Network Working Group	H. T. . Alvestrand
Internet-Draft	Google
Intended status: Standards Track	June 20, 2011
Expires: December 22, 2011	

Overview: Real Time Protocols for Brower-based Applications
draft-alvestrand-rtcweb-overview-01

Abstract

This document gives an overview and context of a protocol suite intended for use with real-time applications that can be deployed in browsers - "real time communication on the Web".
It intends to serve as a starting and coordination point to make sure all the parts that are needed to achieve this goal are findable, and that the parts that belong in the Internet protocol suite are fully specified and on the right publication track.
This work is an attempt to synthesize the input of many people, but makes no claims to fully represent the views of any of them. All parts of the document should be regarded as open for discussion, unless the RTCWEB chairs have declared consensus on an item.
This document is a candidate to become a work item of the RTCWEB working group.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 *[RFC2119]*.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The Internet was, from very early in its lifetime, considered a possible vehicle for the deployment of real-time, interactive applications - with the most easily imaginable being audio conversations (aka "Internet telephony") and videoconferencing.
The first attempts to build this were dependent on special networks, special hardware and custom-built software, often at very high prices or at low quality, placing great demands on the infrastructure.
As the available bandwidth has increased, and as processors and other hardware has become ever faster, the barriers to participation have decreased, and it is possible to deliver a satisfactory experience on commonly available computing hardware.
Still, there are a number of barriers to the ability to communicate universally - one of these is that there is, as of yet, no single set of communication protocols that all agree should be made available for communication; another is the sheer lack of universal identification systems (such as is served by telephone numbers or email addresses in other communications systems).
Development of The Universal Solution has proved hard, however, for all the usual reasons. This memo aims to take a more building-block-oriented approach, and try to find consensus on a set of substrate components that we think will be useful in any real-time communications systems.
The last few years have also seen a new platform rise for deployment of services: The browser-embedded application, or "Web application". It turns out that as long as the browser platform has the necessary interfaces, it is possible to deliver almost any kind of service on it.
Traditionally, these interfaces have been delivered by plugins, which had to be downloaded and installed separately from the browser; in the development of HTML5, much promise is seen by the possibility of making those interfaces available in a standardized way within the browser.
This memo specifies a set of building blocks that can be made accessible and controllable through a Javascript API interface in a browser, and which together form a necessary and sufficient set of functions to allow the use of interactive audio and video in applications that communicate directly between browsers across the Internet.

Other efforts, for instance the W3C WebRTC, Web Applications and Device API working groups, focus on making standardized APIs and interfaces available, within or alongside the HTML5 effort, for those functions; this memo concentrates on specifying the protocols and subprotocols that are needed to specify the interactions that happen across the network.

2. Principles and Terminology

2.1. Goals of this document

The goal of the RTCWEB protocol specification is to specify a set of protocols that, if all are implemented, will allow the implementation to communicate with another implementation using audio, video and auxiliary data sent along the most direct possible path between the participants.
This document is intended to serve as the roadmap to the RTCWEB specifications. It defines terms used by other pieces of specification, lists references to other specifications that don't need further elaboration in the RTCWEB context, and gives pointers to other documents that form part of the RTCWEB suite.
By reading this document and the documents it refers to, it should be possible to have all information needed to implement an RTCWEB compatible implementation.

2.2. Relationship between API and protocol

The total RTCWEB/WEBRTC effort consists of two pieces:

*A protocol specification, done in the IETF

*A Javascript API specification, done in the W3C

Together, these two specifications aim to provide an environment where Javascript embedded in any page, viewed in any compatible browser, when suitably authorized by its user, is able to set up communication using audio, video and auxiliary data, where the browser environment does not constrain the types of application in which this functionality can be used.
The protocol specification does not assume that all implementations implement this API; it is not intended to be possible by observing the bits on the wire whether they come from a browser or from another device implementing this specification.
The goal of cooperation between the protocol specification and the API specification is that for all options and features of the protocol specification, it should be clear which API calls to make to exercise that option or feature; similarly, for any sequence of API calls, it should be clear which protocol options and features will be invoked. Both subject to constraints of the implementation, of course.

