QuARTCS: A Tool Enabling End-to-Any Speech Quality Assessment of WebRTC-Based Calls

Martin Meszaros[1,2(✉)], Franziska Trojahn[1,2(✉)], Michael Maruschke[2], and Oliver Jokisch[2]

[1] immmr GmbH, Winterfeldtstraße 21, 10781 Berlin, Germany
{martin.meszaros,franziska.trojahn}@immmr.com
www.immmr.com
[2] Institute of Communications Engineering,
Leipzig University of Telecommunications (HfTL),
Gustav-Freytag-Straße 43-45, 04277 Leipzig, Germany
www.hft-leipzig.de

Abstract. Recently, the use of Web Real-Time Communication (WebRTC) technology in communication applications has been increasing significantly. The users of IP-based telephony require excellent audio quality. However, in WebRTC-based audio calls the audio assessment is challenging due to the specific functioning principles of WebRTC, such as security requirements, diversity of the endpoints and varying client implementations.

In this article, we illustrate the challenges in established methods of audio quality assessment with regard to WebRTC and discuss necessary modifications in the measurement technique. We present Quality Analyzer for Real Time Communication Scenarios (QuARTCS) as a novel method to overcome the measurement shortcomings and demonstrate the basic functioning by preliminary call samples.

Keywords: WebRTC · Audio quality assessment · Opus codec · VoIP

1 Introduction

The popularity of Internet-based communication is steadily increasing. The demands in regard to quality, availability and type of service have adapted to the changes in daily lifestyle: Multiple services have to be available on all devices, from any place and at any time. While voice-based telecommunication is no longer limited to telephones but also available on computers and tablets, it still needs to be easy-to-use for naive users and to provide interoperability with legacy solutions such as the Public Switched Telephone Network (PSTN).

In particular, the prevalence of Voice over IP (VoIP) communication services based on WebRTC is rising significantly. Their success relies on a good usability

A. Karpov et al. (Eds.): SPECOM 2018, LNAI 11096, pp. 408–418, 2018.
https://doi.org/10.1007/978-3-319-99579-3_43

and highest possible quality. WebRTC enables Internet Protocol (IP) and web-browser-based real-time communication using audio, video and auxiliary data without additional plugins or software installation. By default, WebRTC utilizes the Opus codec, standardized by the Internet Engineering Task Force (IETF) in RFC 6716 [1]. The Opus codec offers Full High Definition (HD) audio coding, by supporting a Fullband (FB) frequency range from 20 Hz to 20 kHz with low delays from 5 ms to 66.5 ms. However, the audio quality depends on several network-related parameters such as network bandwidth, packet loss, delay and jitter. Beyond the network-related parameters, WebRTC exhibits its own configuration of process variables, which may influence the call quality too. Consequently, the overall quality measurement, estimation and adjustment in the network are complex tasks. Therefore, the quality has to be monitored to guarantee a satisfying user experience, represented by e.g. the intelligibility of the call partner, the call continuity and the one-way delay.

In this article, we compare several, frequently used methods of audio quality assessment. Furthermore, we illustrate the challenges that arise for audio assessment in WebRTC-based communication and provide a novel solution approach for both, developers and providers. As a result, application developers can identify the reasons of degraded quality by locating the network segments with the biggest influence instead of detecting degradations in the overall audio quality only.

In Sect. 2, we summarize established methods of audio assessment within the described application environment. Moreover, Sect. 3 is dedicated to the shortcomings in monitoring WebRTC-based calls. Subsequently, we present the QuARTCS method, with its functioning principles for the acquisition of degraded audio signals from Secure Real-Time Transport Protocol (SRTP) streams – captured during an active WebRTC audio call at multiple measurement points in Sect. 4 – followed by preliminary results in Sect. 5 and some conclusions.

2 Methods of Speech Quality Assessment

2.1 Subjective Quality Assessment by Listeners

The ITU-Telecommunication Standardization Sector (ITU-T) recommendation P.800 describes several "methods for subjective determination of transmission quality" [3]. Absolute Category Rating (ACR) listening tests represent a commonly used method, in which the degraded audio signal is played to a group of probands, who rate the quality on a five-point opinion scale. The mean value of all individual ratings is called Mean Opinion Score (MOS)-ACR.

