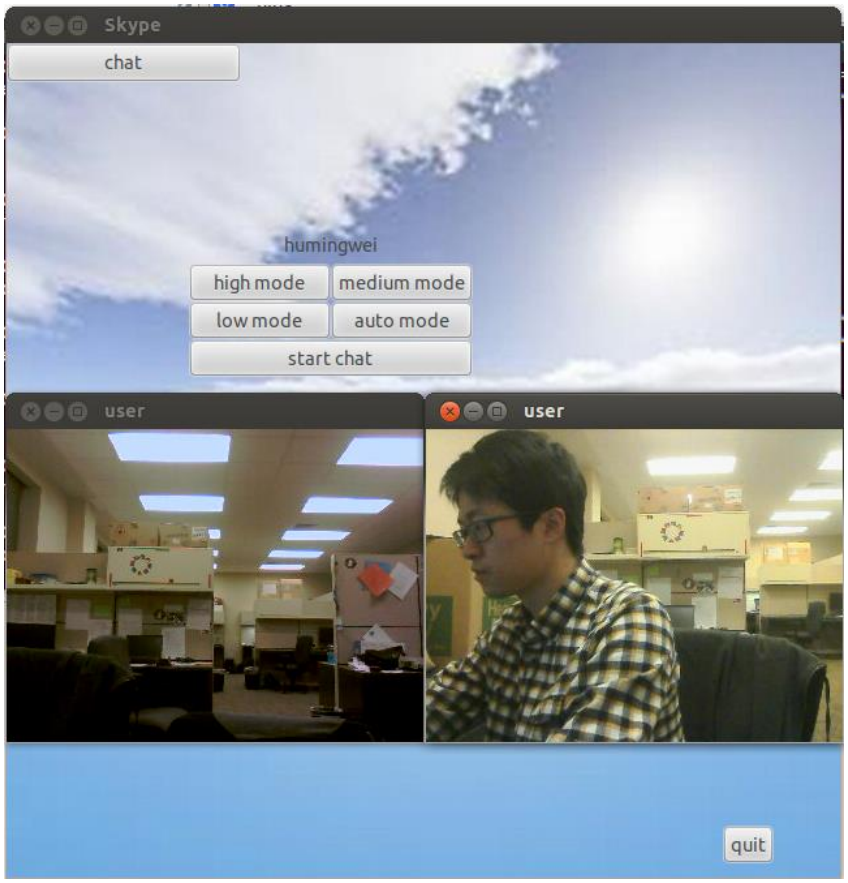
CS414 MP3: Live-conference P2P Streaming System

Mingwei Hu, Chang Li, He Huang

Introduction:

In this final MP, we will use all the components built in previous MPs to build an online conference P2P streaming system. Our Live-conference system enable multiple people video chatting/conference at the same time. This video streaming is interest oriented, double direction and bandwidth adaptation. Each user in our system is interest oriented, as it only have video chatting with the user they are interested. The video chatting is double direction, as the user sending and receiving video streaming simultaneously. The most important character of our system is that it's bandwidth adaptation, it can switch between different video modes based on the networking environment.



Design & Algorithm:

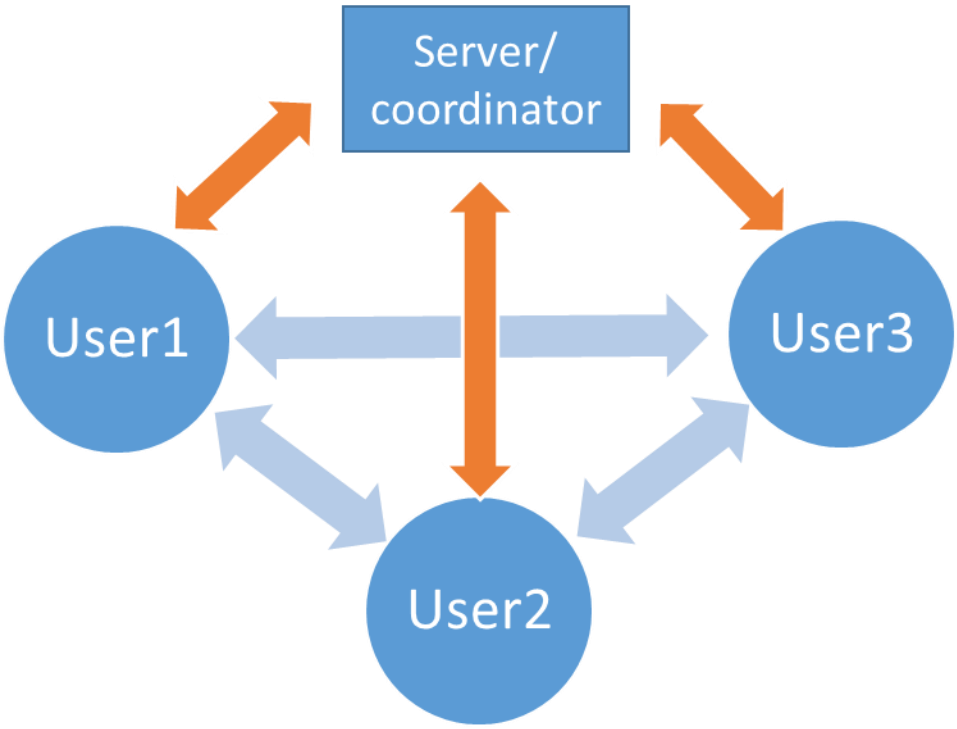
Each user in our Live-conference system composed of three parts, the GUI thread, the control communication thread (with coordinator), and video/audio streaming thread (with other user). As there is no GUI requirement for the coordinator, the coordinator only has control communication thread. Similar to the concept of P2P system, for each user itself is a video streaming server to the other connected user, it's also one client to the other user.

