PeerConnection API

A few slides to get the discussion going

A simple example

Example Overview

```
// set up the call, get access to local media, and establish connectivity
function start(isCaller) {
    pc = new PeerConnection(null);

    // send any ice candidates to the other peer
    pc.onicecandidate = function (evt) {
        signalingChannel.send(JSON.stringify({ "type": "candidate", "sdp": evt.candidate }));
    };

    // once remote stream arrives, show it in the remote video element
    pc.onaddstream = function (evt) {
        remoteView.src = URL.createObjectURL(evt.stream);
    };

    // get the local stream and show it in the local video element
    navigator.getUserMedia({"audio": true, "video": true}, function (stream) {
        selfView.src = URL.createObjectURL(stream);
        pc.addStream(stream);

        var type;
        if (isCaller) {
            pc.createOffer(gotDescription);
            type = "offer";
        } else {
            pc.createAnswer(pc.remoteDescription, gotDescription);
            type = "answer";
        }

        function gotDescription(desc) {
            pc.setLocalDescription(type, desc);
            signalingChannel.send(JSON.stringify({ "type": type, "sdp": desc }));
        }
    });
}

signalingChannel.onmessage = function (evt) {
    var msg = JSON.parse(evt.data);
    switch (msg.type) {
    case "offer":
        // create the PeerConnection
        start(false);
        // feed the received offer into the PeerConnection
        pc.setRemoteDescription(msg.type, new SessionDescription(msg.sdp));
        break;
    case "answer":
        pc.setRemoteDescription(msg.type, new SessionDescription(msg.sdp));
        break;
    case "candidate":
        pc.addIceCandidate(new IceCandidate(msg.sdp));
        break;
    }
};
```

Create PeerConnection and handle generated candidates

Handle incoming streams

Request media, add media to be sent

Create offer/answer

Process incoming offers, answers, and candidates

Example Walkthrough

```
// set up the call, get access to local media, and establish connectivity
function start(isCaller) {
    pc = new PeerConnection(null);

    // send any ice candidates to the other peer
    pc.onicecandidate = function (evt) {
        signalingChannel.send(
                JSON.stringify({ "type": "candidate", "sdp": evt.candidate }));
    };

    // once remote stream arrives, show it in the remote video element
    pc.onaddstream = function (evt) {
        remoteView.src = URL.createObjectURL(evt.stream);
    };

    ...
```

Example Walkthrough

```
// get the local stream and show it in the local video element
navigator.getUserMedia({"audio": true, "video": true}, function (stream) {
    selfView.src = URL.createObjectURL(stream);
    pc.addStream(stream);

    var type;
    if (isCaller) {
        pc.createOffer(gotDescription);
        type = "offer";
    } else {
        pc.createAnswer(pc.remoteDescription, gotDescription);
        type = "answer";
    }

    function gotDescription(desc) {
        pc.setLocalDescription(type, desc);
        signalingChannel.send(JSON.stringify(
                { "type": type, "sdp": desc }));
    }
});
```

Example Walkthrough

```
signalingChannel.onmessage = function (evt) {
    var msg = JSON.parse(evt.data);
    switch (msg.type) {
    case "offer":
        // create the PeerConnection
        start(false);
        // feed the received offer into the PeerConnection
        pc.setRemoteDescription(msg.type, new SessionDescription(msg.sdp));
        break;
    case "answer":
        pc.setRemoteDescription(msg.type, new SessionDescription(msg.sdp));
        break;
    case "candidate":
        pc.addIceCandidate(new IceCandidate(msg.sdp));
        break;
    }
};
```

A few topics for discussion

Got More Candidates?

- How to tell that we got enough candidates?
  - null candidate? (as suggested by Justin on the list)
  - New (simple) event?
  - Leave it to the application (e.g. timeout)

```
pc.onicecandidate = function (evt) {
    var candidate = evt.candidate;
    //...
};
```

```
[Constructor(DOMString type, PeerConnectionIceEventInit eventInitDict)]
interface PeerConnectionIceEvent : Event {
    readonly attribute PeerConnection peer;
    readonly attribute IceCandidate   candidate;
};
```

Need to negotiate?

