Report of An Online Conferencing System Implementation

RUNZHI JIANG 11810112, Southern University of Science and Technology, China
JIYUAN JIA 11810506, Southern University of Science and Technology, China
SHUWEN DONG 11912306, Southern University of Science and Technology, China

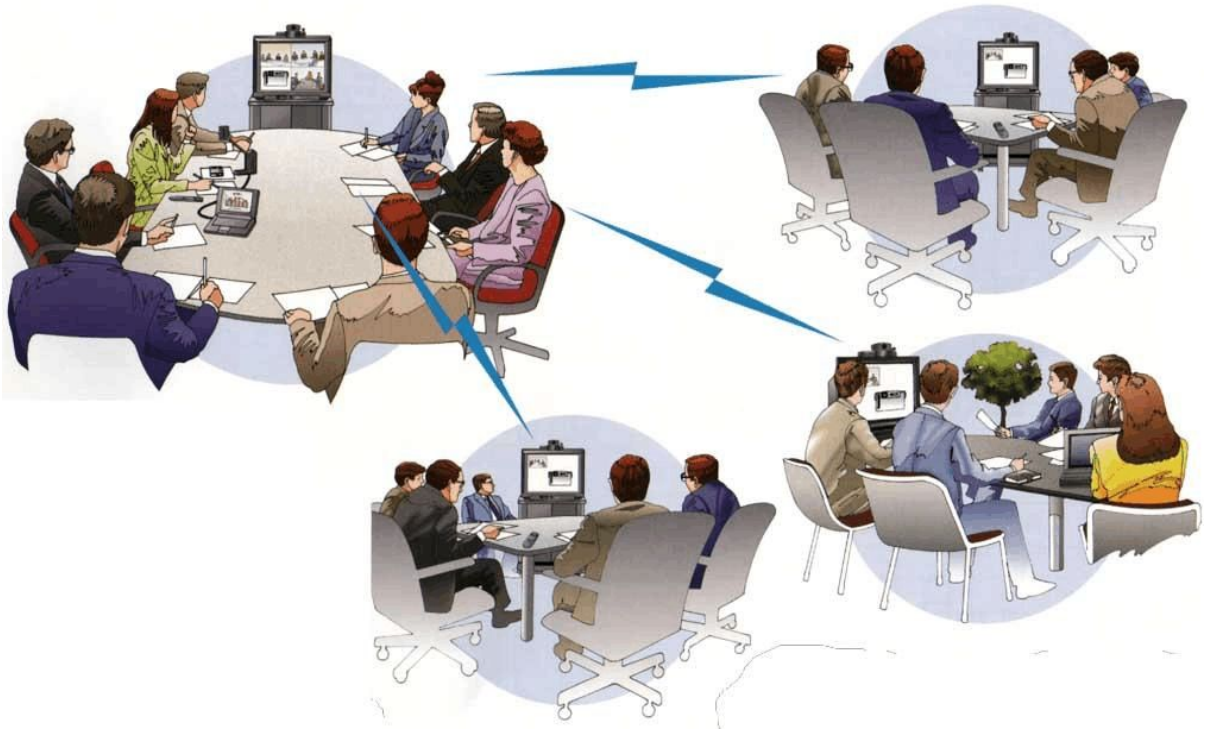

Fig. 1. A video conference in progress.
(The Figure is Retrived from Internet)

This paper implements a client-server model online conference system, which can realize numerous functions such as audio transmission, video transmission, screen sharing and remote control. A sser-friendly interface GUI is designed and data encryption is also implement by AES(ECB) and RSA. Background and related network protocol survey are written at the baginning and summary and future work are written at the end.

Additional Key Words and Phrases: online, conference, remote, control

Authors' addresses: Runzhi Jiang 11810112, Southern University of Science and Technology, Shenzhen, Guangdong, China, 11810112@mail.sustech.edu.cn; Jiyuan Jia 11810506, Southern University of Science and Technology, Shenzhen, Guangdong, China, 11810506@mail.sustech.edu.cn; Shuwen Dong 11912306, Southern University of Science and Technology, Shenzhen, Guangdong, China, 11912306@mail.sustech.edu.cn.

1 BACKGROUND AND RELATED NETWORK PROTOCOL SURVEY

1.1 Background

With the development of technology and society, online conferencing is playing an increasingly important role in people's life. For work and life, people often need to have meetings in different places. Online conferencing allows people to communicate and share their work from far away. Therefore, online meeting greatly facilitates people's life and changes the lifestyle of modern people.

Because of its importance and significants, our group is very interested in online meetings. Being able to implement an online meeting system will be extremely rewarding. Therefore, we decided to choose An Online Conferencing System Implementation as our project. At the same time, it will greatly improve our practical ability and computer network knowledge.

In online conferencing, participants' audio, video, and computer screens should be shared in real time with other participants. It is worth mentioning that remote control will also be implemented. That is really shocking and surprising.

1.2 Related Network Protocol Survey

In a computer network, two entities communicating with each other need to exchange information to coordinate their actions and achieve synchronization, and the exchange of information must be in accordance with the agreed process in advance, which is what we call network transmission protocol, or referred to as transmission protocol. The video conference system needs to be connected with different terminals, so the video conference terminals need to follow a unified protocol.

It is normal to say that many network protocols are required to implement online conferencing. The protocols of online conference can be divided into several categories, the first is the framework protocol, the second is the transmission control protocol, the third is the codec protocol, the fourth is the message transmission protocol.

1.2.1 Network Session Mechanism.

A session is a persistent network protocol that creates an association between the user (or user agent) side and the server side to exchange packets. Session mechanism is a server-side mechanism in which the server makes some kind of data structure (probably a hash table) hold information. Session mechanism is described as follows:

First, the client send a request to the server.

Then, the server checks whether the request contains a session ID and takes different actions depending on the situation.

If the request contains the session ID, it means that the server has saved session information for the client. Then it will retrieve the corresponding session information based on the session ID. If it fails to retrieve the session information (the session information is deleted due to timeout), it will create a file or a data structure variable used to store the session information and generates the session ID associated with the file or data structure variable.

Otherwise, the request does not contain the session ID. Correspondingly, the server will create a file or some data structure variable to hold session information, and generate the session ID associated with the file or data structure variable. Next, the server sends the session ID to the client as a response message. If the client request does not contain the session ID, the server also instructs the client to save the session ID.

1.2.2 Framework Protocol.

The framework protocol of software video conference is the basic framework of the whole video conference system. Now the mainstream framework protocols of video conference are SIP, H323 and MGCP.

Here are the two session protocols we learned about: SIP and H.264.

1.2.3 SIP.

SIP(Session Initiation Protocol) is an application-level control protocol that can be used to establish, modify, and terminate multimedia sessions (or meetings) such as Internet calls. SIP can also invite participants to an existing session, such as a multi-party conference. Media can be easily added (or removed) to an existing session. SIP supports the following five aspects in establishing and maintaining multimedia session protocols:

- **User Location:** Checking the location of end users for communication
- **User Validity:** Checking the willingness of users to participate in sessions
- **User Capability:** Checking media and media parameters
- **Session Establishment:** Setting up session parameters at caller and called party.
- **Session Management:** Including sending and terminating session, modifying session parameters, activating service and so on.

SIP is not a vertically integrated communication system. SIP is more appropriately called a component that can be used as part of other IETF protocols to construct complete multimedia architectures, that, for example, will include the Real-time Transport Protocol (RTP)(RFC 1889) for transmitting real-time data and providing QoS feedback, real-time streaming protocol (RSTP)(RFC 2326) for controlling the transmission of streaming media, Media Gateway Control Protocol (MEGACO)(RFC) 3015) for controlling gateways to the public Switched Telephone Network (PSTN), and the Session Description Protocol (SDP)(RFC 2327) for describing multimedia sessions.