When one user has a video streaming request, it first send its request to the coordinator, information include the destination node, the bandwidth information of itself and its desired video streaming mode (include high quality, low quality, or mute mode).

Then the coordinator make decision according to the bandwidth situation of current network and the network situation of the desired user. The decision include decline the request, permit with the desired mode, upgrade to a higher mode, or decrease to a lower mode. Then the coordinator send messages to the corresponding user, to tell them to build the dynamic pipeline and start video streaming.

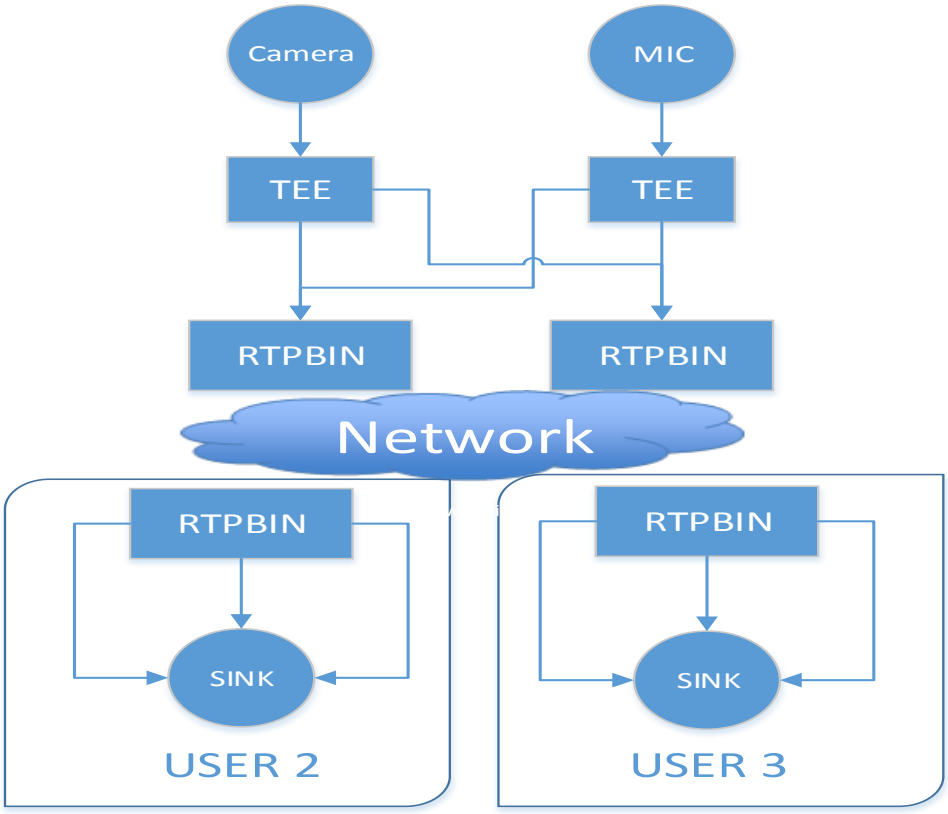
The GUI part is in charge of the user interface and controls the communication thread to let client and server communicate with each other. These communication include reservation, QoS negotiation, and video operation command, such as, Play, Stop, mode switch…

The algorithm and code design as showed below:



Picture 1: Multi-thread logic design

Key Element 1: Dynamic pipeline



Gstreamer has one big weakness it that it cannot efficient apply dynamic pipeline element modification. In this MP, we did great work to make our pipeline dynamic and change over the situation changes. For example, one user has one camera and one camera's corresponding source element, if user1 start video streaming with user2, in user1's pipeline, it only need one branch of the tee element before the RTPBIN element. After a while, the user3 join the video chatting, now in user1's pipeline the tee element need another branch to send the video streaming to user3. So we come out one efficient way to dynamic change the pipeline element user1.

Analysis of Coordinator and Decision Tree

The coordinator is the manager of the network status, it also in charge all the video streaming request of each user, one great feature of our system is that it can dynamically adjust bandwidth between different users.

One simple cases is that when user1 and user2 start video streaming, at the beginning, they all select Auto mode, which means, they want to acquire the maximum of bandwidth available between in user1 and user2. Since there is no connection before, the channel between user1 and user2 is High mode.

Now the network status is :

User1 <==> User2 : High mode

After a while, user 3 join the chatting room and start video streaming with user2, (we assume the max bandwidth of each user is 2 Low mode and one High mode). Then the coordinator decrease pervious bandwidth of channel user1 and user 2, from high mode to low mode, thus enable user2 chat with user3 in low mode. Now the network status is :

User1 <==> User2 : Low mode

User3 <==> User2 : Low mode

Sometime later, user3 finish chatting with User 2 and leave, the coordinator find the available bandwidth of user1 and user2 can upgrade to High mode, so the coordinator send message to user1 and user2 to upgrade/increase the bandwidth to High mode.

Above is one simple case of dynamic allocation bandwidth of our system coordinator, inside the coordinator, it use the concept of P2P system manager and have a decision tree to decide which channel need to upgrade or down grade to desired communication level.