- Event to indicate that negotiation is needed
  - Track added/removed
  - Track ended
- Drive call setup(?)

```
[Constructor (IceServers configuration, optional MediaConstraints constraints)]
interface PeerConnection {
            ...
            attribute Function?          onrenegotationneeded;
            ...
};
```

A lot of Tweaking

- We currently got five places for tweaking when sending a stream

```
navigator.getUserMedia(constraints, function (stream) { ... }); // 1

pc.addStream(stream, constraints); // 2

pc.createOffer(function (offer) {
    offer.tweakOffer(tweaks); // 4
    pc.setLocalDescription("offer", offer);
    // ...
}, constraints); // 3

tweakSdp(offer.sdp); // 5
```

Configuring PeerConnection

- Does this need to be an object or would DOMString[] do?
  - Future extensions

```
interface IceServers {
    attribute DOMString servers[];
};
```

```
{ servers:[
    "stun:stun.example.net",
    "turn:user@turn.example.org myPassword" ]
}
```

Creating answers

- createProvisionalAnswer
- Offer matching

```
[Constructor (IceServers configuration, optional MediaConstraints constraints)]
interface PeerConnection {
    ...
    void        createAnswer (SessionDescription offer,
                              SessionDescriptionCallback successCallback,
                         optional PeerConnectionErrorCallback failureCallback,
                         optional MediaConstraints constraints,
                         optional Boolean createProvisionalAnswer=false);
    ...
};
```

ICE Restart

- Explicit API – updateIce() – then do signaling?
- Hint to createOffer()?

```
[Constructor (IceServers configuration, optional MediaConstraints constraints)]
interface PeerConnection {
    ...
    void        createOffer (SessionDescriptionCallback successCallback,
                    optional PeerConnectionErrorCallback failureCallback,
                    optional MediaConstraints constraints);
    ...
    void        updateIce (optional IceServers configuration,
                           optional MediaConstraints constraints,
                           optional Boolean restart=false);
    ...
};
```

More topics

- What’s being offered?
- Less verbose syntax on offer/answer handling
- How long is my newly created offer valid?
- ICE states (current use case?)

```
[Constructor (IceServers configuration, optional MediaConstraints constraints)]
interface PeerConnection {
    void        createOffer (SessionDescriptionCallback successCallback,
                    optional PeerConnectionErrorCallback failureCallback,
                    optional MediaConstraints constraints);
    void        createAnswer (SessionDescription offer,
                              SessionDescriptionCallback successCallback,
                     optional PeerConnectionErrorCallback failureCallback,
                     optional MediaConstraints constraints,
                     optional Boolean createProvisionalAnswer=false);
    void        setLocalDescription (SdpType action, SessionDescription description);
    readonly attribute SessionDescription localDescription;
    void        setRemoteDescription (SdpType action, SessionDescription description);
    readonly attribute SessionDescription remoteDescription;
    readonly attribute PeerState          readyState;
    void        updateIce (optional IceServers configuration,
                           optional MediaConstraints constraints,
                           optional Boolean restart=false);
    void        addIceCandidate (IceCandidate candidate);
    readonly attribute IceState           iceState;
    readonly attribute MediaStream[]      localStreams;
    readonly attribute MediaStream[]      remoteStreams;
    DataChannel createDataChannel ([TreatNullAs=EmptyString] DOMString? label,
                                                    optional DataChannelInit? dataChannelDict);
             attribute Function?          ondatachannel;
    void        addStream (MediaStream stream, optional MediaConstraints constraints);
    void        removeStream (MediaStream stream);
    void        close ();
             attribute Function?          onrenegotationneeded;
             attribute Function?          onicecandidate;
             attribute Function?          onconnecting;
             attribute Function?          onopen;
             attribute Function?          onstatechange;
             attribute Function?          onaddstream;
             attribute Function?          onremovestream;
             attribute Function?          onicechange;
};
```

Always good to have the idl at hand

```
[Constructor (DOMString description)]
interface SessionDescription {
    attribute SdpType   type;
    attribute DOMString sdp;
    stringifier DOMString ();
};


[Constructor (DOMString candidate)]
interface IceCandidate {
    attribute DOMString candidate;
    stringifier DOMString ();
};


interface IceServers {
    attribute DOMString servers[];
    // alt.
    attribute DOMString servers[][];
};


callback SessionDescriptionCallback = void (SessionDescription  sdp);


callback PeerConnectionErrorCallback = void (DOMString errorInformation);
```