Contribution to RTC-WEB Workshop

by Jan Linden and Niklas Enbom, Global IP Solutions

1. Introduction

In this contribution we will address some of the questions asked in the invitation to the RTC-WEB workshop. In particular we will address, from a media processing perspective, codec choice and other issues such as bandwidth management and interfaces to soundcard and web cameras.

2. Choice of Codecs

In the invitation the following question is asked:

> **What are the right set of codecs for doing real-time interactive voice and video in the browser?**

The choice of codec is dependent on several factors, the main ones being:

- Basic quality (in the case of audio one important aspect is sample rate support)
- Bit rate
- Robustness against network degradation
- Complexity (footprint and CPU utilization)
- Latency
- Cost
- Interoperability

Naturally, each codec has been designed as a trade-off between these conflicting requirements. Hence, one codec is not likely to fulfill all needs. We will in the following propose the codecs we believe offer the best trade-off for the RTC-WEB activity. Codecs that are branded free are naturally favored heavily.
It is our opinion that for the sake of interoperability a small set of codecs need to be mandated. This set should be big enough to offer at least one codec option for virtually any interoperability scenario but small enough to facilitate wide platform support. Too many mandated codecs will inevitably slow down proliferation to mobile devices and impact server scalability. We have divided the need for interoperability into two distinct areas. One is interoperability between RTC-WEB enabled solutions and the other is interoperability with existing clients/services and the PSTN. In the following tables the most obvious codec choices are compared.

Voice Codec	**Interop opportunity**	**Technical Advantage**	**License**	**Remarks**
G.711	All PSTN equipment	Low	Considered free	Extremely low CPU

	and almost all other clients		(patents expired)	reqs. High bit rate.
G.722	All wideband speech PSTN equipment.	Low	Considered free (patents expired)	Baseline in most wideband offerings
G.729	Most commercial SIP endpoints	Medium	Commercial	A low bit rate option for PSTN interop.
iLBC	Most free VoIP clients and many commercial SIP enpoints	Medium	Free	Current license not optimal for open source
Speex	Adobe Flash Player. Some open SIP clients	Medium	Free	
IETF codec	TBD	High	Free	Current license not optimal for open source
iSAC	Many commercial VoIP clients	High	Commercial	

Video Codec	**Interop opportunity**	**Technical Advantage**	**License**	**Remarks**
H.263	Almost all SIP video clients	Medium	Considered free	Mostly used for interop, quality subpar
H.264	New SIP video clients. Teleconferencing equipment. Adobe Flash can decode.	High	Commercial	
WebM	TBD	High	Free	Needs profile for realtime applications.

Based on this the following are our suggestions for voice and video codecs that offer good trade-offs between the requirements mentioned above:

1. Audio codecs that facilitate interoperability between browser clients
    - IETF Codec (still in progress)

    - Preferably one more open codec (Speex, iLBC, iSAC?)
2. Video codec that facilitates interoperability between browser clients
    - WebM
    - Any other open codecs available?
3. Audio codecs that facilitate interoperability with other clients and services including the PSTN
    - G.711
    - G.722
    - iLBC
    - G.729 would be good for interoperability but is far from free and is probably not necessary
4. Video codecs that facilitate interoperability with other clients and services
    - H.263
    - H.264 would be good for interoperability but is far from free and is probably not necessary

Clearly there will be a need to support other codecs than the mandated ones. In order to facilitate this it is important to offer an easy process to add additional codecs, and that such a codec can be negotiated between two independent implementations.

3. How to Achieve High Quality and Interoperability yet Leave Room for Innovation

The question posed in the invitation is:

> **How do we enable application providers to innovate in areas like bandwidth estimation and rate control (an area which has purposefully been left to implementers to innovate on), while still enabling interoperability?**

Only the codec affects interoperability and hence it is necessary to agree on a set of supported codecs. However, just because two endpoints use the same codec doesn't mean that they offer the same performance. All other media processing such as echo cancellation, noise suppression, video pre-processing, jitter buffers, packet loss concealment and bandwidth management also impact the end user experience. In order for this activity to be successful it is important that reasonably good media processing is available to all implementers. However, there is still room for differentiation in this area.

3.1 What Media Processing does not need to be covered by this effort?

- AEC, AGC, Noise Reduction
- Jitter buffer, packet loss concealment
- Other pre- and post-processing techniques (for example video quality enhancements, gaze correction etc).

3.2 Bandwidth estimation and rate control

It's our opinion that as much as possible in this area should be kept codec independent. Based on that we see a need for a framework outside the codec that:

- Provides means to signal the max bit rate the client wants to receive
- Provides enough statistics the makes it possible to estimate the available bandwidth:
  - Immediate feedback on packet loss
  - Fast and accurate RTT estimation

As already stated the actual algorithm used to estimate the available bandwidth should be kept outside any standardization effort. Looking at existing standards for RTP the means exist but are locked to certain codecs or also includes the actual bandwidth estimation algorithm:

- RFC 5104 (Codec Control Messages in the RTP Audio-Visual Profil): TMMBR, TMMBN
- RFC 3448 (TCP Friendly Rate Control): feedback packets

This means that there might exist a need for another standards effort on the bandwidth management side.

3.3 Sound card and camera handling

The media processing software needs access to the soundcard, camera, and display for capture and rendering of audio and video. From this perspective we see little need for direct access to multimedia devices at the highest HTML5 level. As much as possible of the sound card and camera behavior should be controlled through the media stream settings, such as:

- Video resolution and frame rate (video stream settings)
- Audio sampling frequency (voice codec used)

Some device specific setting needs to be exposed to the end user, such as :

- Device enumerations and selection
- Volume and muting settings

For security reasons it's advantageous to not let a web application control all such settings and modify the settings through the browser settings only - to avoid for example a web application to unmute the microphone without notifying the end user. A drawback of this approach is that this might limit the application functionality, for example using multiple cameras in a creative way. Also, we understand the usability issues around this and expect a discussion on the topic.