Security is particularly important for the services provided to achieve the desired level of security. SIP provides a suite of security services, including denial of service prevention, authentication services (user-to-user, proxy-to-user), integrity assurance, encryption, and privacy services.

The following figure shows the basic process of SIP - based session invitation.

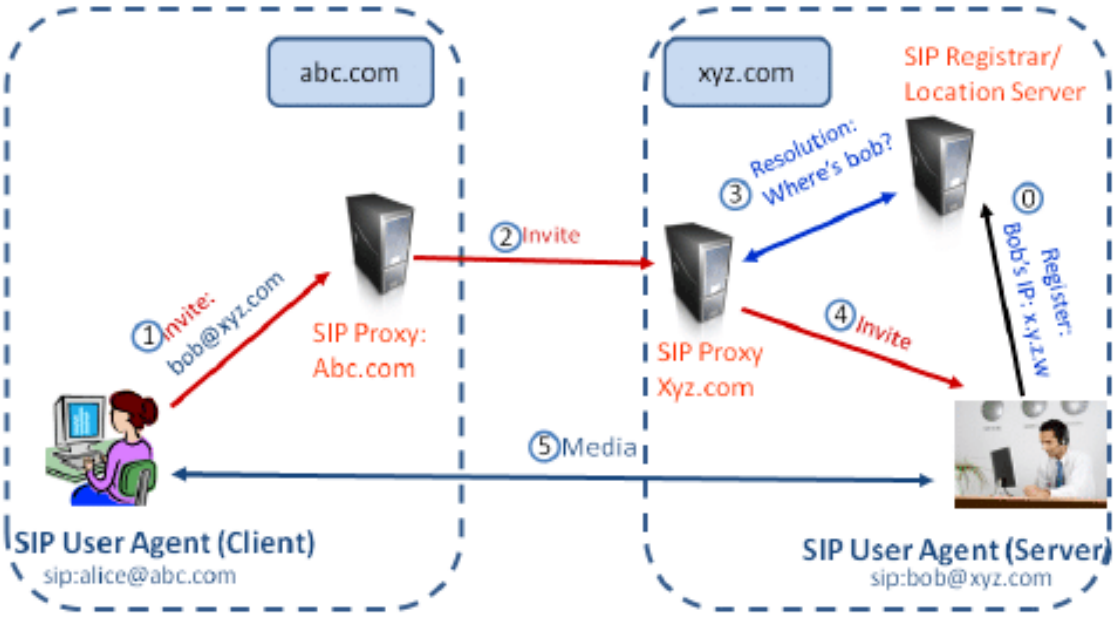

Fig. 2. SIP Protocol: Call setup when alice@abc.com invites bob@xyz.com
(The Figure is Retrived from Internet)

1.2.4 H.264.

H.264, also known as MPEG-4 Part 10, Advanced Video Coding, is a block-oriented video coding standard based on motion compensation. The main goal of the H.264 standard is to provide better image quality at the same bandwidth as other existing video coding standards. It introduces IP packet - oriented coding mechanism, which is beneficial to packet transmission and supports video streaming media transmission in network. H.264 has strong bit error resistance and can adapt to video transmission in wireless channels with high packet loss rate and serious interference. H.264 supports hierarchical coding across different network resources to achieve smooth image quality.

By 2014, H.264 had become one of the most commonly used formats for high-precision video recording, compression, and distribution.

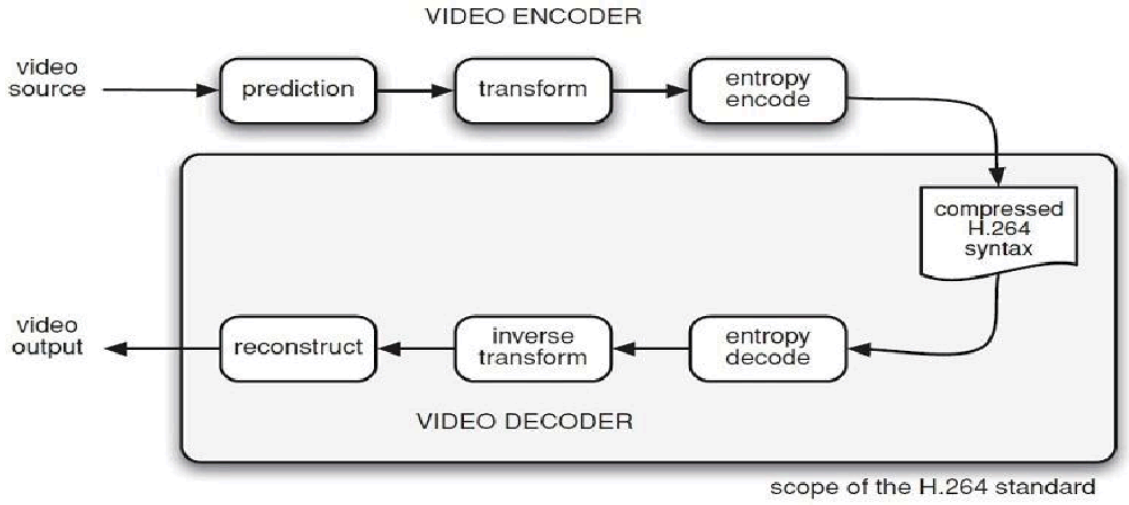

Fig. 3. The H.264 video coding and decoding process
(The Figure is Retrived from Internet)

1.2.5 Transmission Control Protocol.

In the field of online conferencing transmission, control protocols are mainly TCP, UDP, RUDP and RTP protocols. TCP and UDP are transport layer control protocols. TCP is a control protocol with connections, but it is slow to transmit data in real time because of the three-way handshake. UDP is built on a connectionless channel, so the reliability of data transmission cannot be guaranteed. In the online conference system, RUDP protocol is generally used, which is the reliable UDP transmission protocol. It is added with the control protocol on the basis of UDP protocol, so as to ensure the real-time performance of data and the reliability of data.

1.2.6 Real-time Transport Protocol (RTP).

Real-time Transport Protocol (RTP) is a network Transport Protocol. It was published by the Multimedia Transport Working Group of IETF in RFC 1889 in 1996. The RTP protocol specifies the standard packet format for delivering audio and video over the Internet. It was originally designed as a multicast protocol, but has since been used in many unicast applications. An RTP packet consists of two parts: the header and the payload.

RTP Session Process. When an application establishes an RTP session, the application determines a pair of destination transport addresses. The destination transport address consists of a network address and a pair of

ports, one for RTP packets and one for RTCP packets, so that RTP/RTCP data can be sent correctly. The RTP data is sent to the even-numbered UDP ports, and the corresponding control signal RTCP data is sent to the odd-numbered UDP ports (the even-numbered UDP ports +1), thus forming a UDP port pair. The RTP sending process is as follows, and the receiving process is the opposite.

- **1)** The RTP protocol receives the streaming information code flow from the upper level (such as h.263) and encapsulates it into RTP packets.
- **2)** RTCP receives control information from the upper level and encapsulates the RTCP control package.
- **3)** RTP sends the RTP packet to the udp port for the even number port; RTCP sends the RTCP control package to the receiving port in the udp port.

