Edge ORTC/WebRTC Q&A

Angelina Gambo
Bernard Aboba
May 18, 2017

Support

Interpreting the Edge Platform Status website

Q: How do I interpret the Edge Platform Status website (e.g. meaning of “in preview”, “in development”, “under consideration”)

A:

1. Under Consideration means on backlog for implementation in a future release. Priorities have the following explanations:
    a. Low — We are still evaluating this technology. There may be significant spec stabilization, foundational work, or additional community input required before we can begin development.
    b. Medium — Development is likely for a future release.
    c. High — We intend to begin development soon.
2. In Development means we have developers writing code actively on the feature.
3. In Preview means available in latest Windows Insider build.

Note that you can vote for features that are “under consideration”!

If you have additional questions on our web platform, you can first check out our FAQ page. Please send us feedback if you could not find answers to your questions there.

Getting the developer previews

Q: How can I obtain Windows Preview Builds?
A: Go to https://insider.windows.com/ and sign up for Window Insider builds. You can also try Virtual Machines. To develop applications requiring features under development, you will need to download recent Windows Insider Preview builds, as noted here.

Bug reporting

Q: How do I submit bugs?
A: Microsoft Edge is an evergreen browser and we will continue to evolve both the web platform and the user interface with regular updates. If you are a web developer or designer and have feature requests for the web platform, we encourage you to share your feedback at the Microsoft Edge Platform UserVoice. If you are a web developer or designer and have a bug report, please file an issue at the Microsoft Edge Platform Issues site.

To send feedback on user experience, broken or malicious sites, use the "Send Feedback" menu item in Microsoft Edge.

Stream & Port handling

Multiple Streams & Unified Plan

Q: Does Edge support multiple streams in ORTC? Will you support it in WebRTC 1.0?
A: Edge ORTC enables multiple audio and video streams to be carried over a single *DtlsTransport*. In the Windows 10 Spring 2018 Creator's Update, the WebRTC 1.0 API supports multiple audio streams within PeerConnection.

Multiplexing

Q: Does Edge support A/V multiplexing? RTP/RTCP multiplexing?
A: Edge supports both A/V multiplexing and RTP/RTCP multiplexing in the ORTC and WebRTC 1.0 native APIs. In ORTC, RTP/RTCP multiplexing is required by the *DtlsTransport*, but RTP/RTCP non-mux is supported for use with the *SrtpSdesTransport*.

DataChannel

Q: What is the status of the WebRTC data channel?
A: The WebRTC data channel is currently listed as under consideration with a low priority on the Edge Platform Status site.

As part of a potential WebRTC data channel work item, the interaction of data channel congestion control with A/V congestion control needs to be considered. The current data channel implementations in Skype and Skype for Business were based on unified congestion control handling for both A/V and dat**a**, taking into account both delay and packet loss, which ensures that data transfer does not adversely impact A/V quality. However, in WebRTC, the data channel and A/V congestion control algorithms operate independently, and can compete with each other. For example, if the data channel utilizes loss-based congestion control, queues can build, causing a delay-based A/V congestion control scheme to back off, yielding more bandwidth to the data channel and a video quality death spiral. To understand the right congestion control algorithms to use for A/V as well as data channel congestion control, more research is needed.

Q: What is the status of QUIC data exchange in WebRTC?
A: The ORTC API specification has recently added support for peer-to-peer QUIC data exchange. However the API is still preliminary so QUIC data exchange should also be considered to be under consideration with a low priority.

Media Recorder plans

Q: What formats are you considering support for in Media Recorder?
Media Recorder is "under consideration" with low priority. For video it is likely that H.264 (MP4) will be the preferred format due to the broad support on hardware platforms and devices.

