

An Adaptive Bitrate Switching Algorithm for Speech Applications in Context of WebRTC

MOHANNAD ALAHMADI, National University of Ireland, Galway

PETER POCTA, University of Zilina

HUGH MELVIN, National University of Ireland, Galway

133

Web Real-Time Communication (WebRTC) combines a set of standards and technologies to enable high-quality audio, video, and auxiliary data exchange in web browsers and mobile applications. It enables peer-to-peer multimedia sessions over IP networks without the need for additional plugins. The Opus codec, which is deployed as the default audio codec for speech and music streaming in WebRTC, supports a wide range of bitrates. This range of bitrates covers narrowband, wideband, and super-wideband up to fullband bandwidths. Users of IP-based telephony always demand high-quality audio. In addition to users' expectation, their emotional state, content type, and many other psychological factors; network quality of service; and distortions introduced at the end terminals could determine their quality of experience. To measure the quality experienced by the end user for voice transmission service, the E-model standardized in the ITU-T Rec. G.107 (a narrowband version), ITU-T Rec. G.107.1 (a wideband version), and the most recent ITU-T Rec. G.107.2 extension for the super-wideband E-model can be used. In this work, we present a quality of experience model built on the E-model to measure the impact of coding and packet loss to assess the quality perceived by the end user in WebRTC speech applications. Based on the computed Mean Opinion Score, a real-time adaptive codec parameter switching mechanism is used to switch to the most optimum codec bitrate under the present network conditions. We present the evaluation results to show the effectiveness of the proposed approach when compared with the default codec configuration in WebRTC.

CCS Concepts: • **Information systems** → **Multimedia information systems**; *Information systems applications*; **Multimedia streaming**;

Additional Key Words and Phrases: E-model, Opus codec, speech coding, WebRTC, QoE

ACM Reference format:

Mohannad Alahmadi, Peter Pocta, and Hugh Melvin. 2021. An Adaptive Bitrate Switching Algorithm for Speech Applications in Context of WebRTC. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 4, Article 133 (November 2021), 21 pages.
<https://doi.org/10.1145/3458751>

1 INTRODUCTION

Web Real-Time Communication (WebRTC) [7] enables web browsers to provide audio, video, and auxiliary data exchange without the need for additional plugins. The standards used in

Authors' addresses: M. Alahmadi and H. Melvin, School of Computer Science, National University of Ireland, Galway, University Road, Galway, Ireland; emails: {m.alahmadi1, hugh.melvin}@nuigalway.ie; P. Pocta, Department of Multimedia and Information-Communication Technology, Faculty of Electrical Engineering and Information Technology, University of Zilina, Univerzitna 8215/1, SK-010 26, Zilina, Slovakia; email: peter.pocta@feit.uniza.sk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

1551-6857/2021/11-ART133 \$15.00

<https://doi.org/10.1145/3458751>

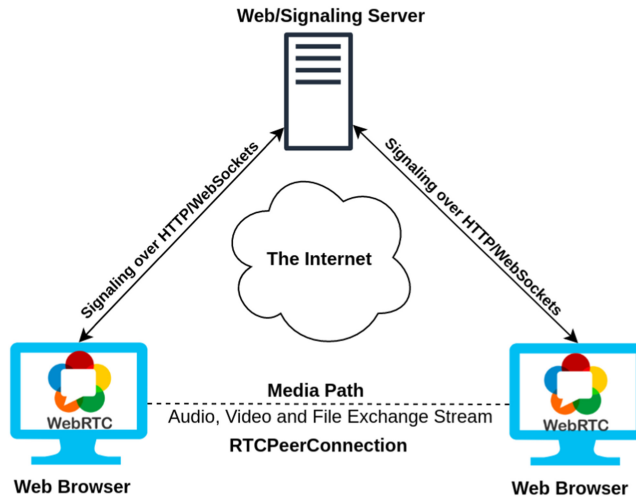


Fig. 1. WebRTC Triangle Architecture.

WebRTC make it possible for users to establish peer-to-peer multimedia sessions over IP networks. The Internet Engineering Task Force defines the communication protocols for the communication channel of WebRTC. World Wide Web Consortium (W3C) defined the HTML5 and JavaScript tags to enable transmission between the supported devices, which makes it easier for the developers to build interactive web and mobile applications using WebRTC. The multimedia communication added by WebRTC enables web browsers to interact using the so-called Triangle Architecture shown in Figure 1. WebRTC has rapidly evolved since it was released on May 2011 by Google. Shortly after the initial release, Ericsson Labs implemented its first development toolkit built on top of WebRTC. In 2013, the first cross-browser audio and video call was supported between Google Chrome and Mozilla Firefox, followed by the support for the Android operating system in 2014. The first Google RTC application known as Google Hangouts was partially built on WebRTC, whereas the latest Google Duo is fully based on the technologies provided by WebRTC. As of now, WebRTC is supported by Microsoft Edge, Apple Safari, and Opera in addition to Chrome and Firefox web browsers, as well as mobile operating systems like iOS and Android.

