WebRTC: IETF Standards Update
September 2016

Colin Perkins
The SIP framework is overly complex and rigid – hinders innovation

Embed standard media stack (RTP, ICE, etc.) into browsers, expose a standard control API rather than a standard signalling protocol – innovate above that API
WebRTC

WebRTC API

JavaScript Application

WebRTC API

Signalling

HTTP

Path Discovery

IPv4/IPv6

TCP

UDP

Media Transport

Data Channel

IETF

W3C

Colin Perkins – https://csperkins.org/ – Copyright © 2016 All Rights Reserved
WebRTC in IETF

JavaScript Application

WebRTC API

Media Transport  Data Channel  Signalling  Path Discovery

HTTP

UDP  TCP

IPv4/IPv6
WebRTC in IETF: Signalling

- JSEP and SDP exposed via API
- JSEP extracts SDP offer-answer out into reusable API component
  - SDP not easy to process with JavaScript
  - Extension and modification model poorly specified – simple applications are simple, but over-complicates other scenarios
  - An ORTC-like API might be cleaner?
- SDP BUNDLE extension groups
  - WebRTC traffic on single port:
    - RTP, Data Channel, STUN, DTLS
    - Complexity in identifying m= lines when bundled → msid, rid
    - Complexity in handling bundled attributes, signalling multiplexed flows
- Major issues resolved, but details remain open...

<table>
<thead>
<tr>
<th>Draft</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-ietf-rtcweb-use-cases-and-requirements</td>
<td>RFC 7478</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-overview</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-security</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-security-arch</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-jsep</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-sdp</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-constraints-registry</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-mmusic-sdp-bundle-negotiation</td>
<td>WG last call</td>
</tr>
<tr>
<td>draft-ietf-mmusic-msid</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-mmusic-sdp-mux-attributes</td>
<td>IESG review</td>
</tr>
<tr>
<td>draft-ietf-mmusic-rid</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-mmusic-sdp-simulcast</td>
<td>WG last call</td>
</tr>
<tr>
<td>draft-ietf-mmusic-mux-exclusive</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-mmusic-4572-update</td>
<td>WG last call</td>
</tr>
<tr>
<td>draft-ietf-mmusic-dtls-sdp</td>
<td>WG last call</td>
</tr>
</tbody>
</table>
WebRTC in IETF: Path Discovery

- STUN and TURN to discover NAT bindings and relay traffic
- Privacy concern around local IP address leak resolved
- Ongoing ICE revisions based on deployment experience with SIP

<table>
<thead>
<tr>
<th>Draft</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-ietf-rtcweb-transports</td>
<td>Approved</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-ip-handling</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-stun-consent-freshness</td>
<td>RFC 7675</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-mmusic-sctp-sdp</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-ice-dualstack-fairness</td>
<td>IESG review</td>
</tr>
<tr>
<td>draft-ietf-ice-rfc5245bis</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-ice-trickle</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-alpn</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-tsvwg-rtcweb-qos</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-mmusic-ice-sip-sdp</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-tram-stunbis</td>
<td>In progress</td>
</tr>
</tbody>
</table>
WebRTC in IETF: Data Channel

- Direct peer-to-peer data between browsers; no server involvement
- SCTP in secure UDP tunnel:
  - UDP tunnel ensures deployability but prevents SCTP multihoming
WebRTC in IETF: Media Transport

- **Audio and video codecs**
  - Opus, G.711, and DTMF digits required; AMR recommended
  - H.264 and VP8 required
  - Support for other codecs optional

- **Modern RTP and RTCP stack**
  - Bundled media on a single UDP port
  - Multiparty multimedia group conferencing – details around multiparty RTP sessions with different media types clarified
  - Secure RTP with DTLS-SRTP handshake
  - Detailed reception quality feedback, with NACK, retransmission, and FEC possible
  - Circuit breaker and congestion control for safe deployment on constrained paths

---

<table>
<thead>
<tr>
<th>Draft</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft-ietf-rtcweb-rtp-usage</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-audio</td>
<td>RFC 7874</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-audio-codecs-for-interop</td>
<td>RFC 7875</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-video</td>
<td>RFC 7742</td>
</tr>
<tr>
<td>draft-ietf-rtcweb-fec</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-avtcore-rtp-circuit-breakers</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-avtcore-rtp-multi-stream</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-avtcore-rtp-multi-stream-optimisation</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-avtcore-multi-media-rtp-session</td>
<td>With RFC Editor</td>
</tr>
<tr>
<td>draft-ietf-avtext-rid</td>
<td>IESG review</td>
</tr>
<tr>
<td>draft-ietf-avtext-sdes-hdr-ext</td>
<td>RFC 7941</td>
</tr>
<tr>
<td>draft-ietf-payload-flexible-fec-scheme</td>
<td>In progress</td>
</tr>
<tr>
<td>draft-ietf-avtextcore-rfc5761-update</td>
<td>IESG review</td>
</tr>
</tbody>
</table>
WebRTC in IETF: Status Summary

