WebRTC

Advisor: Professor Chris Pollett

Submitted by: Yangcha K. Ho

March 27, 2018
What is WebRTC?

• It is a collection of communications protocols and APIs.

• It enable peer-to-peer video, audio, and data communication in a web browser.

• WebRTC is an open source project/tool.

• It is not a ready to use application.
WebRTC APIs: (2,4)

• The WebRTC comes with JavaScript APIs:
  • MediaStream: capture audio and video/record audio and video.
  • RTCPeerConnection: stream audio and video between users.
  • RTCDataChannel: stream data between users.
WebRTC does not come with signaling methods and protocols:

• WebRTC is not a standalone API.
• It needs to do several things to make it work.
• One of the critical parts in building WebRTC is taking care of signaling methods and protocols which are not part of WebRTC standards.
What is signaling?
signaling continued:

• Signaling is the process of coordinating communication between clients who need a way to exchange information between them.
• For two devices in different networks to find each other they need to use a central service called a signaling server through which two devices can discover each other and exchange messages.
• WebRTC does not specify signaling methods or protocols.
Signaling continued:

• Once signaling has taken place, video/audio/data is streamed directly between clients, using WebRTC's PeerConnection API.

• This peer-to-peer direct connection allows you to stream high-bandwidth robust data, like video.

• In this project, we are going to use HTML5 WebSockets – a bidirectional socket connection between two endpoints – a web server and a web browser.
WebSocket:

• WebSocket: Bringing Sockets to the Web
• The WebSocket specification defines an API establishing "socket" connections between a Web browser and a server. In plain words: There is an persistent connection between the client and the server and both parties can start sending data at any time.
WebSocket continued:

• You open up a WebSocket connection simply by calling the WebSocket constructor:

• WebSocket Code snippet:

```javascript
var connection = new WebSocket('ws://html5rocks.websocket.org/echo', ['soap', 'xmpp']);
```
WebRTC compatible browsers and smartphones:

• Chrome
• Firefox
• Edge
• Opera
• PC operating systems such as Mac OS X, Windows, Linux, and Android.

• The Web site: [http://caniuse.com/#feat=rtcpeerconnection](http://caniuse.com/#feat=rtcpeerconnection) lists up to date information.
WebRTC Architecture

JavaScript Interface

Session management and abstract information control session (libjingle)

Audio engine
- iSAC/iLBC
- Noise cancellation

Video engine
- VP8
- Image enhancement

transmission
- Multiplexing
- ICE+STUN +TURN

Audio capture and rendering

Video capture

Network I/O

System interface
RTCPeerConnection API

• The main purpose is to create a peer connection and attach media streams.

• It also manages a UDP connection with another user.

• code snippet:
  
  ```javascript
  var conn = new RTCPeerConnection(conf);
  conn.onaddstream = function(stream) {
    // use stream goes here ...
  };
  ```
RTCPeerConnection API cont.

- The object accepts a `conf` parameter.

- The `onaddstream` event is fired when the remote user adds a video or audio stream to their peer connection.
MediaStream API

- This API has three functionalities:
  - It allows a developer access to a stream object that represent video and audio streams.
  - It manages the selection of input user devices in case a user has multiple cameras or microphones on his device.
MediaStream API demo:

• One of its feature is asking user’s permission to use video
• A simple HTML page which contains a <video> element.
• MySample.html
  
  ```html
  <html>
    <head>
      <meta charset = "utf-8">
    </head>
    <body>
      <video autoplay></video>
      <script src = "client.js"></script>
    </body>
  </html>
  ```
MediaStream API demo cont.

- Code Snippet of a SampleClient.js file –

```javascript
// checks if the browser supports WebRTC
function hasUserMedia() {
    navigator.getUserMedia = navigator.getUserMedia ||
    navigator.webkitGetUserMedia ||
    navigator.mozGetUserMedia ||
    navigator.msGetUserMedia;
    return !!navigator.getUserMedia;
}

if (hasUserMedia()) {
    navigator.getUserMedia = navigator.getUserMedia ||
    navigator.webkitGetUserMedia ||
    navigator.mozGetUserMedia ||
    navigator.msGetUserMedia;
```
WebRTC Use Cases (5)

WebRTC can be used for:

- real-time marketing
- real-time advertising
- back office communications
- HR management
- social networking
- Customer Service
WebRTC Video demo:

• myWebRTC.html is started from Node server hosted site:
• It asks whether camera is allowed to use. Once click Ok from this page.
WebRTC video demo: live image appears
When clicked Takesnap, it takes a snapshot and appear both (one live, one still photo)