Security & Encryption

No HTTPS requirement

Q: Is Edge considering following Chrome's HTTPS requirement for getUserMedia()?
A: We currently allow both HTTP and HTTPS origins to access capture devices through getUserMedia(), although we apply more restricted user permission rules in the HTTP case. Meanwhile, we are actively tracking usage of the API. When the majority of websites have migrated to HTTPS when using getUserMedia(), we will consider making a change.

"IP Leakage" prevention

Q: How do you handle IP address privacy?
A: We provided an option for Microsoft Edge users to not expose local IP addresses. You can manually turn on the option in the about:flags. For enterprise customers, you can enable the Group Policy mapped to the same functionality.

Screen sharing security

Q: Are you considering using extensions to address some of the security issues in Screen Sharing?
A: Screen Capture is supported within the Windows 10 Spring 2018 Creator Update. The user experience and permission control are seamlessly integrated with the Windows platform, so Edge browser users don't have to take extra steps in enabling the feature.

ECDSA Support

Q: What is the status of elliptic curve cipher suites? Are there potential interoperability issues a developer needs to be aware of?
A: Support for DTLS 1.2 and elliptic curve cipher suites was first included in the Windows 10 Anniversary Update. This means that Edge can now validate elliptic curve certificates, but will still generate RSA certificates by default. Since the certificate management API is not supported, it is not yet possible for applications to generate elliptic curve certificates.

The Edge team is currently examining backward compatibility issues understand how to best support developers. Potential issues include legacy mobile applications based on WebRTC that only support DTLS 1.0 and RSA ciphersuites and applications that do not utilize the certificate management API.

Device Fingerprinting

Q: How do you prevent device fingerprinting?
A: We currently don't support persistent user permission yet for media capture devices. So, based on the W3C spec, we don't persist the *deviceId* across browsing sessions. We also don't reveal the device "label" when the webpage enumerates the capture devices unless the user has granted permission to the specific device type.

WebRTC 1.0 & Interoperability

What's in 1.0?

Q: Is the WebRTC 1.0 API supported?
A: Yes. Edge RTC platform provides two ways for developers to utilize the WebRTC 1.0 API: adapter.js running over ORTC, and a native WebRTC 1.0 API implementation.,

adapter.js provides the ability to create WebRTC 1.0 audio and video applications that run on all WebRTC browsers. The Edge Test Drive site as well as webrtc examples on github provide examples of applications running on multiple WebRTC browsers, using *adapter.js* (or an equivalent adaptation layer). Recent versions of *adapter.js* support addTrack/ontrack/removeTrack and there is work in progress to support addTransceiver.

Since developers needing features not initially supported in *adapter.js* requested native WebRTC 1.0 support, we provide native support for part of the WebRTC 1.0 API (addStream/removeStream as explained in Jan-Ivar's blog). However Edge does not yet natively support addTransceiver.

Based on feedback on our initial native WebRTC 1.0 API support, we will evaluate plans for enhancements.

Adapter.js

Q: Can I develop WebRTC 1.0 applications supporting interoperable audio and video using Edge with *adapter.js*?
A: Yes. The Edge RTC platform supports both H.264/AVC and VP8 by default. The latest version of adapter.js supports interop with VP8, H.264/AVC as well as RTX, enabling development of interoperable A/V applications running on Edge, Chrome and Firefox. For known problems check the Issues list.

Q: What is the status of Edge automated testing?
A: We have quite extensive coverage in our internal test automation. Currently the WPT tests have issues on Edge due to problems with webdriver, but once they become more mature, we hope to include them within our automated testing. Given the complexity of more advanced scenarios such as multi-stream, simulcast and scalable video coding, the KITE test framework is also of interest.

Q: Are there plans to help maintain adapter.js support now that Edge supports the WebRTC 1.0 API natively?
A: We have worked closely with Philipp Hancke to provide technical support for adapter.js on Edge ORTC, including support for H.264/AVC, VP8 and RTX interoperability. Since our native WebRTC 1.0 implementation currently lacks features supported in adapter.js,

developers creating interoperable applications based on the latest WebRTC 1.0 API may want to consider adapter.js , particularly if their roadmap includes support for the WebRTC 1.0 object model or multi-stream video.

