Media Stream Track Suspend/Resume/Remove/Stop

Adam Roach

Tuesday, November 12th, Shenzen, China

Monday, November 11th, Kirkland, WA, USA

What are the operations we care about here?

1. Hardware or OS mute/unmute
2. Javascript disables/enables a track
3. Javascript suspends/resumes RTP for a track
4. Javascript stops a track or removes it from the session

User Presses Hardware or OS “Mute” Button on Camera or Mic

- Application in sending browser receives “onmute” event
  - Assuming the browser can figure out that the user did this
- Sending browser may either:
  - Continue to feed whatever is coming off the hardware into the codec, or
  - Encode black frames / silence / comfort noise
- RTP flows as normal
- Receiving party receives no events, produces no state changes.

Sender Blacks Out/Silences Media

- Javascript sets “enabled=false” on sending MediaStreamTrack
- RTP keeps flowing, but encodes black frames or silence / comfort noise, according to media type
- No events are triggered on sending end
- Receiving party receives no events, produces no state changes

Receiver Blacks Out/Silences Media

- Javascript sets “enabled=false” on receiving MediaStreamTrack
- RTP keeps flowing, but playout (e.g. into <audio> or <video> tag) is suspended
  - Browser may elect to generate comfort noise
- No events are triggered on receiving end
- Sending party receives no events, produces no state changes, cannot detect this condition at all.

Sender Suspends RTP

- Javascript twiddles “new thing” on sending track
- “onnegotiationneeded” is triggered; media direction is updated in SDP

```
m=audio 25384 RTP/SAVPF 0 96
a=msid:ma ta
a=recvonly
```

- On receipt of this offer, the other side triggers “onmute.”

Receiver Suspends RTP

- Javascript twiddles "new thing" on receiving track
- "onnegotiationneeded" is triggered; media direction is updated in SDP

```
m=audio 25384 RTP/SAVPF 0 96
a=msid:ma ta
a=sendonly
```

- On receipt of *resulting answer*, this side triggers "onmute."
- Other side doesn't trigger any events

RTP Suspended Both Ways

- Javascript twiddles "new thing" on both sending and receiving track
- "onnegotiationneeded" is triggered; media direction is updated in SDP

```
m=audio 25384 RTP/SAVPF 0 96
a=msid:ma ta
a=inactive
```

- On receipt of this offer, the other side triggers "onmute" for stream we're sending
- On receipt of *resulting answer*, this side triggers "onmute" for stream we're receiving

Remove Sending Track

- Javascript calls stop() *or* removeStream on sending track
- "onnegotiationneeded" is triggered
  - MSID is removed
  - Media direction is updated in SDP

```
m=audio 25384 RTP/SAVPF 0 96
~~a=msid:ma ta~~
a=recvonly
```

- Other side calls onremovestream
- If the track is later re-added, it triggers an onaddstream
  - Does this cause issues? It's not clear that anything breaks.

Remove Received Stream

- Javascript calls stop() *or* removeStream on received stream
- "onnegotiationneeded" is triggered; media direction is updated in SDP

```
m=audio 25384 RTP/SAVPF 0 96
a=msid:ma ta
a=sendonly
```

- Other side doesn't trigger any events
- **Issue: SDP signaling is indistinguishable from suspended RTP**
  - Do we care?

Remove Both Streams

- Javascript calls stop() *or* removeStream on both streams
  - Although not necessarily at the same time
- "onnegotiationneeded" is triggered; port is set to 0

```
m=audio 0 RTP/SAVPF 0
```
~~a=msid:ma ta~~
~~a=sendrecv~~

- Other side calls "onremovestream" for stream it was receiving
- **Issue: Is there some event called on the stream it had been sending?**

Event Summary

Operation	Local Event	Remote Event
HW or OS Mute	onmute	-
HW or OS Unmute	onunmute	-
Disable sending MST	-	-
Enable sending MST	-	-
Disable receiving MST	-	-
Enable receiving MST	-	-
Suspend sending MST	onnegotiationneeded	onmute
Resume sending MST	onnegotiationneeded	onunmute
Suspend receiving MST	onnegotiationneeded, onmute	?
Resume receiving MST	onnegotiationneeded, onunmute	?
Stop or remove sending MST	onnegotiationneeded	onremovestream
Stop or remove receiving MST	onnegotiationneeded	?