Besides listening tests, several instrumental methods for the assessment of audio quality exist. Figure 1 illustrates common steps of two communicating VoIP endpoints (not depicted in the figure) as well as the general functioning principle of subjective and objective audio quality assessments. In contrast to the ACR listening test, where only the degraded audio signal is taken into account during the assessment, a reference-based objective assessment algorithm additionally requires the original *reference audio sample*.

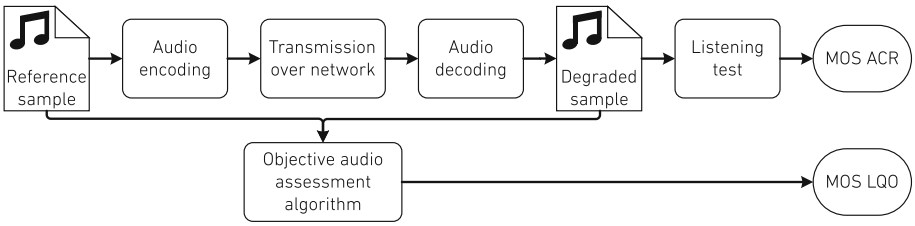

Fig. 1. Principle of subjective and objective audio quality assessment (derived from Maruschke et al. [2]).

2.2 Objective, Instrumental Quality Assessment

The ITU-T standardized several objective assessment methods for audio quality, which do not require a human rater, e.g. the well-known Perceptual Evaluation of Speech Quality (PESQ) algorithm [4] resulting to the measure Mean Opinion Score (MOS)-Listening Quality Objective - Narrowband (LQO_n). However, this assessment method is limited to Narrowband (NB) speech with a frequency range from 300 Hz to 3.4 kHz[1].

Meanwhile, real-time audio codecs enable a frequency range up to FB (e.g. the Opus codec), which also led to advanced audio assessment algorithms, such as Perceptual Objective Listening Quality Assessment (POLQA) [6]. The perceptual model of POLQA (defined in ITU-T P.863 version 2) supports Super-Wideband (SWB) speech with a frequency range from 50 Hz to 14 kHz, delivering a MOS-Listening Quality Objective - Super-Wideband (LQO_{sw}) measure. However, studies show that POLQA can even be used for a FB assessment of music or voice signals under certain conditions [2,7]. Recently, an update of ITU-T P.863 was introduced with version 3, which supports speech with a frequency range from 20 Hz to 20 kHz POLQA [6].

Apart from that, single-ended methods of assessment have been developed, which do not require a reference sample and which can therefore be utilized in a more flexible way, as it is limited to the access to the receiving communication party. The ITU-T P.563 algorithm from 2004 is the first standardized method supporting a single-ended, objective assessment [8]. However, it allows speech quality assessments for NB telephony only.

Beyond the chosen assessment method, VoIP calls pose a challenge, since the degraded audio samples have to be acquired after the network transmission.

2.3 Audio Injection and Recording Methods

To guarantee reproducibility and to minimize a possible influence of characteristics of the transmitted speech material itself, it is advantageous to inject

[1] An extension for the assessment of Wideband (WB) speech with a frequency range from 50 Hz to 7 kHz exists with ITU-T recommendation P.862.2. [5].

prerecorded audio samples into the sending endpoint. Especially reference-based assessment methods require well-defined speech samples.

According to the ITU-T recommendation P.863.1, an injection of reference samples in the sending endpoint can be done in three ways [9]:

Acoustically by an artificial mouth (from a head and torso simulator) connected to the client [10];
Electrically by connecting an audio cable from a playback device to a line input of the client;
Digitally by using Application Programming Interface (API) functions of the communication software (browser and web application) or methods provided by the operating system.

Additionally, the audio signal has to be recorded to acquire the degraded audio signal at the receivers' side after the network transmission – basically by utilizing one of the methods described for injection, with slight adaptions. However, performing the recording **acoustically** requires special equipment, and background noise has to be kept at a minimum to avoid additional distortion of the signal.

Recording the degraded sample **electrically** requires an audio output at the receiving endpoint, for example a sound card with a 3.5 mm line output jack, and an external recorder has to be connected to the endpoint output. A drawback of this method lies in the additional Digital to Analog Conversion (DAC) at the receiving endpoint and Analog to Digital Conversion (ADC) at the recording device. As the connection between both devices is analog, the transmitted signal is prone to interferences through radio waves or ground loops [11].