Codec and Robustness Support

ORTC & 1.0 Parity
Q: Are all supported codecs available within both the ORTC and WebRTC 1.0 APIs?
A: Support for H.264/AVC and VP8 is available both within the WebRTC 1.0 and ORTC APIs, as well as support for G.711, Opus, Comfort Noise and DTMF. However, support for the proprietary H.264UC and SILK codecs as well as extended telemetry is only available via the ORTC API.

Audio

Opus Support
Q: Why is the "Opus audio format" listed in about:flags ? Isn't the Opus codec always turned on by default?
A: Opus is currently supported within the Edge RTC platform and is always available. The "Opus audio format" check box in about:flags refers to playback of Opus in streaming scenarios.

FEC
Q: Does Edge support Opus internal FEC? What about DTX?
A: Opus internal FEC and discontinuous operation is currently not supported. However, if Comfort Noise (CN) is added to the list of audio codecs, voice activity detection can be supported with Opus.

RED
Q: Do you support use of RED with Opus?
A: It is possible to use RED with Opus. However, we have not tested interoperability with other browsers.

Audio codecs
Q: What audio codecs does Edge support?
A: Currently Edge supports G.711, Opus, SILK, DTMF and CN audio codecs, but not G.729. G.722 is only supported on the XBOX platform.

Video

H.264/AVC
Q: What is the status of H.264/AVC support?
A: H.264/AVC is on by default. Here is what is supported:
- ORTC and WebRTC 1.0 APIs
- packetization-mode = 1, as required by RFC 7742

- Constrained Baseline Profile with Level up to 4.2 (i.e. with profile-level-id=42c02a)
- Support for Google REMB and the Absolute send time header extension
- Picture Loss Indication (PLI), defined in RFC 4585 Section 6.3.1
- Generic NACK (in preview), defined in RFC 4585 Section 6.2.1
- Retransmission (in preview), defined in RFC 4588.

VP8

Q: What is the status of VP8?
A: VP8 is now enabled by default, accessible via both ORTC and WebRTC 1.0 APIs.

VP9

Q: What is the status of VP9 in Edge? What does "playback" mean?
A: "VP9 video format" is automatically supported as noted in about:flags "Video Playback" means that Edge can play back VP9 video with the HTML5 video tag, but VP9 is not currently supported within the Edge RTC platform.

SVC & Simulcast

Q: I heard that Edge supports scalable video coding and simulcast. Is this general (e.g. any codec) or is it codec-specific?
A: Edge ORTC supports both encoding and decoding of simulcast and scalable video coding only within the H.264UC video codec, which implements H.264/SVC (RFC 6190) with Skype extensions. Support for H.264UC is not provided within the WebRTC 1.0 native API.

Q: Do the H.264/AVC or VP8 codecs support simulcast?
A: Neither the ORTC nor WebRTC 1.0 APIs enable use of simulcast with H.264/AVC or VP8.

Q: Does Edge support encoding or decoding of temporal scalability within VP8?
A: The VP8 specification requires that the decoder be able to decode any valid VP8 encoding. Therefore, the Edge RTC VP8 decoder can decode temporal scalability (with either the WebRTC 1.0 or ORTC APIs). However, we do not support encoding of temporal scalability with either API.

RTX, FEC, ULPFEC, RED

Q: Does Edge support retransmission? flexible FEC? Ulpfec?
A: Retransmission (RFC 4588) is supported within the WebRTC 1.0 and ORTC APIs, and is also supported in the latest release of adapter.js. Support for RED + ulpfec is under consideration with medium probability. Currently support for flexible FEC is not under consideration.

PLI, NACK, REMB

Q: What feedback messages does Edge RTC support?
A: PLI, REMB and Generic NACK are supported.