2.3. On interoperability and innovation

The "Mission statement of the IETF" [RFC3935] states that "The benefit of a standard to the Internet is in interoperability - that multiple products implementing a standard are able to work together in order to deliver valuable functions to the Internet's users."
Communication on the Internet frequently occurs in two phases:

- Two parties communicate, through some mechanism, what functionality they both are able to support

- They use that shared communicative functionality to communicate, or, failing to find anything in common, give up on communication.

There are often many choices that can be made for communicative functionality; the history of the Internet is rife with the proposal, standardization, implementation, and success or failure of many types of options, in all sorts of protocols.
The goal of having a mandatory to implement function set is to prevent negotiation failure, not to preempt or prevent negotiation.
The presence of a mandatory to implement function set serves as a strong changer of the marketplace of deployment - in that it gives a guarantee that, as long as you conform to a specification, and the other party is willing to accept communication at the base level of that specification, you can communicate successfully.
The alternative - that of having no mandatory to implement - does not mean that you cannot communicate, it merely means that in order to be part of the communications partnership, you have to implement the standard "and then some" - that "and then some" usually being called a profile of some sort; in the version most antithetical to the Internet ethos, that "and then some" consists of having to use a specific vendor's product only.

2.4. Terminology

The following terms are used in this document, and as far as possible across the documents specifying the RTCWEB suite, in the specific meanings given here. Other terms are used in their commonly used meaning.
The list is in alphabetical order.

API: Application Programming Interface - a specification of a set of calls and events, usually tied to a programming language or an abstract formal specification such as WebIDL, with its defined semantics.

Interactive: Communication between multiple parties, where the expectation is that an action from one party can cause a reaction by another party, and the reaction can be observed by the first party,

with the total time required for the action/reaction/observation is on the order of no more than hundreds of milliseconds.

Media: Audio and video content. Not to be confused with "transmission media" such as wires.

Protocol: A specification of a set of data units, their representation, and rules for their transmission, with their defined semantics. A protocol is usually thought of as going between systems.

Real-time media: Media where generation of content and display of content are intended to occur closely together in time (on the order of no more than hundreds of milliseconds).

NOTE: Where common definitions exist for these terms, those definitions should be used to the greatest extent possible.
TODO: Extend this list with other terms that might prove slippery.

3. Functionality groups

The functionality groups that are needed can be specified, more or less from the bottom up, as:

*Data transport: TCP, UDP and the means to securely set up connections between entities, as well as the functions for deciding when to send data: Congestion management, bandwidth estimation and so on.

*Data framing: RTP and other data formats that serve as containers, and their functions for data confidentiality and integrity.

*Data formats: Codec specifications, format specifications and functionality specifications for the data passed between systems. Audio and video codecs, as well as formats for data and document sharing, belong in this category. In order to make use of data formats, a way to describe them, a session description, is needed.

*Connection management: Setting up connections, agreeing on data formats, changing data formats during the duration of a call; SIP and Jingle/XMPP belong in this category.

*Presentation and control: What needs to happen in order to ensure that interactions behave in a non-surprising manner. This can include floor control, screen layout, voice activated image switching and other such functions - where part of the system require the cooperation between parties. Cisco/Tandberg's TIP was one attempt at specifying this functionality.

*Local system support functions: These are things that need not be specified uniformly, because each participant may choose to do these in a way of the participant's choosing, without affecting the bits on the wire in a way that others have to be cognizant of. Examples in this category include echo cancellation (some forms of it), local authentication and authorization mechanisms, OS access control and the ability to do local recording of conversations.