- Media transport and data channel essentially complete
- Path discovery and signalling protocols near completion – resolving details

Why are the standards taking so long?
- IPR around choice of mandatory to implement codec
- Decoupling SDP offer/answer from SIP to form JSEP, and complexity of resulting API interactions
- Complexity of bundled media: signalling and feature interaction; corner cases around use of RTP and RTCP with multiple simultaneous media types; demultiplexing and QoS with several protocols on a single port
- Revisions to STUN, TURN, and ICE
Challenges and Future Directions

• How might WebRTC evolve in future?
  • Quality of service support
  • Congestion control
  • ECN and ensuring low latency
  • Multicast and IPTV
  • Relation to new path layer protocols
Challenges and Future Directions

• How might WebRTC evolve in future?
  • Quality of service support ——— Differential QoS on a single UDP flow
  • Congestion control
  • ECN and ensuring low latency
  • Multicast and IPTV
  • Relation to new path layer protocols

Applications set different DSCP code points for the different media types and the data channel, and for different flow priorities
  • RFC 7657 and draft-ietf-tsvwg-rtcweb-qos-18

Do QoS-marked flows traverse the network?
  • Forwarding behaviour for some DSCP values is implementation defined – unclear what’s typical
  • DSCP field can be re-written or zeroed at network boundaries
  • Networks can discard packets with certain DSCP values due to security or business concerns

Unclear whether QoS support offers any benefits for interdomain use – or indeed, whether it hurts media quality
Challenges and Future Directions

- How might WebRTC evolve in future?
  - Quality of service support
  - Congestion control
  - ECN and ensuring low latency
  - Multicast and IPTV
  - Relation to new path layer protocols

RTP Circuit Breaker

New algorithm – does it work in the wide range of scenarios where WebRTC is deployed?

Congestion control for interactive media

Algorithms under development: Google Congestion Control, NADA, SCReAM

- Evaluation at an early stage – unclear any of these are stable in all desired scenarios, or with different types of cross traffic

Generic feedback mechanism under development

- Early work – unclear RTCP feedback can meet the timeliness requirements with reasonable overhead

Initial WebRTC deployments will have evolving congestion control – does this matter?
Challenges and Future Directions

- How might WebRTC evolve in future?
  - Quality of service support
  - Congestion control
  - **ECN and ensuring low latency**
  - Multicast and IPTV
  - Relation to new path layer protocols

**Explicit Congestion Notification**

Desire to move away from loss as congestion signal
- High latency → must fill queue to trigger loss
- Disruptive to user experience

Use of ECN with AQM allows smaller queues
- Requires support from network (CoDel, PIE, …)
- Requires support from circuit breaker
- Requires support from congestion controller
- Incrementally deployable

IETF L4S and TCP Prague experiments use ECT(1) with radically different congestion control: potentially much lower latency, but disruptive change
- Congestion response: $\frac{1}{ip} \rightarrow \frac{1}{p}$
- Not interoperable: dual queue AQM required

Response to ECN-CE mark should be less aggressive than response to packet loss
Challenges and Future Directions

• How might WebRTC evolve in future?
  • Quality of service support
  • Congestion control
  • ECN and ensuring low latency
  • Multicast and IPTV
  • Relation to new path layer protocols

Support for IP Multicast in WebRTC

Two approaches to video streaming:
• HTTP adaptive streaming – browser native format
• Multicast IPTV – designed for managed networks

WebRTC media stack is very similar to the multicast IPTV media stack:
• Missing MPEG-2 codec and payload format
• Missing source-specific multicast support
• Missing rapid channel change extensions
Incremental additions → not complex

Longer term: media interworking and interoperability?
• Different delivery modes need different encoding
• Hand-off between devices and delivery modes is difficult and non-scalable

Should WebRTC support multicast, so browsers can act as native IPTV clients?
• Better scaling for live streams
• Lower latency
Challenges and Future Directions

- How might WebRTC evolve in future?
  - Quality of service support
  - Congestion control
  - ECN and ensuring low latency
  - Multicast and IPTV
  - Relation to new path layer protocols

**Substrate protocols and the path layer**

Biggest challenge with WebRTC was making bundled media work
- Significant impact on RTP, congestion control, QoS
- Extremely complex signalling

New work in IETF: SPUD prototype and PLUS BoF
- Common UDP-based substrate layer on which new transport protocols can be run
- A secure path layer, with scope for edge-network communication

Can/should WebRTC migrate to run over this layer?
Challenges and Future Directions

• How might WebRTC evolve in future?
  • Quality of service support
  • Congestion control
  • ECN and ensuring low latency
  • Multicast and IPTV
  • Relation to new path layer protocols

• A transport-oriented viewpoint – what else?
  • Signalling APIs – ORTC vs. SDP-based approaches
  • Simplified JavaScript libraries
  • Monitoring and management tools and interfaces
Conclusions

- WebRTC provide a good baseline – a flexible, evolvable, framework
- Core IETF standards essentially done
- Clear path to evolve the network with lower latency, more adaptive media

Interesting challenges remain, but WebRTC is ready for deployment