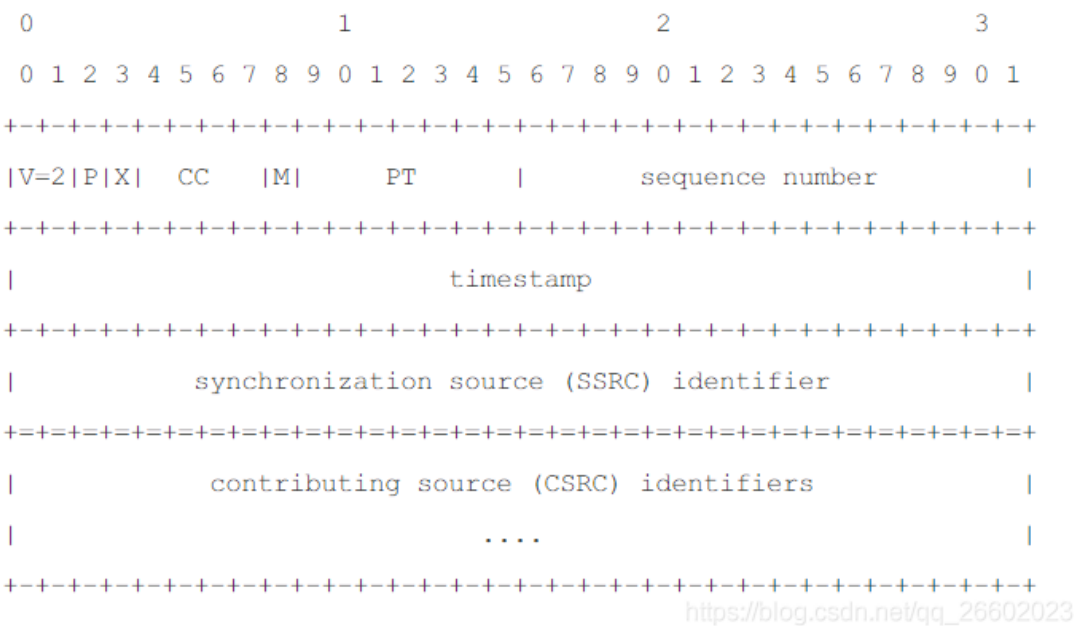

Fig. 4. The RTP Header
(The Figure is Retrived from Internet)

RTP Header. The first 12 bytes are fixed, and the CSRC can have multiple or zero bytes.

- **V:** RTP version number, which contains two characters. The current version number is 2.
- **P:** Fill flag, 1 bit, if P=1, fill the end of the packet with one or more additional 8-bit groups that are not part of the payload.
- **X:** The extension flag, which is 1, if x=1, has an extension header on the RTP header.
- **CC:** The CSRC counter, which accounts for four bits, indicates the number of the CSRC identifier.
- **M:** Different meanings for different payloads, for video, mark the end of a frame; For audio, mark the beginning of the session.
- **PT:** Payload type, 7-bits.
- **Sequence Number:** 16-bits, sequence number of RTP packets sent by the sender. The sequence number increases by 1 for each RTP packet sent.

- **Timestamp:** 32-bits, the timing of packets can be obtained from the RTP packet timestamp.
- **SSRC:** 32-bits, used to identify the synchronization source.
- **CSRC:** Each CSRC identifier is 32 bits and can have 0 to 15 CSRC. Each CSRC identifies all the supplied sources contained in the PAYLOAD of the RTP message.

2 OVERALL SYSTEM STRUCTURE AND SYSTEM PROTOCOL DESCRIPTION

2.1 Overall System Structure

The System Structure is a server-client mode. The structure diagram is shown as below:

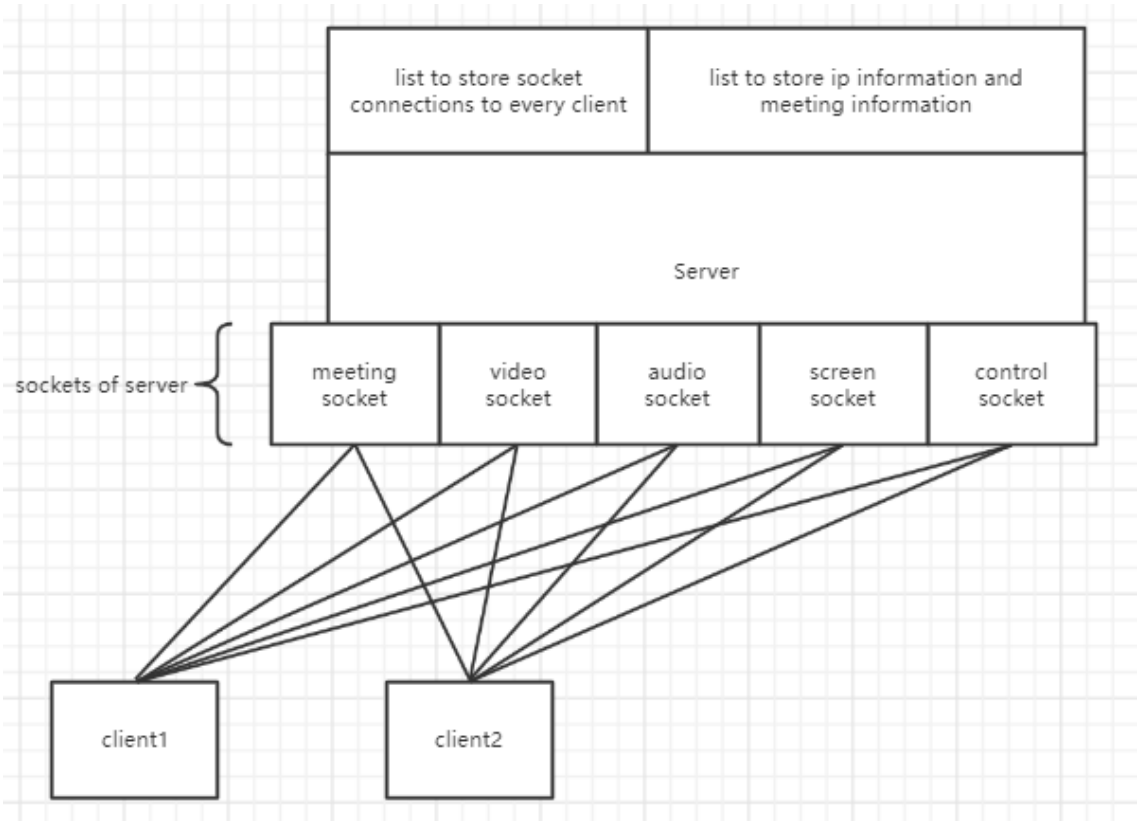

Fig. 5. The system structure of online meeting, simple view.

For each client, the connection with server is shown as below detailedly:

2.1.1 Server Implementation.

We implemented the server by two files: server.py and server_sockets.py.

The main server file implements the concurrency via multi-threading. We use a list to store current threadings. For meeting, audio, video and screen-sharing, we give them different ports. Use multi-threading to transmit these functions' information at same time.