Within each functionality group, it is important to preserve both freedom to innovate and the ability for global communication. Freedom to innovate is helped by doing the specification in terms of interfaces, not implementation; any implementation able to communicate according to the interfaces is a valid implementation. Ability to communicate globally is helped both by having core specifications be unencumbered by IPR issues and by having the formats and protocols be fully enough specified to allow for independent implementation.
One can think of the three first groups as forming a "media transport infrastructure", and of the three last groups as forming a "media service". In many contexts, it makes sense to use a common specification for the media transport infrastructure, which can be embedded in browsers and accessed using standard interfaces, and "let a thousand flowers bloom" in the "media service" layer; to achieve interoperable services, however, at least the first five of the six groups need to be specified.

4. Data transport

Data transport refers to the sending and receiving of data over the network interfaces, the choice of network-layer addresses at each end of the communication, and the interaction with any intermediate entities that handle the data, but do not modify it (such as TURN relays).
It includes necessary functions for congestion control: When not to send data.
The data transport protocols used by RTCWEB are described in <WORKING GROUP DRAFT "TRANSPORTS">.
The interactions with intermediate boxes, such as firewalls, relays and NAT boxes, is described in <WORKING GROUP DRAFT "PEER TO PEER CONNECTIVITY">.

5. Data framing and securing

SRTP [RFC3550] is used for transport of all real-time media.
The detailed considerations for usage of functions from RTP and SRTP, as well as for non-media real-time data, are given in <WORKING GROUP DRAFT "MEDIA TRANSPORTS">.

6. Data formats

The intent of this specification is to allow each communications event to use the data formats that are best suited for that particular instance, where a format is supported by both sides of the connection. However, a minimum standard is greatly helpful in order to ensure that communication can be achieved. This document specifies a minimum baseline that will be supported by all implementations of this specification, and leaves further codecs to be included at the will of the implementor.
The mandatory to implement codecs, as well as any profiling requirements for both mandatory and optional codecs, is described in <WORKING GROUP DRAFT "MEDIA PROCESSING">.

7. Connection management

The methods, mechanisms and requirements for setting up, negotiating and tearing down connections is a large subject, and one where it is desirable to have both interoperability and freedom to innovate.
The particular choices made for RTCWEB are described in <WORKING GROUP DRAFT "SIGNALING AND NEGOTIATION">.

8. Presentation and control

The most important part of control is the user's control over the browser's interaction with input/output devices and communications channels. It is important that the user have some way of figuring out where his audio, video or texting is being sent, for what purported reason, and what guarantees are made by the parties that form part of this control channel. This is largely a local function between the browser, the underlying operating system and the user interface; this is being worked on as part of the W3C API effort. <INSERT REFERENCE TO W3C API SPEC WHEN AVAILABLE>

9. Local system support functions

These are characterized by the fact that the quality of these functions strongly influences the user experience, but the exact algorithm does not need coordination. In some cases (for instance echo cancellation, as described below), the overall system definition may need to specify that the overall system needs to have some characteristics for which these facilities are useful, without requiring them to be implemented a certain way.
Local functions include echo cancellation, volume control, camera management including focus, zoom, pan/tilt controls (if available), and more.
Certain parts of the system SHOULD conform to certain properties, for instance:

*Echo cancellation should be good enough that feedback (defined as a rising volume of sound with no local sound input) does not occur.

*Privacy concerns must be satisfied; for instance, if remote control of camera is offered, the APIs should be available to let the local participant to figure out who's controlling the camera, and possibly decide to revoke the permission for camera usage.

*Automatic gain control, if present, should normalize a speaking voice into <whatever dB metrics makes sense here - most important that we have one only>

The requirements on RTCWEB systems in this category are found in <WORKING GROUP DRAFT "MEDIA PROCESSING">.

10. IANA Considerations

This document makes no request of IANA.
Note to RFC Editor: this section may be removed on publication as an RFC.

