

MediaTrackTransmissionController

[http://www.w3.org/2011/04/webRTC/wiki/
Transport_Control](http://www.w3.org/2011/04/webRTC/wiki/Transport_Control)

Background

- Need to control/influence some transmission related parameters identified
 - Some have a bearing to SDP
 - Other don't
 - Examples include
 - bitrates, priority, agc on/off, encoding delay, codecs, "the new thing (stop send RTP)"
- A simple proposal - modeled after DTMF-handler – included

Proposal

Will show from

http://www.w3.org/2011/04/webRTC/wiki/Transport_Control#API_proposals

Including examples

Addition to PeerConnection

```
partial interface RTCPeerConnection {  
    RTCMediaTrackTransmissionController getMediaTrackTransmissionController  
    (MediaStreamTrack track);  
}
```

New API

```
interface RTCMediaTrackTransmissionController {  
    readonly attribute MediaStreamTrack track;  
    attribute boolean theNewThing;  
}  
RTCMediaTrackTransmissionController implements  
Constrainable;
```

Initial constraints

Priority (enum: verylow, low, medium, high, veryhigh)

Bitrate (PropertyValueRange?)

Examples

```
var handlerX = PC.getMediaTrackTransmissionController(X); //Getting the  
    handler
```

```
//Build set of constraints:
```

```
var XConstraints = { optional: [  
    {"bitrate" : { max: 400}}, //max 400 kbps  
    {"priority" : 4high}  
    ] }
```

```
//Apply
```

```
handlerX.applyConstraints(transmissionConstraints, success, error); //  
    Applying the constraints
```

```
//Stop send RTP
```

```
handlerX.theNewThing = false; //ask PC to stop sending RTP
```

Remains to be specced out

- What does priority mean?
- How is bitrate measured? Leaky bucket model?
- Signaling for "theNewThing" ("negotiationneeded" event with change directional attribute in the SDP?)

Summary

- A proposal that uses the models from DTMFHandler and the Constraints API
- Initial set of constraints small - but easy to extend without breaking backward compatibility
- Only method so far: "theNewThing"