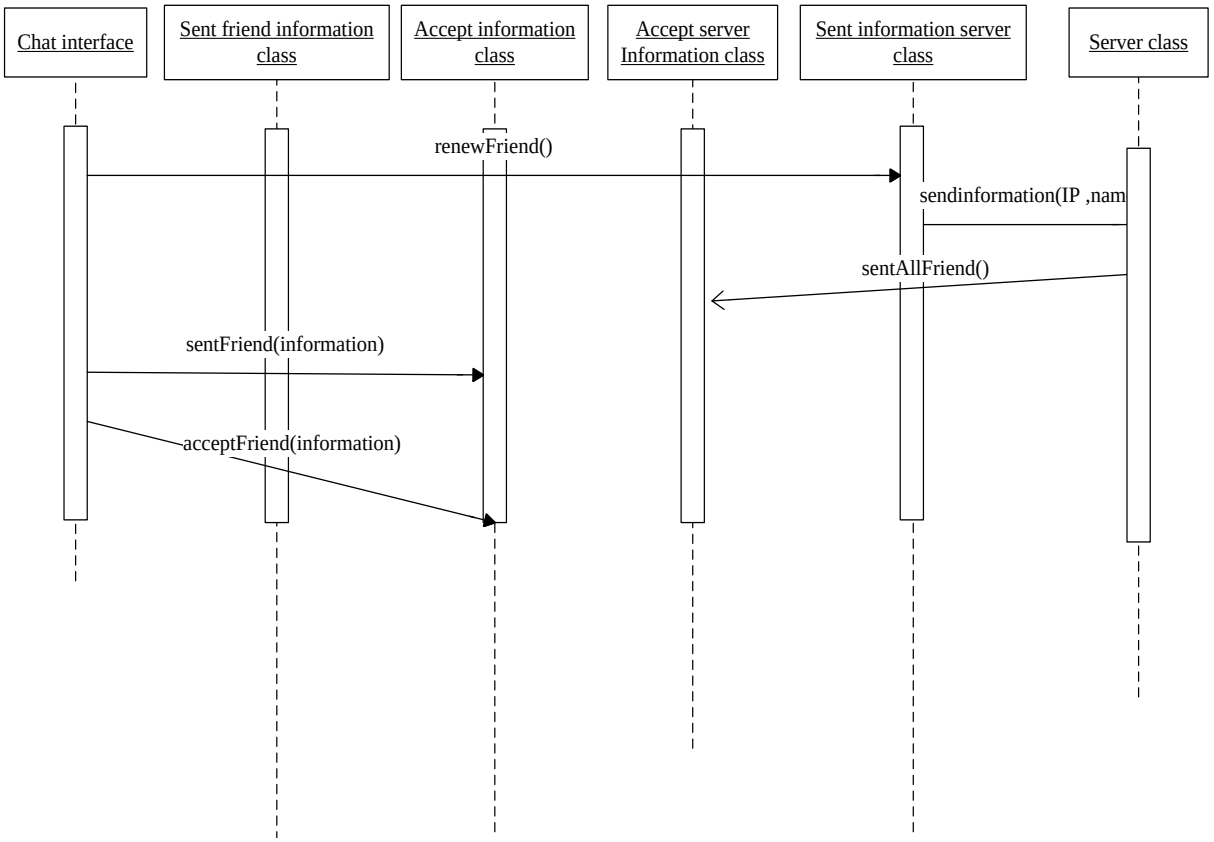

Figure 6. Interaction events of all system

5. Conclusions

This paper has deeply studied the model of a centralized server communication system problem in a distributed multi-service instant messaging system model. Use of C/S and P2P mixed mode, a distributed instant messaging system. System Login can select the server when the server is disconnected, you can select the server to log in again. To achieve the LAN behind a firewall or proxy server restrictions instant messaging services, or when outside the LAN connection is disconnected, even if a user can also communicate with each other on the same LAN.

Distributed multi-service instant messaging system model is proposed in this paper that given the specific design of the model. This is including: application structure model, instant messaging format system user identification rules and systems, etc. Since the data transmission between the LAN instant messaging systems may need the firewall, the paper presents the system sends data to the other local server by calling a remote Web server, data transfer functions. This paper designs a distributed multi-server instant messaging system, which ensures the integrity of user information to users on multiple servers' registration information to be consistent.

Acknowledgements

This work was supported by the Science and Technology Department in Tangshan in China. (13110203b).

References

1. Xie M.J. (2012) Secure instant messaging in enterprise-like networks, *Computer Networks*, 56(1), p.p.448-461.
2. Jorrit S. (2010) A Massively Scalable

Architecture for Instant Messaging & Presence. *Electronic Notes in Theoretical Computer Science*, 261(1), p.p.109-130.
3. Donal C. (2007) Building a secure instant messaging environment. *Network Security*, 2007 (1), p.p.18-20.
4. Zhang H. (2013) A Method for Detecting and Blocking Instant Messaging Software. *TELKOMNIKA*, 11(10), p.p. 5684-5693.
5. Zhang H. (2010) Research on detection of instant messaging software. *International Symposium on Information and Automation*. 86(1), p.p.664-669..
6. Zhang W.L. (2009) Design and Implementation of P2P Enterprise Instant Messaging System. *Applied Mechanics and Materials*, 687-691, (4), p.p. 2740-2743.
7. Zhou X. (2010) Understanding the Nature of Social Mobile Instant Messaging in Cellular Networks. *Communications Letters*, 18 (3), p.p.389-392.
8. Elena V. (2012) A distributed system for learning programming on-line. *Computers & Education*, 58(1), p.p. 1-10.