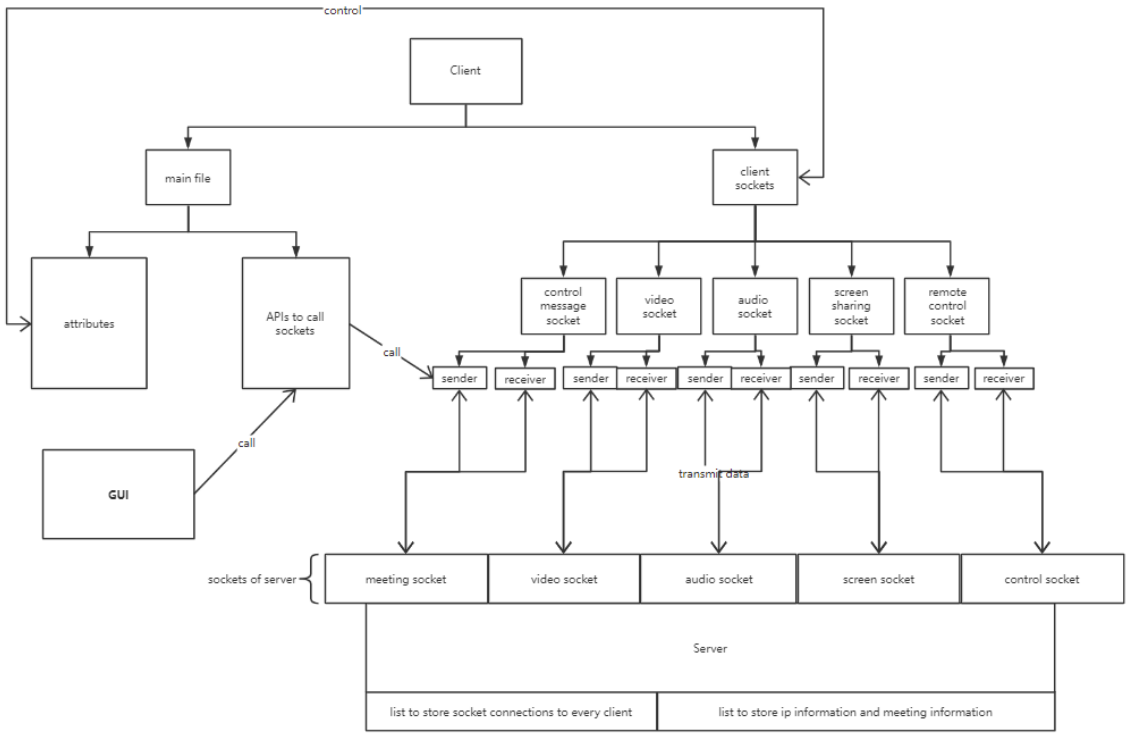

Fig. 6. The system structure of online meeting, detailed view.

In the server_sockets.py file, detailed functions of the server are implemented. Transmition of information of encrypted data, create new meetings, broadcast video, audio, and screen information and so on, are all implemented in the file.

2.1.2 Client Implementation.

We implemented the client by two files: client.py and client_sockets.py.

The main client file controls the audio, video, screen sharing status, and contains meeting number etc. The client file is the console of client, which is connected with the GUI. We can use the meeting-console interface to determine the client's behavior.

The GUI is implemented by PyQt5 and QtDesigner. Use QtDesigner to develop the interface, then use a command to transform the generated .ui file into a .py file. In the last, connect the GUI with the client to complete the system.

The client_sockets.py file implemented detailed functions of the client. It uses multi-threading to receive data and send data. In its sender and receiver, the data are encrypted and decrypted. For each different function, a exclusive socket is created and used.

In main file, the APIs send request through the senders in the client_sockets.py file. By these senders, the requests are sent to the server. The data transmitted by the server is received by the receiver. And the receiver will change the attributes in the main client file, Correspondingly.

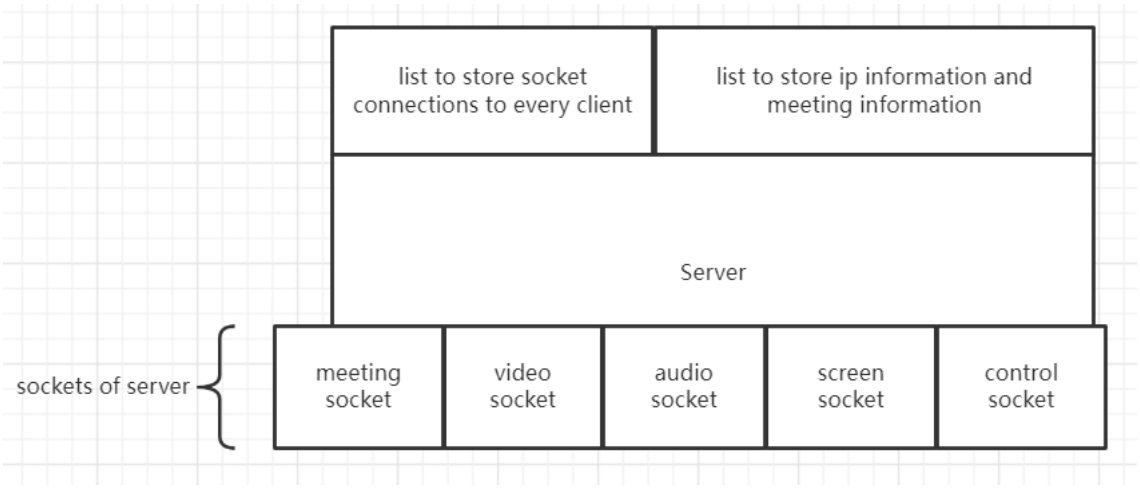

Fig. 7. The server structure of online meeting.

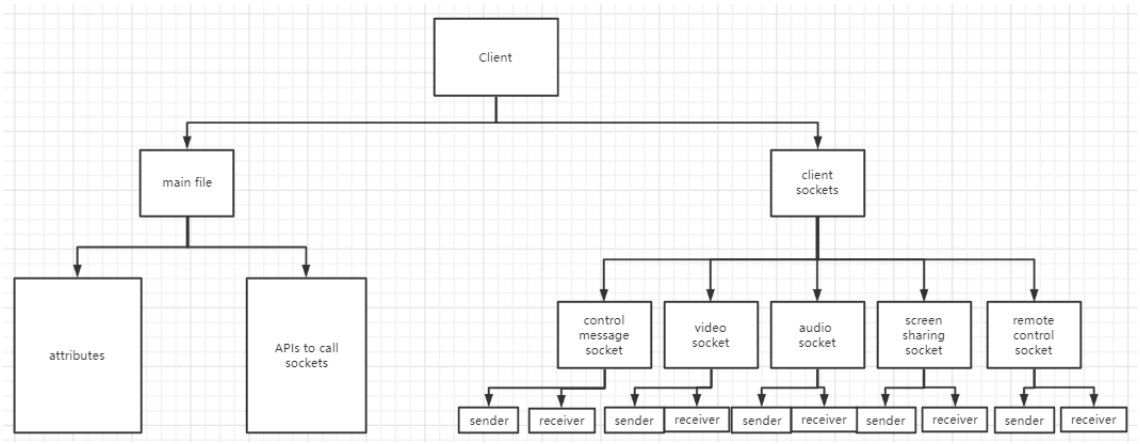

Fig. 8. The client structure of online meeting.

2.2 System Protocol Description

2.2.1 Conference Related Protocols.

SSDP, Simple Service Discovery Protocol is a application layer protocol used by the online conference. It is one of the core protocols that constitute UPnP technology. Simple Service Discovery Protocol provides a mechanism for discovering devices in a local area network. The control point (that is, the client receiving the service) can use the simple service discovery protocol to query the device that provides the specific service in its local network according to its own needs. The device (that is, the server that provides the service) can also announce its presence to control points in its local network by using the Simple Service Discovery Protocol.

Online meetings use many protocols, among which the transport layer protocol is TCP. TCP ensures the reliability of data transmission and realizes the transmission of conference information. But at the same time, it also brings high latency.

We implemented encrypted transmission by using AES(ECB) encryption. The keys of AES(ECB) are transmitted by RSA Encryption.

Advanced Encryption Standard (AES) in cryptography, also known as Rijndael Encryption, is a block Encryption Standard adopted by the U.S. federal government.