The **digital** approach of recording the audio is far less applicable between different devices, since modifications of the VoIP endpoint might be necessary. However, the advantage of this method lies in the non-modified recording of the degraded sample, which eliminates the described, potential signal distortions.

For the digital recording of a VoIP call, one can use an alternative method: In general, the encoded voice is transmitted over the network within Real-Time Transport Protocol (RTP) packets. Thus, one can capture the network traffic with a packet sniffer like *Wireshark* [12]. To acquire degraded audio signals, the RTP payload has to be extracted and eventually to be decoded. Utilizing this approach allows an audio recording independent of the receiving endpoint as target of the audio recording.

3 Limitations of Call Assessments in WebRTC

WebRTC is standardized by two major standardization bodies, namely the World Wide Web Consortium (W3C), which is responsible for the JavaScript (JS) API and the IETF for the corresponding protocols [14,15]. Merely a browser that follows the WebRTC protocol specifications and implements the JS API, defined by the W3C [14], is necessary. In some cases though, WebRTC native application, so-called "non-browsers", are preferable over WebRTC browsers.

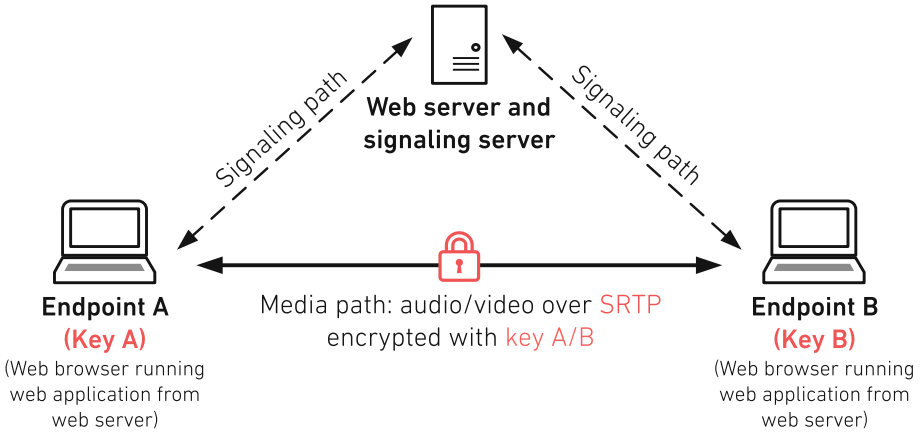

Fig. 2. WebRTC triangle architecture [13].

These WebRTC non-browsers do not require implementations of the JS API but must comply with the protocol specification [15].

A typical variant of the WebRTC architecture is depicted in Fig. 2. Two communication paths exist:

- The signaling path between the web-/signaling server or servers. Each WebRTC-client (in this example provided through web browsers) can also be represented by non-browsers,
- The media path between the communication parties.

The web and signaling servers provide the web application, which can be downloaded by the client, and also handle the signaling flow. The signaling protocol is not standardized, and various protocols, including standardized and proprietary ones, can be used but the inter-working with Session Initiation Protocol (SIP) over a signaling gateway must be possible. Therefore, the WebRTC media negotiation must include a representation of the same semantics as contained in Session Description Protocol (SDP) offers/answers used in SIP based VoIP communication [15,16].

The clients in a WebRTC call are named WebRTC endpoints and can either be WebRTC browsers or WebRTC non-browsers. Usually, the media path is established directly between two endpoints in terms of a Peer-to-Peer (P2P) connection. Under certain conditions, for example when symmetric Network Address Translation (NAT) is used, the traffic might be relayed through a Traversal Using Relays around NAT (TURN) server [17]. In all cases, the media data must be sent over SRTP for every channel that is established [18,19]. This means, that encryption must be used for the media path and that a cipher suite including a key exchange mechanism is necessary.