Video Interop

Q: What additional work has been done relating to video interoperability?
A: In addition to support for the H.264/AVC and VP8 video codecs, Edge supports the PLI and Generic NACK feedback messages (RTP/SAVPF), congestion control (REMB) and retransmission (RTX). These features are available for both ORTC and WebRTC 1.0 native APIs.

ICE & TURN

Trickle-ICE Support

Q: Are there plans to support Trickle ICE?
A: EdgeRTC platform supports ICE (RFC 5245). Trickle ICE is under consideration with priority medium.

TURN Variations

Q: Does Edge support TURN? TURN/TCP? TURN/TLS?
A: Edge RTC platform supports TURN (RFC 5766), TURN IPv6 (RFC 6156) and ICE TCP (RFC 6544). Allocation of TCP candidates within TURN is not supported (RFC 6062). Support for TURN over TCP/TLS is under consideration with medium priority.

Developing with Edge

Open sourcing the Edge Engine?

Q: Will the Edge RTC engine be open sourced?
The Edge Javascript Engine (Chakra) is now open-source and an open-source implementation of ORTC is available (ORTC Lib). Also, an open-source H.264/SVC encoder/decoder is available from Polycom, and the source code for the Skype SILK codec has been published. However, the Edge RTC stack is based on the Skype media library, and is not available in open source.

Release Notes

Q: Do you plan to have a ORTC/WebRTC release notes for every release like Firefox and Chrome have?
A: ORTC API documentation for each Edge release is available here. We also maintain current versions of this FAQ and the Edge RTC platform roadmap there.

Browser-Debug tools

Q: Do you plan to add any RTC debugging tools like about:webrtc and chrome://webrtc-internals?
A: The Edge RTC platform supports the Statistics API as well as extensive session summary statistics within the msStats API. In addition, ORTC objects such as the *IceTransport* and *DtlsTransport* provide information on the state of individual transports. General plans for F12 are discussed here; currently this does not include RTC-specific debugging.

ORTC Lib includes support for Event Tracing for Windows (ETW), which is very helpful for advanced debugging of real-time communications applications. ETW is a kernel-level tracing

facility that lets you log kernel or application-defined events to a log file, which tools can analyze.

Demos

Q: Will the demos on https://developer.microsoft.com/en-us/microsoft-edge/testdrive/ be updated and maintained? Will there be more? If a developer has a good ORTC demo, how do they get it put up there?
A: Demos on the testdrive page will be updated based on our progress with Edge RTC. We currently have two audio RTC demos from developers outside of Microsoft, one from Twilio, and one from &Yet. Now that H.264/AVC and VP8 support is in preview, we are working on getting video demos as well. If you'd like to have your RTC video demo included on our testdrive page, please contact me directly (at bernarda@microsoft.com).

References

Q: What references do you recommend for developers looking creating applications on Edge?
A: A tutorial is under development covering both WebRTC and ORTC API development using Edge.

Windows Mobile Support

Q: Can you clarify your support for Edge with ORTC on XBOX or various mobile devices? WebRTC 1.0 plans for the same?
A: Edge currently supports the ORTC and WebRTC 1.0 APIs on XBOX devices. There are no plans for Edge support on Windows Phone.

Skype for Web on ORTC

Q: Does Skype for Web use ORTC in Edge?
A: Yes, the plugin-free Skype for Web preview on Edge is based on the ORTC API. To provide A/V quality comparable to existing Skype clients, it takes advantage of the SILK codec, as well as the H.264UC video codec and advanced session statistics.

Q: Why does the new Skype for Web require non-Edge browser users to have the latest version?
A: Edge browser users may be prompted to install a plugin if the remote participant doesn't use the latest version of the Skype client on desktop or mobile. The plugin allows use of legacy protocols compatible with older versions of the Skype client. However, we'd strongly encourage Skype users to update their Skype apps to get the best experience based on the latest Skype client.