ECB(Electronic Codebook) mode is one of the most basic working modes of block cipher. In this mode, the information to be processed is divided into appropriate groups, and then each group is encrypted or decrypted independently.

The RSA public-key cryptosystem uses different encryption keys and decryption keys. It is computationally infeasible to derive decryption keys from known encryption keys. In the public-key cryptosystem, the encryption key PK is the public information, and the decryption key SK is the secret key. Encryption algorithm E and decryption algorithm D are also public. Although the decryption key SK is determined by the public key PK, the SK cannot be calculated from PK.

2.2.2 Real-time Audio Transmission.

PyAudio is a Python open source toolkit that provides access to voice manipulation. It provides recording playback processing and other functions, can be regarded as the voice field of OpenCv. Meanwhile, PyAudio is an audio processing module that can be used to stream audio to a computer sound card. In theory, this module can play back any valid audio frame decoded by the decoder.

To implement real-time audio transmission, the PyAudio module is used. We use PyAudio to record the voice from clients and play the voice at clients. A port of client is used to transmit audio between the client and the server by TCP.

2.2.3 Real-time Video Transmission.

OpenCV is an open source computer vision library funded by Intel. It consists of a series of C functions and a few C++ classes, and implements many general algorithms in image processing and computer vision. OpenCV has a cross-platform mid - and high-level API that includes over 300 C/C++ functions. It does not depend on other external libraries, although some can be used.

To implement real-time video transmission, the cv2 module is used. cv2 means OpenCV2, which is a python module. We use cv2 to open camera, and record videos for online meeting.

2.2.4 Remote Control.

To implement remote control, is is needed to control the other's mouse and keyboard. Also, the controlled client's screen is also needed to be shared. Therefore, PIL(Python Image Library) is used to share screen, and mouse and keyboard module are used for remote control.

PIL provides basic image processing functions. In PIL, any Image is represented by an Image object, and this class is exported by the module with the same name.

2.3 System Mechanisms Description

2.3.1 Conference Related Mechanisms.

Client-server model which supports multiple group meetings at the same time is implemented by the online meeting. All clients send data to the server and then the server would forward/broadcast data to other related clients. The group meeting including video/audio/screen sharing and one-to-one remote desktop control.

We designed a simple but easily-used GUI(Graphic User Interface) for the online conference client. The server is controlled by CLI, since the server is invisable to users.

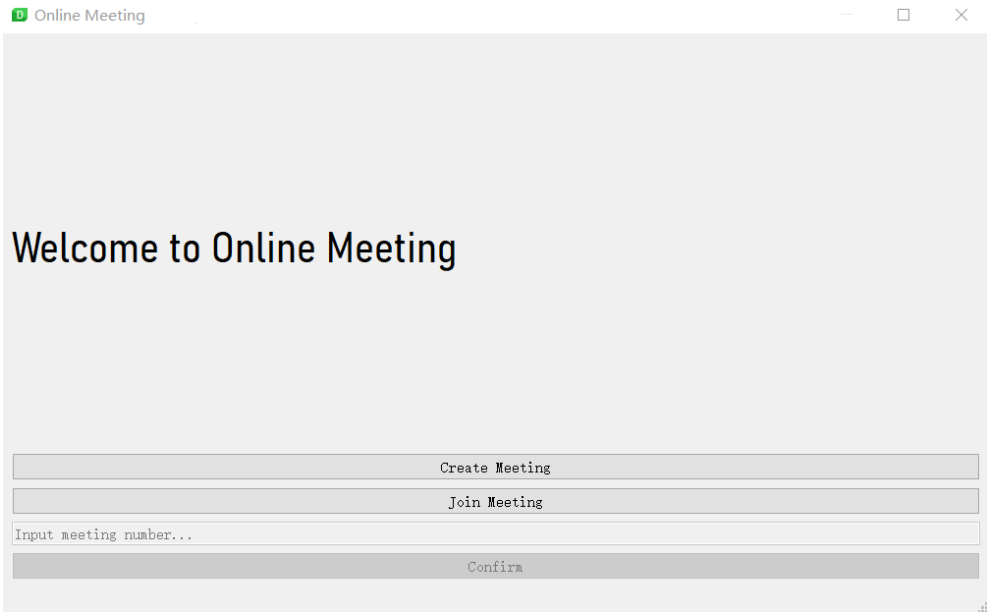

Fig. 9. The start interface of online meeting.

To start a online meeting, the server is firstly demanded to run and ready to be connected. Then, start to run clients. To create a meeting, the user just need to click the button ”Create Meeting” on the start interface of clients. A random number is generated by the server as the meeting number. At this time, there are just one person at this meeting.

Then, if there’s another client started, it can also create another meeting or choose to join the existing meeting. To join a existing meeting, the user need to click the button ”Join Meeting”, then input the meeting number at the input box. After inputting the meeting number, click the confirm button, the user will join the meeting successfully. If the meeting number is not exist, a prompt message appears.

After one joined a meeting or created a meeting, the meeting-console interface will occur and the start interface will disappear. There are seven functions on the interface, they are ”Share Screen”, ”Control Others Screen”, ”Agree”, ”Share Video”, ”Share Audio”, ”Leave” and ”Close”. As there names indicated, ”Share Screen” means share one’s screen, and ”Share Video”, ”Share Audio” are similar. To control others screen, first, click the ”Control Others Screen”, then input the ip of another client, then click ”Confirm”. The ”Agree” button is used to accept others request for control.

2.3.2 Transmission Control Mechanisms.

We use lient-server model which supports multiple group meetings at the same time to implement the online meeting. All clients send data to the server and then the server would forward/broadcast data to other related clients. The clients and the server are connected by TCP connection.

After the three-way handshake between client and the server, the server will generate a public key and a secret key of RSA algorithm. Then the server transmit the public key by TCP to the client. After receiving the public key, the client will generate a AES(ECB) key, then encrypts the AES(ECB) key by the RSA public-key as data. By using TCP, the client transmit the data to the server. The server will decrypt the data by using RSA private key, and the AES(ECB) key is obtained.

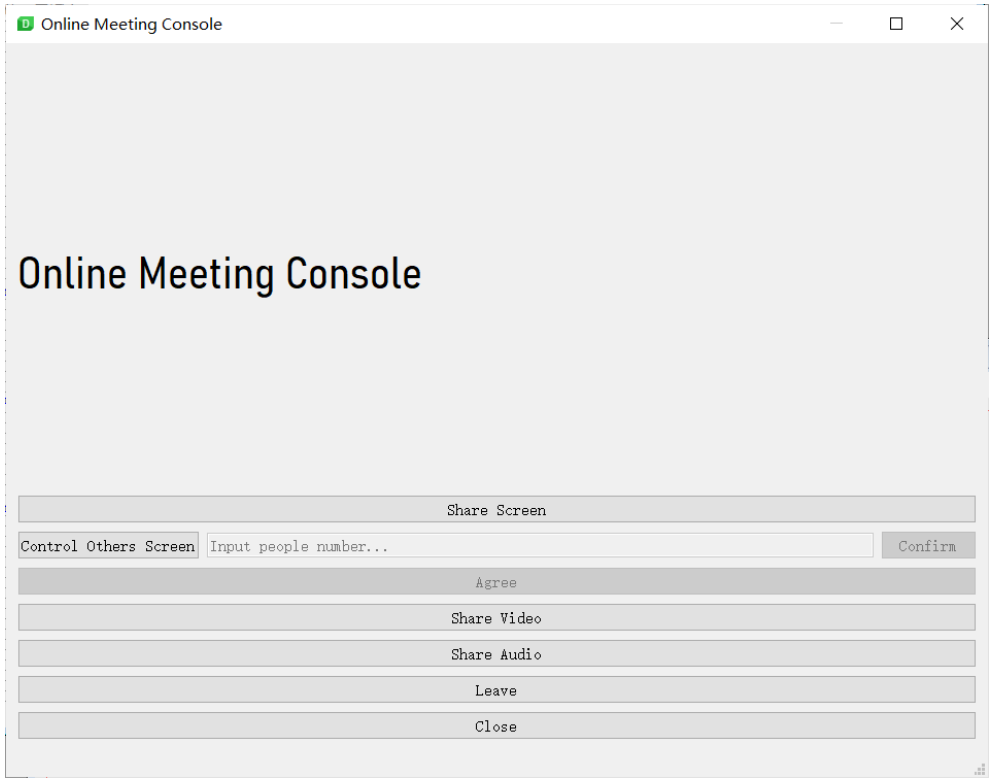

Fig. 10. The meeting-console interface of online meeting.