For WebRTC-based communication, a large variety of end devices (endpoints) can be used. Due to the heterogeneous nature of these end devices

in regard to hard- and software (e.g. operating systems, availability of audio jacks), a universal solution for capturing WebRTC audio signals does currently not exist. Hence, the use of a device-independent recording mechanism is stringently required. As described in Subsect. 2.3, the digital recording by capturing the network traffic is a suitable method for device agnostic acquisition of audio signals. Albeit, the traffic capturing method for the acquisition of the degraded audio signals, is still not trivial due to the encryption of the WebRTC-originated media streams.

4 QuARTCS Concept and Tooling

4.1 Design Principles

We developed QuARTCS as a tool, which allows the acquisition of degraded audio signals from SRTP streams captured during an active WebRTC audio media call at multiple measurement points along the network transmission path including the receiving endpoint. Consequently, the quality influences from one end to any point in the transmission path can be reflected by audio assessments (End-to-Any (E2A) assessment). The data acquisition includes the decryption of the SRTP packets of the captured stream, the payload extraction and the audio segmentation as preparation for an objective quality assessment, e.g. POLQA.

4.2 Functioning Details

Figure 3 illustrates the functioning principle of QuARTCS. In a first instance, a *reference sample* will be injected into the WebRTC application running within the WebRTC client on *Endpoint A*. At one of the endpoints (A or B), the encryption key, cipher suite and SDP messages have to be obtained (referred to as *Endpoint/reference information* in Fig. 3). The reference information is logged in *Endpoint A*. Before a call is established, the traffic capturing has to be started. The capturing can be conducted at any network node in the network path between *Endpoint A* and *B* or directly at *Endpoint B*. This can be accomplished with traffic capturing tools such as *Wireshark* or *tcpdump* running along the WebRTC application on the endpoint[2] [12,20]. Within the network path, the traffic can be captured by using a switch with *mirroring port* functionality, i.e., the actual traffic can be recorded with a third device connected to that *mirroring port* (cf. Meszaros and Maruschke [21]).

During the call, the *Reference sample* can be looped by the sending endpoint to provide several test samples during one call. Consequently, it is encoded and transmitted over the network by *Endpoint A*, whilst at the same time the traffic gets captured at the chosen capturing point(s). After the call finishes, the logged reference information, captured traffic as well as the *Reference sample* (if applicable) injected into *Endpoint A*, is delivered to QuARTCS.

[2] The devices need enough processing power to handle the call as well as the capturing simultaneously to prevent negative effects like a packet loss.

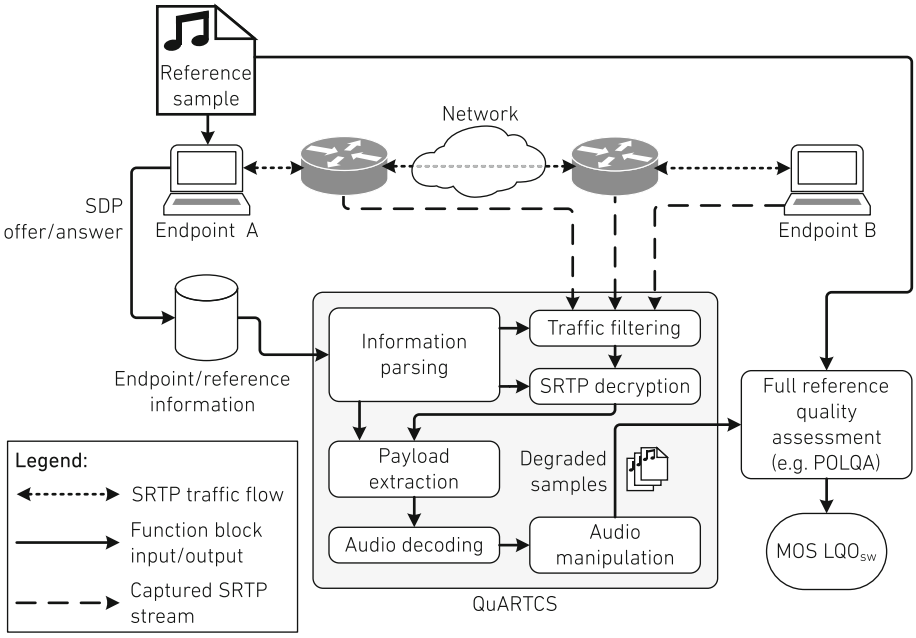

Fig. 3. General functioning principle of QuARTCS.

