Simulcast at TPAC 2015

Plan X

Plan X Example

```
var sender = pc.addTransceiver({sendEncodings: [
  {rid: “F”},
  {rid: “H”, scaleResolutionDownBy: 2},
  {rid: “Q”, scaleResolutionDownBy: 4}
]).sender;
// SDP has simulcast in it (Assuming all goes well in MMUSIC)
pc.createOffer();
// … setLocalDescription, get answer, setRemoteDescription
// Sender parameters has negotiated a subset of encodings
sender.getParameters();
```

If all goes well in MMUSIC….

```
// In the client’s description
a=rid:F send
a=rid:H send
a=rid:Q send
a=simulcast rid:F,H,Q

// In the server’s description
a=rid:F recv
a=rid:H recv
a=simulcast rid:F,H
```

And all doesn’t go well in MMUSIC

Just send it without negotiating. And leave a big warning. Maybe like this:

“If multiple encodings are set in sendEncodings, more than one simultaneous encoding may be sent even though the remote endpoint has not negotiated it. If the remote endpoint is not expecting multiple encodings, this will result in unpredictable and/or garbled results. Make sure you only do so when you are sure the receiving endpoint can handle receiving multiple encodings, either through previous arrangement or via application-level signaling”.

The general idea behind this approach is that is allows deferral of the decision to have “full, SDP-signaled” simulcast in 1.0 to be made very late in the game, with no resulting change in the JS API. If we get to the point that all other facets of the specification are ready to go, and the MMUSIC work is foundering, we can “pull the ripcord” and detach from the IETF work by simply adding such a warning.

Most importantly, this also gets us all the on-the-wire formats necessary to polyfill whatever the IETF eventually settles on.

Either way

- An API for sending multiple encodings identified by the RID header extension.
- No API for PT-based simulcast
- No API mapping for max-width, max-height, max-fs, etc
- Browsers *may* support those more advanced forms of simulcast SDP, but it’s optional. The RID portion, including the header extension, is mandatory.

Ship it?

Some Minor Questions

- Should the JS be allowed to set the RID?
  - If they pick a value too big, it creates big RTP packets (“don’t do that then”)
  - We need to define behavior for duplicates (recommendation: check for duplicates, throw exception)
  - But it gives the app flexibility to put some meaning in each simulcast layer’s RID
  - Recommendation: Let JS pick the RID (it can already via SDP munging :)
- Should we just use RtpSender.setParameters() instead of sendEncodings?
  - Problem #1: Parameters is more than just encodings. Do you have to set all the things?
  - Problem #2: Can the JS add more encodings later?
  - Recommendation: Stick with putting it in addTransceiver.
- Can a browser *receive* simulcast?
  - Recommendation: Out of scope for 1.0