Since then, all of the data transmitted between the client and the server is encrypted and decrypted by the AES(ECB) key and transmitted by TCP.

3 SYSTEM TESTING RESULTS AND PACKET CAPTURE ANALYSIS

3.1 System Testing Results

3.1.1 Basic Functions.

Connect client with server.
When the GUI started, a client object is created. It will find the server by its ip, and connect with it by TCP.

Create Meeting.
Click the ”Create Meeting” button, it shows that the meeting successfully created. The randomly generated meeting number is ’469’ as shown in the CLI.

Join Meeting.
Click the ”Join Meeting” button, then input the meeting number in the input box, then click ”Confirm”. Then we will go into a existing meeting.

Leave Meeting.
Click the ”Leave Meeting” button, then we will leave the current meeting.

Fig. 11. Client successfully connected to the Server.

Fig. 12. Client successfully created a meeting.

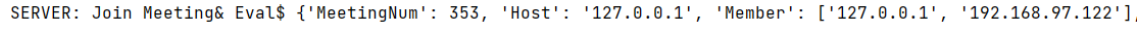

Fig. 13. Client successfully joined a meeting.

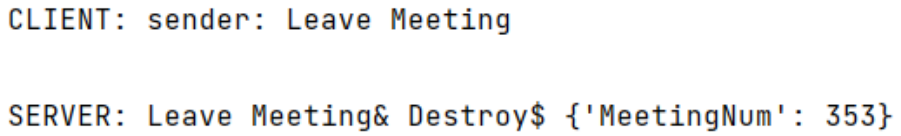

Fig. 14. Client successfully left a meeting.

Share Video.
Click the ”Share Video” button, then the success information is occured and the video window is shown.

Share Audio.
Click the ”Share Audio” button, then the success information is occured.

Screen Sharing.
Click the ”Share Screen” button, then the success information is occured. Also, the screen cab be seen by other clients.

Remote Control.
Click the ”Control Others Screen” button, then input the client’s id in the input box, then click ”Confirm”. Then we can control other’s screen.

```
 CLIENT: sender: Start Video Sharing

 SERVER: Start Video Sharing& Message$ video sharing permitted
********************
********************
********************

_CLIENT: client video run
```

Fig. 15. Success information of sharing video.

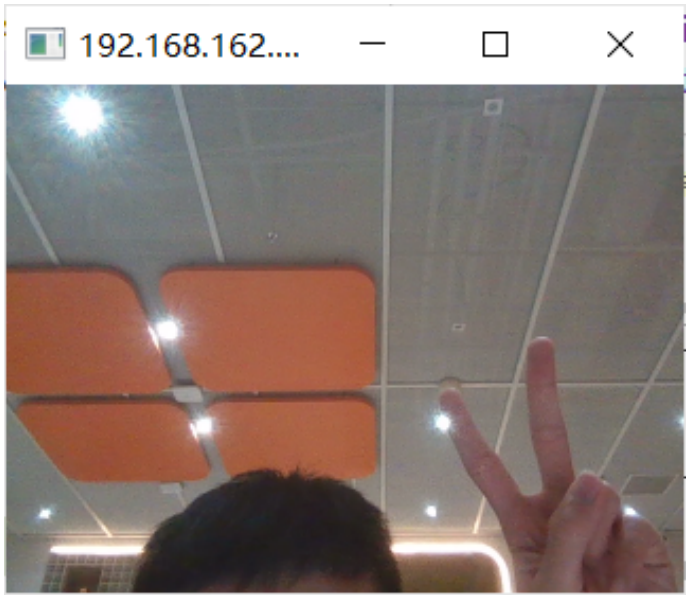

Fig. 16. The video window.

```
CLIENT: sender: Start Audio Sharing

SERVER: Start Audio Sharing& Message$ audio sharing permitted
```

Fig. 17. Success information of sharing audio.
Then one can here voice of the client side.

Multi Meeting Support.

Exception Handling Mechanisms.
When one client drops errorly and suddenly, the meeting will continue going.

3.1.2 *Advanced Functions.*

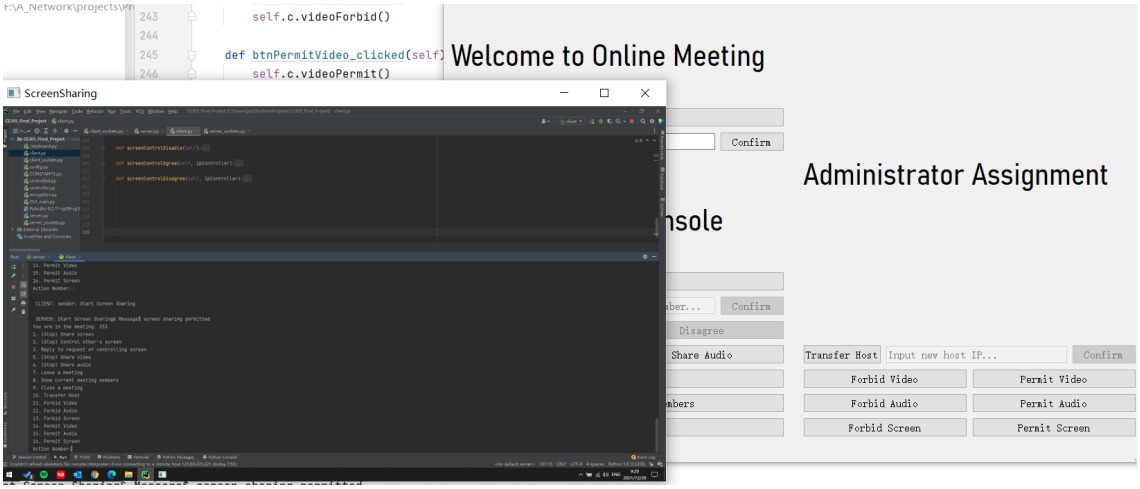

Fig. 18. Others screen shared can be seen.

