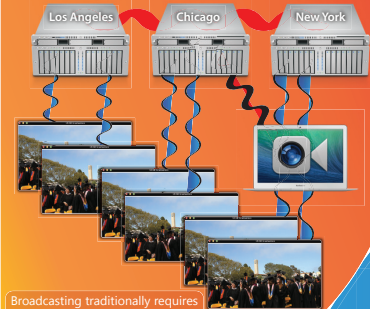


Problem with Today's Infrastructure



Broadcasting traditionally requires sending video to a server. Thousands of viewers means thousands of video streams from service provider's servers to users.

Problem

Live broadcasting services require great technical and monetary resources, making it difficult for alternative services to be created. Existing solutions such as YouTube Live, Ustream, and Twitch have issues with requiring expensive centralized servers, additional third-party applications, or even paid subscriptions to broadcast.

Solution

Using the emerging web technology WebRTC, live video broadcasting can be implemented using peer to peer methods. WebRTC is a framework that allows web browsers to connect and send multimedia to each other directly. Users wanting to watch a live broadcast automatically connect to a user already watching the same broadcast — specified by the P2PCast server — who rebroadcasts it to them. P2PCast alleviates the need for expensive infrastructure and additional third-party applications for users.

P2PCast Homepage

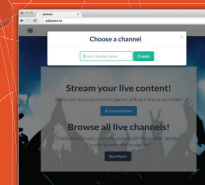


Broadcaster

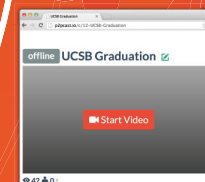
Step 1 User wants to broadcast live video

Step 2 User registers

Step 3 User creates a new channel



Step 4 Authorize webcam to be used by web browser



WebRTC Connection Setup

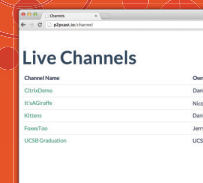
WebRTC uses the concept of a handshake, where one peer makes an offer and the other side replies with an answer. WebRTC also requires that connection information be transmitted from one peer to another peer. This connection information is known as ICE, or interactive connectivity establishment, which is how both peers figure out how they are connected to the Internet.

Initially, assistance is needed to send these small messages between the two peers. The P2PCast server routes these messages between the peers until the connection has been established.

Viewer

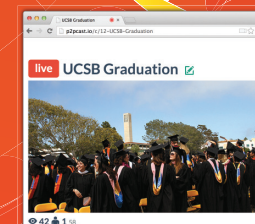
Step 1 User wants to watch a live broadcast

Step 2 User chooses a channel with a broadcast

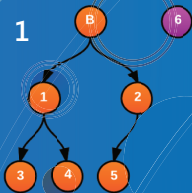


Step 3 User's web browser contacts P2PCast server to direct it to another user

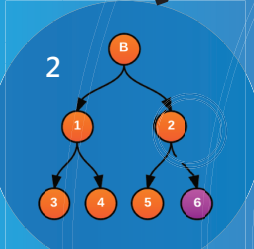
Step 4 User's web browser connects to another user and the live broadcast begins



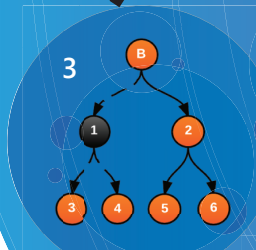
Step 5 Done!



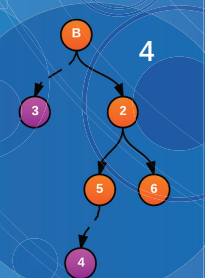
State 1 A new user, identified with the number 5 (peer 5), wants to begin viewing the live broadcast.



State 2 Peer 5 has been added to the tree and its connection to peer 2 is monitored for any disconnection.



State 3 Peer 1 is detected as disconnecting and so peer 3 and peer 4 must get replacement upstream peers.



State 4 Peers 3 and 4 have replacement upstream peers selected (similar to new peers) and each user's live streams restart.



P2PCast Server WebRTC allows for peer connections, but it can not setup or maintain connections on its own. The server coordinates the connection of peers using the tree and monitors each peer for issues, allowing for reconnects.

UCSB Computer Science Capstone 2014

P2PCast

Team NP-Compete

TEAM | Daniel **Vicory** Nicole **Theokari**
MEMBERS | Jerry **Medina** Omar **Masri** Justin **Liang**

SPECIAL THANKS | Citrix Allan **Knight** Ashish **Thapliyal** Hang **Yu**
UCSB Chandra **Krintz** Timothy **Sherwood**
Geoffrey **Douglas** Janet **Kayfetz**

node.js

sails

WebRTC

HTML5

CITRIX®