The *Traffic filtering* function of QuARTCS then filters the SRTP stream with direction from *Endpoint A* to *Endpoint B* according to information acquired from the SDP message by the *Information parsing* function. A possible filter condition can be the Synchronization Source (SSRC) identifier of the stream [22]. Additionally, the *Traffic filtering* has to incorporate a jitter buffer, resembling the jitter buffer functionality of the receiving endpoint. In the next step, the filtered SRTP stream is passed to the *SRTP decryption* function, which uses the key and cipher suite provided by the *Endpoint/Reference information* to the *Information parsing* function, which generates an unencrypted RTP stream. After the decryption, the payload – corresponding to the encoded audio – can be extracted from the RTP stream.

Afterwards, the encoded audio is decoded using the *Audio decoding* function[3]. If one *Reference sample* gets looped throughout the communication session by the sending *Endpoint A*, the result of the *Audio decoding* function will be a concatenation of the *Degraded sample*. As a result, this concatenation has to be split into multiple *Degraded sample* files to have the same length as the *Reference sample*, which is accomplished by the *Audio manipulation* function.

Finally, the *Degraded samples* are passed to the quality assessment model. In our example, the full-reference quality assessment model POLQA is utilized

[3] The decoding function has to incorporate an appropriate decoder for the specific audio codec, that was used for the communication session.

to estimate MOS-LQO$_{sw}$ values by comparing the *Degraded samples* with the *Reference sample*. Equally, a single-ended assessment method, such as P.563, can be used instead of POLQA, if no reference is available.

4.3 Exemplary Speech Assessment

To verify the functioning of QuARTCS, we conducted a preliminary test with a setup depicted in Fig. 4. *Endpoint A* (callers' PC) and all intermediary network devices were interconnected via an Ethernet connection supporting a maximum bit rate of 1 Gbit/s. *Endpoint B* (callee's smartphone in different positions) was connected to *Access Point 1* via a 2.4 GHz IEEE 802.11n wireless connection. The network traffic was captured simultaneously at *Switch 2* via a mirroring port, as well as directly at *Endpoint B* with *tcpdump*. Thereafter, a WebRTC call was established. During the call, a FB reference speech sample from ITU-T recommendation P.501 [23] was injected digitally into *Endpoint A* and repeated nine times by using API functions of the web browser. The repetition of the reference sample will result in 9 degraded samples that can be captured at each capturing point and consequently can be compared with the reference sample. After finishing the call, the traffic files from the two capturing points as well as the *Endpoint/reference information* (cf. Subsect. 4.2) acquired from *Endpoint A* was provided to the *PC with QuARTCS and POLQA*. The nine degraded samples acquired from the two capturing points, respectively, were evaluated with POLQA version 2.4 in SWB mode by comparing it with the injected speech sample as reference.

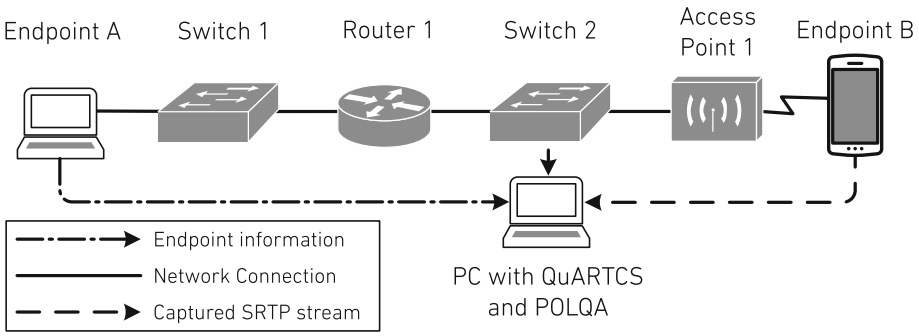

Fig. 4. Exemplary test design for verifying the functioning principle of QuARTCS.