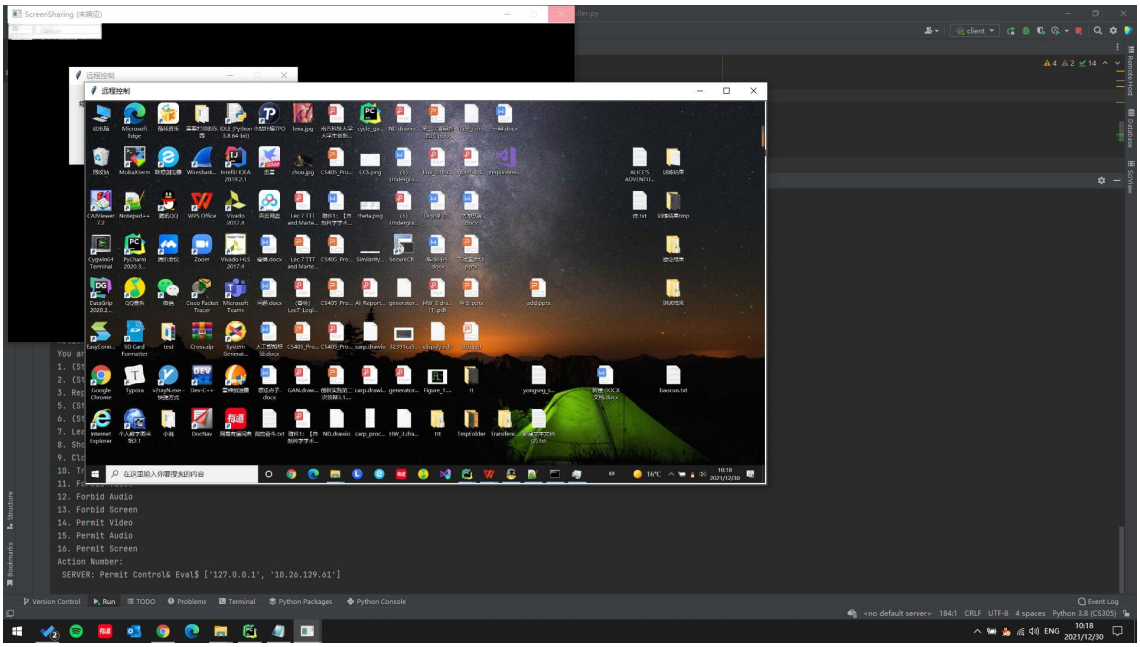

Fig. 19. The screen can be controlled by other clients.

Reasonable Mechanism Design Test.

We implemented conferencing management mechanisms, e.g., host transfer, administrator assignment, and so on.

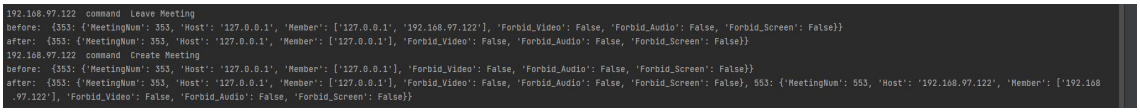

Fig. 20. The server can support multi meeting.

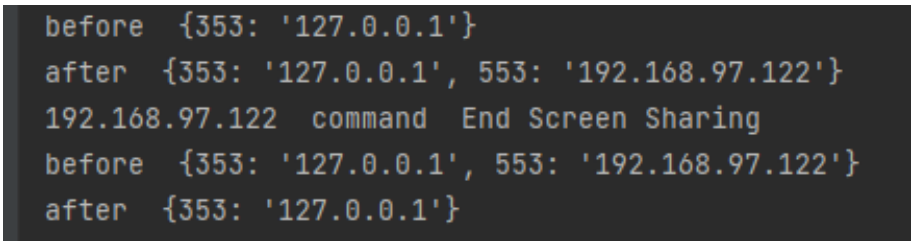

Fig. 21. The meeting continue going without the droped client.

The host of the meeting is the administrator. As an administrator, one can forbid video, forbid audio, forbid screen, permit video, permit audio, and permit screen. Meanwhile, the administrator can transfer the host to other persons in the same meeting.

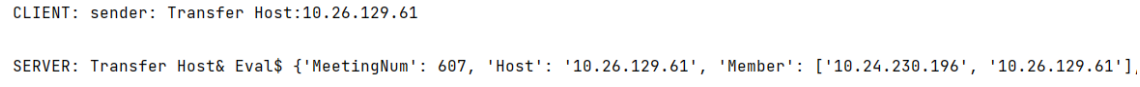

Fig. 22. The host is transfered to 10.26.129.61.

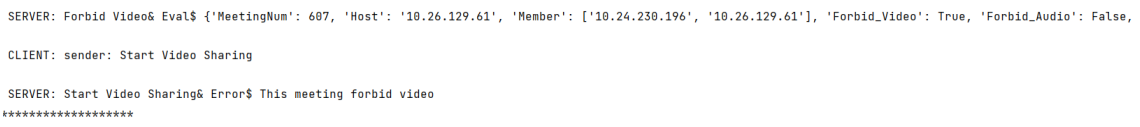

Fig. 23. The host can set forbid video.
In this case, it is forbidden for clients to share their video.

Reasonable and Tidy GUI.

3.1.3 *Additional Functions.*

Implementation of Existing Protocols.

We implemented RDP protocol, which means Remote Desktop Protocol. RDP is based on TCP/IP. As a large amount of data is transmitted, a network connection layer is defined at the bottom layer of the protocol.

Transmission Security.

The data transfered between server and clients are encrypted by AES(ECB). While the key of the AES(ECB) encryption is encrypted by RSA algorithm. Therefore, the reliability of data is guaranteed. Through looking into the packets captured by Wireshark, we can see the details of the encryption. And we cannot decrypt the data from the packets without the AES(ECB) key.

```
CLIENT: sender: Close Meeting

SERVER: Close Meeting& Destroy$ {'MeetingNum': 573}
```

Fig. 24. The host can close the current meeting, while other user cannot.

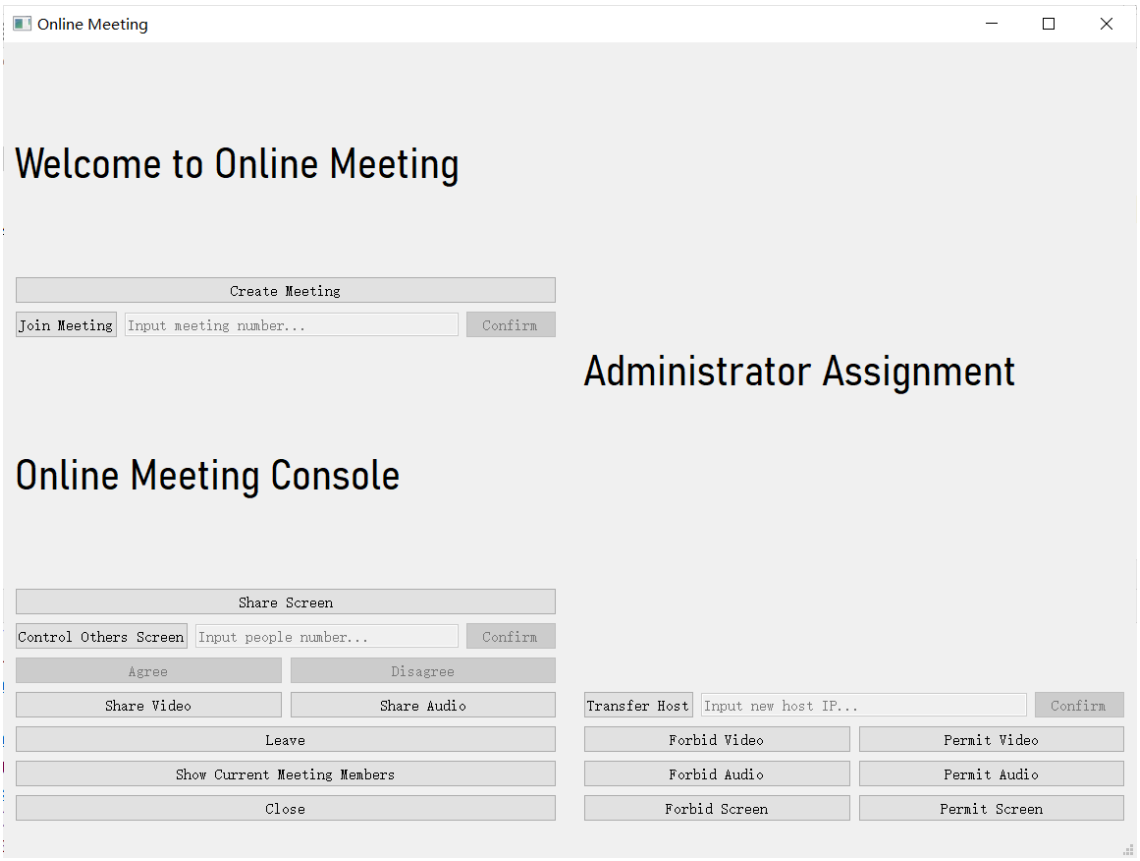

Fig. 25. The GUI designed by PyQt5 is used.
The GUI is tidy and simple.