11. Security Considerations

Security of the web-enabled real time communications comes in several pieces:

*Security of the components: The browsers, and other servers involved. The most target-rich environment here is probably the browser; the aim here should be that the introduction of these components introduces no additional vulnerability.

*Security of the communication channels: It should be easy for a participant to reassure himself of the security of his communication - by verifying the crypto parameters of the links he himself participates in, and to get reassurances from the other parties to the communication that they promise that appropriate measures are taken.

*Security of the partners' identity: verifying that the participants are who they say they are (when positive identification is appropriate), or that their identity cannot be uncovered (when anonymity is a goal of the application).

The security analysis, and the requirements derived from that analysis, is contained in <WORKING GROUP DRAFT "SECURITY">.

12. Acknowledgements

The number of people who have taken part in the discussions surrooounding this draft are too numerous to list, or even to identify. The ones below have made special, identifiable contributions; this does not mean that others' contributions are less important.
Thanks to Cary Bran, Cullen Jennings, Colin Perkins, Magnus Westerlund and Joerg Ott, who offered techincal contributions on various versions of the draft.
Thanks to Justin Uberti and Simon Leinen for document review.

13. References

13.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC1890]	Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 1890, January 1996.
[RFC3550]	Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
[RFC3711]	Baugher, M., McGrew, D., Naslund, M., Carrara, E. and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
[RFC4585]	Ott, J., Wenger, S., Sato, N., Burmeister, C. and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.
[RFC5124]	Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.
[I-D.ietf-codec-opus]	Valin, J, Vos, K and T Terriberry, "Definition of the Opus Audio Codec", Internet-Draft draft-ietf-codec-opus-10, October 2011.
[I-D.ietf-hybi-thewebsocketprotocol]	Fette, I and A Melnikov, "The WebSocket protocol", Internet-Draft draft-ietf-hybi-thewebsocketprotocol-17, September 2011.
[I-D.alvestrand-dispatch-rtcweb-datagram]	Alvestrand, H, "A Datagram Transport for the RTC-Web profile", Internet-Draft

	draft-alvestrand-dispatch-rtcweb-datagram-01, February 2011.
[I-D.westin-payload-vp8]	Westin, P and H Lundin, "Proposal for the IETF on "RTP Payload Format for VP8 Video"", Internet-Draft draft-westin-payload-vp8-02, March 2011.

13.2. Informative References

[I-D.ietf-avt-rtp-svc]	Wenger, S, Wang, Y, Schierl, T and A Eleftheriadis, "RTP Payload Format for Scalable Video Coding", Internet-Draft draft-ietf-avt-rtp-svc-27, February 2011.
[RFC3935]	Alvestrand, H., "A Mission Statement for the IETF", BCP 95, RFC 3935, October 2004.

Appendix A. Change log

This section may be deleted by the RFC Editor when preparing for publication.

Appendix A.1. Changes from draft-alvestrand-dispatch-rtcweb-datagram-00 to -01

Added section "On interoperability and innovation"
Added data confidentiality and integrity to the "data framing" layer
Added congestion management requirements in the "data transport" layer section
Changed need for non-media data from "question: do we need this?" to "Open issue: How do we do this?"
Strengthened disclaimer that listed codecs are placeholders, not decisions.
More details on why the "local system support functions" section is there.

Appendix A.2. Changes from draft-alvestrand-dispatch-01 to draft-alvestrand-rtcweb-overview-00

Added section on "Relationship between API and protocol"
Added terminology section
Mentioned congestion management as part of the "data transport" layer in the layer list

Appendix A.3. Changes from draft-alvestrand-rtcweb-00 to -01

Removed most technical content, and replaced with pointers to drafts as requested and identified by the RTCWEB WG chairs.
Added content to acknowledgements section.
Added change log.

Spell-checked document.

Author's Address

Harald T. Alvestrand Alvestrand Google Kungsbron 2 Stockholm, 11122 Sweden EMail: harald@alvestrand.no