5 Results and Discussion

Each of the nine samples, acquired with *Switch 2* as capturing point, achieved a MOS-LQO$_{sw}$ of 4.75 – the maximum in POLQA version 2.4. Considering

the samples obtained from *Endpoint B* as the capturing point, two out of nine achieved a lower rating than the maximum possible. Namely, *sample 4* was rated with a MOS-LQO$_{sw}$ of 3.88, while *sample 8* scores to 4.56. By analyzing the captured traffic itself, it can be observed, that no packet loss occurred in the network segment between *Endpoint A* and *Switch 2*. However, in the network segment between *Switch 2* and *Endpoint B*, several packets where lost during the transmission of *sample 4*. While *sample 8* was transmitted, even slightly more packets where lost. The fact that POLQA rated this sample higher than *sample 4* anyway can be justified on the grounds that most of the packets where lost during a period of silence that was part of the injected sample.

The preliminary tests showed that, QuARTCS allows an E2A assessment at multiple measurement points in the network transmission path simultaneously, including the receiving endpoint. This concept enables the identification of network segments, which cause the most significant degradations to the audio signal. As the tooling is accomplished by decrypting, extracting and analyzing the payload of the SRTP traffic, QuARTCS allows a quality assessment, which is independent of the endpoint characteristics and the WebRTC client implementation. The function blocks of QuARTCS work strictly modular and can easily be adapted to various audio codecs, provided that a standalone decoder is available. The digital acquisition of the degraded audio samples prevents an additional degradation due the measurement method itself.

Additionally, QuARTCS is able to pre-process the degraded audio samples (e.g. providing time alignment) to fulfill the requirements of a specific audio assessment method and is not limited to the usage of a certain assessment method. Established methods such as PESQ, POLQA and ITU-T P.563 can be utilized [4,6,8].

Nevertheless, a challenge lies in the determination of the key required for the decryption of the SRTP packets, depending on the key exchange algorithm within the WebRTC application. For instance, if Session Description Protocol Security Descriptions for Media Streams (SDES) is used for key exchange, the key can be obtained from the SDP messages [24]. However, if Datagram Transport Layer Security (DTLS) is utilized, the acquisition of the key might not be possible without the modification of the WebRTC application [25]. Additionally, the calculation of the one-way delay is not yet possible due to the encryption.

6 Conclusions

In this contribution, different assessment methods for voice call quality were compared, and the limitations of a quality assessment in WebRTC-based audio calls were described. Subsequently, we presented QuARTCS as a novel concept and tooling to enable the assessment of WebRTC calls. We described the general working principles of QuARTCS and demonstrated the basic functioning with a preliminary test. Finally, we illustrated the advantages of our approach but also its limitations.

The future studies will address the limitations, namely the calculation of the one-way delay despite the encryption, as well as the determination of the encryption key if DTLS is used for key exchange.

References

1. Valin, J., Vos, K., Terriberry, T.: Definition of the Opus Audio Codec. RFC 6716 (Proposed Standard). RFC. RFC Editor, Fremont, CA, USA, September 2012. https://doi.org/10.17487/RFC6716
2. Maruschke, M., Jokisch, O., Meszaros, M., Trojahn, F., Hoffmann, M.: Quality assessment of two fullband audio codecs supporting real-time communication. In: Ronzhin, A., Potapova, R., Németh, G. (eds.) SPECOM 2016. LNCS (LNAI), vol. 9811, pp. 571–579. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43958-7_69
3. ITU-T: Methods for Objective and Subjective Assessment of Quality-Methods for Subjective Determination of Transmission Quality. REC P.800, August 1996. http://www.itu.int/rec/T-REC-P.800-199608-I/en
4. ITU-T: Methods for Objective and Subjective Assessment of Quality Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs. REC P.862, February 2001. http://www.itu.int/rec/T-REC-P.862-200102-I/en
5. ITU-T: Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs. REC P.862.2, November 2007. https://www.itu.int/rec/T-REC-P.862.2-200711-I/en
6. ITU-T: Perceptual Objective Listening Quality Assessment (POLQA): An Objective Method for End-to-End Speech Quality Assessment of Wide-Band and Superwide-Band Telephone Networks and Speech Codecs. REC P.863. http://www.itu.int/rec/T-REC-P.863/en
7. ITU-T Study Group 12: A Subjective ACR LOT Testing Fullband Speech Coding and Prediction by P.863. Contribution SG12-C.22, 19 January 2017. https://www.itu.int/md/T17-SG12-C-0022/en
8. ITU-T: Single-Ended Method for Objective Speech Quality Assessment in Narrow-Band Telephony Applications. REC P.563, May 2004. https://www.itu.int/rec/T-REC-P.563/en
9. ITU-T: Application Guide for Recommendation ITU-T P.863. REC P.863.1, September 2014. https://www.itu.int/rec/T-REC-P.863.1/en
10. ITU-T: Application Guide for Objective Quality Measurement Based on Recommendations P.862, P.862.1 and P.862.2. REC P.862.3, November 2007. https://www.itu.int/rec/T-REC-P.862.3/en
11. Digital audio transmission for use in studio, stage or field applications. US4922536 A. Hoque, T. I., 1 May 1990. http://www.google.com/patents/US4922536
12. Wireshark-Community: Wireshark · Go Deep, 30 November 2017. https://www.wireshark.org/. Accessed 13 Dec 2017
13. Maruschke, M., Jokisch, O., Meszaros, M., Iaroshenko, V.: Review of the Opus codec in a WebRTC scenario for audio and speech communication. In: Ronzhin, A., Potapova, R., Fakotakis, N. (eds.) SPECOM 2015. LNCS (LNAI), vol. 9319, pp. 348–355. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23132-7_43