3.2 Packet Capture Analysis

By using Wireshark, we captured all the packets from establishing the conference link to ending the conference. These packets include SSDP packets, TCP packets, TCP Dup ACK packets, TCP retransmition packets and so on. Through these packets, we can clearly see TCP's three-way handshake and four-way wave.

At the beginning, four SSDP packet are captured. They are used by the client to broadcast its existance.

Then TCP three-way handshake started between the client and the server to build a connection.

After the TCP three-way handshake, the Public key of RSA is transmitted by TCP from the server to the client.

184	0.824708	10.24.230.196	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
373	1.966163	10.24.230.196	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
573	2.950460	10.24.230.196	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
782	3.935156	10.24.230.196	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1

Fig. 26. SSDP packets occured when the client is started.

902	4.430484	10.24.230.196	10.25.217.34	TCP	66 53213 → 2222 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
903	4.430555	10.25.217.34	10.24.230.196	TCP	66 2222 → 53213 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
904	4.433753	10.24.230.196	10.25.217.34	TCP	54 53213 → 2222 [ACK] Seq=1 Ack=1 Win=131328 Len=0
1035	5.176904	10.25.217.34	10.24.230.196	TCP	524 2222 → 53213 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=470
1099	5.413493	10.24.230.196	10.25.217.34	TCP	398 53213 → 2222 [PSH, ACK] Seq=1 Ack=471 Win=130816 Len=344
1100	5.413493	10.24.230.196	10.25.217.34	TCP	66 53214 → 2223 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1101	5.413593	10.25.217.34	10.24.230.196	TCP	66 2223 → 53214 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1102	5.455308	10.25.217.34	10.24.230.196	TCP	54 2222 → 53213 [ACK] Seq=471 Ack=345 Win=65280 Len=0

Fig. 27. SSDP packets occured when the client is started.

```
0030  01 00 d6 08 00 00 50 61  73 73 77 6f 72 64 20 26   ·····Pa ssword &
0040  20 6d 65 73 73 61 67 65  24 20 2d 2d 2d 2d 2d 42    message $ -----B
0050  45 47 49 4e 20 50 55 42  4c 49 43 20 4b 45 59 2d   EGIN PUB LIC KEY-
0060  2d 2d 2d 2d 0a 4d 49 49  42 49 6a 41 4e 42 67 6b   ----·MII BIjANBgk
0070  71 68 6b 69 47 39 77 30  42 41 51 45 46 41 41 4f   qhkiG9w0 BAQEFAAO
0080  43 41 51 38 41 4d 49 49  42 43 67 4b 43 41 51 45   CAQ8AMII BCgKCAQE
0090  41 74 2b 63 6d 53 44 39  61 38 78 56 6c 72 55 77   At+cmSD9 a8xVlrUw
00a0  6c 41 75 6b 59 0a 73 43  43 56 38 4a 73 54 37 76   lAukY·sC CV8JsT7v
00b0  76 2b 74 44 4c 69 59 43  6d 43 63 6a 4b 62 47 5a   v+tDLiYC mCcjKbGZ
```

Fig. 28. Public key of RSA is transmitted by TCP.

At the same time, we can clearly see how the online meeting system works when the network is congested.

11262	49.868592	10.25.217.34	10.24.230.196	TCP	1514 [TCP Retransmission] 2223 → 53214 [PSH, ACK] Seq=1730505 Ack=1 Win=131328 Len=1460
11263	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1526913 Win=2146048 Len=0
11264	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1534213 Win=2146048 Len=0
11265	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1580933 Win=2114048 Len=0
11266	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1627653 Win=2067200 Len=0
11267	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1674373 Win=2020608 Len=0
11268	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1721093 Win=1973760 Len=0
11269	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1726933 Win=1967872 Len=0
11270	49.954091	10.24.230.196	10.25.217.34	TCP	54 53214 → 2223 [ACK] Seq=1 Ack=1731965 Win=1963008 Len=0
11271	49.956671	10.24.230.196	10.25.217.34	TCP	66 [TCP Dup ACK 11270#1] 53214 → 2223 [ACK] Seq=1 Ack=1731965 Win=1963008 Len=0 SLE=1730505 SRE=1731965
11312	50.112804	10.25.217.34	10.24.230.196	TCP	1514 2223 → 53214 [ACK] Seq=1731965 Ack=1 Win=131328 Len=1460

Fig. 29. TCP retransmition occured when the network is congestion.

4 SUMMARY AND FUTURE WORKS

4.1 Summary

Through this project, we have a deeper understanding of the knowledge of computer networks. We learned that implementing an online meeting system requires a lot of protocols and a lot of work. In the beginning, we knew nothing about the protocols and practices of video conferencing. However, we successfully completed this project through learning and collecting materials, which gave us a great sense of achievement.

We've never made GUIs in Python before, so we don't know much about Python making GUIs. After learning PyQt5 and Qt Designer, we also successfully made a normal use of the graphical user interface, making this online conference system more complete.

In addition, we have addressed the GUI and functionality linking, which is difficult and challenging. We use multi-threading to solve the problem of interface stuck, and achieve multi-access conference. Through this project, we practiced the ability to use multithreading, which was one of the important things we learned.

4.2 Future Works

4.2.1 About GUI.

Our online meeting system is still rudimentary and needs further improvement. In the future, in addition to making guIs more beautiful, we will be able to integrate video, screens, and so on into the GUI to make it a complete system.

4.2.2 About Video Transmission.

Currently, our video and screen controls have been successfully implemented, but there is still a lot of room for improvement. Our current video travels slowly, with some lag and stuttering. Therefore, optimizing the video function is also the direction of our future efforts. Trying to learn and use better protocols may be a solution.

4.2.3 About System Structure.

Currently, our online conference system structure is server-client based. This is a viable system architecture, but not necessarily the best system architecture. The application of appropriate design patterns will optimize the structure of the system for future improvement and expansion. To study better system structure and operation logic may be one of our future efforts.

ACKNOWLEDGMENTS

To Professor ZHANG Jin, TA WANG Wei and SAs, thanks for your patient guidance and dedication.

REFERENCES

[1] Schulzrinne, H., & Rosenberg, J. (1998, July). A Comparison of SIP and H. 323 for Internet Telephony. In Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV) (pp. 83-86). sn.

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., ... & Schooler, E. (2002). SIP: session initiation protocol.

[3] Peterson, J. (2002). A privacy mechanism for the session initiation protocol (SIP). RFC 3323, November.

[4] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H. 264/AVC video coding standard. IEEE Transactions on circuits and systems for video technology, 13(7), 560-576.