An Online Conferencing System Implementation

Description:

You are required to implement an online conferencing system based on the **client-server model** which is similar to a simpler version of Zoom/Tencent Meeting. To complete this project, you need to **leverage existing networking protocols** or **design your own protocols**.

This system should support the following features:

1. **Group meeting** including video/audio/screen sharing
2. One-to-one **remote desktop control**.
3. A reasonable **conferencing mechanism** including **creating, joining, leaving, closing** a conference.
4. **Client-server model**. (All clients send data to the server and then the server would forward/broadcast data to other related clients)

Team Members:

The number of members is no more than **3**.

Assessment Criteria (100 points + 20 points):

1. **Report and presentation** (20 points).
   (1) Background and related network protocol survey
   (2) Overall system structure and system protocol description
   (3) System testing results and packet capture analysis
   (4) Summary and future works

2. **Code** (80 points = 60 points for basic features + 20 points for advanced features).
   (1) **Basic features** (60 points)
      ① Network conferencing mechanism including creating, joining, leaving, closing a conference. (10 points)
      ② Real-time video transmission. (10 points)
      ③ Real-time audio transmission. (10 points)
      ④ Remote desktop sharing and control. (10 points)
      ⑤ Client-server model which supports multiple group meetings at the same time. (10 points)
      ⑥ Exception handling mechanisms. The server and client would continue working even if there are exceptions/errors. (5 points)
      ⑦ Necessary and clear comments. (5 points)
   (2) **Advanced features** (20 points)
      ① Reasonable mechanisms, including but not limited to (10 points)
         1) Conferencing management mechanisms, e.g., host transfer, administrator

assignment, and so on.

2) Transmission control mechanisms such as the message of starting and ending the transmission.
3) Remote desktop control mechanism which should include the request, reply, release, and so on.

② User-friendly interface (CLI/GUI). (10 points)

3. **Bonus** (20 points, optional). Complete additional meaningful functions. Including but not limited to the following:

(1) Implement existing protocols (SIP, RTP, RDP, etc.)
(2) Transmission security. You need to simulate the attack without a security design and then show how your design ensures transmission security.

Reference:

1. Remote desktop protocol review: https://www.cnblogs.com/qpanda/p/4331782.html
2. Instant multi-person voice chat: https://github.com/TomPrograms/Python-Voice-Chat
3. Audio chat project based on P2P model: https://github.com/theintencity/p2p-sip
4. Video chat project based on client-server model: https://www.lanqiao.cn/courses/672
5. SIP protocol: http://www.kamailio.org/docs/tutorials/sip-introduction/
6. SIP protocol (RFC3261): https://www.rfc-editor.org/info/rfc3261
7. RTP protocol (RFC3550): https://www.rfc-editor.org/info/rfc3550
8. Demo for RTP protocol: https://github.com/gabrieljablonski/rtsp-rtp-stream

Possibly needed packages:

1. Network: socket
2. Multi-threading: threading, asyncio
3. The control of mouse and keyboard: mouse, keyboard
4. Image: PIL, cv2, Image
5. Voice：Pyaudio
6. GUI：Tkinter、Qt

The framework is provided in *online_conferencing_framework* folder.