14. Jennings, C., Narayanan, A., Burnett, D., Bergkvist, A.: WebRTC 1.0: Real-time Communication Between Browsers. W3C Editor's Draft, 30 November 2017. http://w3c.github.io/webrtc-pc/
15. Alvestrand, H.T.: Overview: Real Time Protocols for Browser-based Applications. Internet-Draft, Fremont CA, USA, 12 November 2017. https://tools.ietf.org/html/draft-ietf-rtcweb-overview-19
16. Rosenberg, J., Schulzrinne, H.: An Offer/Answer Model with Session Description Protocol (SDP). RFC 3264 (Proposed Standard). RFC. Updated by RFC 6157. RFC Editor, Fremont, CA, USA, June 2002. https://doi.org/10.17487/RFC3264
17. Takeda, Y.: Symmetric NAT Traversal using STUN. Internet-Draft, Fremont CA, USA, June 2003. https://tools.ietf.org/id/draft-takeda-symmetric-nat-traversal-00.txt
18. Baugher, M., McGrew, D., Naslund, M., Carrara, E., Norrman, K.: The Secure Real-time Transport Protocol (SRTP). RFC 3711 (Proposed Standard). RFC. Updated by RFCs 5506, 6904. RFC Editor, Fremont, CA, USA, March 2004. https://doi.org/10.17487/RFC3711
19. Perkins, C., Westerlund, M., Ott, J.: Web Real-Time Communication (Web-RTC): Media Transport and Use of RTP. Internet-Draft, Fremont, CA, USA, 18 September 2016. https://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage-26
20. The Tcpdump Team: Tcpdump/Libpcap Public Repository, 3 September 2017. http://www.tcpdump.org. Accessed 13 Dec 2017
21. Meszaros, M., Maruschke, M.: Verhaltensanalyse von Einplatinencomputern Beim Transcoding von Echtzeit-Audiodaten. In: Elektronische Sprachsignalverarbeitung 2016. Tagungsband Der 27. Konferenz, vol. 81, pp. 237–245 (2016)
22. Lennox, J., Ott, J., Schierl, T.: Source-Specific Media Attributes in the Session Description Protocol (SDP). RFC 5576 (Proposed Standard). RFC. RFC Editor, Fremont, CA, USA, June 2009. https://doi.org/10.17487/RFC5576
23. ITU-T: Test Signals for Use in Telephonometry. REC P.501, March 2017. https://www.itu.int/rec/T-REC-P.501-201703-I/en
24. Andreasen, F., Baugher, M., Wing, D.: Session Description Protocol (SDP) Security Descriptions for Media Streams. RFC 4568 (Proposed Standard). RFC. RFC Editor, Fremont, CA, USA, July 2006. https://doi.org/10.17487/RFC4568
25. Rescorla, E., Modadugu, N.: Datagram Transport Layer Security Version 1.2. RFC 6347 (Proposed Standard). RFC. Updated by RFCs 7507, 7905. RFC Editor, Fremont, CA, USA, January 2012. https://doi.org/10.17487/RFC6347