With increasing bandwidth capacity in IP networks, great effort is being placed on the **Quality of Experience (QoE)** for multimedia applications. For **Voice over Internet Protocol (VoIP)** applications, this is evident by support for a wide range of codecs, from the traditional **narrowband (NB)** telephony with a bandwidth of 300 to 3,400 Hz and **wideband (WB)** of 100 to 7,000 Hz, up to **super-wideband (SWB)** that offers a bandwidth of 50 to 14,000 Hz, enabling high-quality voice transmission. However, in public networks including wireless (WiFi) and cellular environments, network impairments such as delay and packet loss persist. Therefore, call quality assessment is a key to enable current VoIP systems to provide high-quality call services. To evaluate the degradation introduced by such impairments, subjective listening quality measurement is used to assess the quality perceived by the end user under various network conditions [39]. Such a method is known to be very accurate; however, it also can be time consuming and expensive to conduct in order to evaluate the quality of a voice call. As such, instrumental models have been developed to evaluate the quality using objective methods, which include intrusive models such as **Perceptual Evaluation of Speech Quality (PESQ)** [44], Perceptual Objective Listening Quality Assessment (POLQA) [46], non-intrusive models, e.g. ITU-T P.563 [38] and ANIQUE+ [26], and parametric models like the E-model [12].

The E-model is a simple computational model that was designed for transmission planning purposes. It considers a wide range of telephony-band impairments, particularly the impairment due to low bitrate codecs, one-way delay, packet loss, and the “classical” telephony impairments of loss, noise, and echo. The E-model computes an overall rating of the expected conversational quality, termed a *transmission rating factor* and denoted R , by combining effects of the different impairments on conversational speech quality. It is worth noting here that the transmission rating factor can be calculated in real time due to the E-model’s computational simplicity. For the NB E-model, the transmission rating factor scale ranges from 0 (poor) to 100 (excellent) and can be calculated as follows:

$$R = R_0 - I_s - I_d - I_{e,eff} + A, \quad (1)$$

where R_0 describes the basic signal-to-noise ratio. I_s combines the simultaneous distortions. I_d represents the impairment factor due to delay. $I_{e,eff}$ is the effective equipment impairment factor covering an impact of speech coding and in the presence of packet loss. A is the advantage factor that can be used to compensate for the impairments through other quality advantage factors. This work is focused on the impact of speech coding and packet loss and therefore consider no other distortions involved and use the default values defined in reference [15] R_0 , I_s , I_d and A parameters. Thus, Equation (1) can be re-written as follows:

$$R = R_{max} - I_{e,eff}. \quad (2)$$

R_{max} is the maximum value that can be used for NB conditions and is set to 93.2. The $I_{e,eff}$ can be calculated using Equation (3) coming from the ITU-T Rec. G.107 as follows:

$$I_{e,eff} = I_e + (95 - I_e) \frac{Ppl}{\frac{Ppl}{BurstR} + Bpl}. \quad (3)$$

I_e is the equipment impairment factor that represents the impairments caused by the speech codec. Ppl is the percentage of packet loss. Bpl represents the packet loss robustness. Both I_e and Bpl are codec-specific values and can be found in ITU-T Rec. G.113 [15]. The $BurstR$ was introduced by McGowan [30] as the ratio of average number of consecutively lost packets during a random loss event and can be measured using

$$BurstR = \frac{\text{average number consecutively lost packets}}{\text{average number of consecutively lost packets for random loss}}. \quad (4)$$

When $BurstR > 1$, the loss is considered bursty, and when $BurstR = 1$, it is considered random.

The resulted R factor from (1) can be translated into the Mean Opinion Score (MOS) using the following relation.

$$\text{For } MOS = 1.0 : R = 0 \quad (5a)$$

$$\text{For } 1.0 < MOS < 4.5 : MOS = 1 + 0.035 \cdot R + R \cdot (R - 60) \cdot (100 - R) \cdot 7 \cdot 10^{-6} \quad (5b)$$

$$\text{For } MOS \geq 4.5 : R = 100 \quad (5c)$$

As the E-model was designed for NB transmission systems, the scale for the maximum R was extended to cover WB speech codecs to reflect the improvement in the quality over NB speech codecs [34, 35, 48]. As a result, the $R_{max,wb}$ for WB, standardized in reference [13] is scaled up $R_{max,wb} = 129$. Therefore, (2) can be re-written as follows:

$$R_{wb} = 129 - I_{e,eff,wb}. \quad (6)$$

Similarly, recent contributions have suggested an extension to the R scale to cover SWB speech codecs [63]. As a result, the maximum scale for SWB proposed was $R_{max,swb} = 148$ and the R

factor for the SWB E-model can be calculated using

$$R_{swb} = 148 - I_{e,eff,swb}. \quad (7)$$

The following modification to Equation (3) is used for deriving the effective equipment impairment factor and packet loss robustness values for SWB codec conditions according to Mittag et al. [32]:

$$I_{e,eff,swb} = I_{e,swb} + (132 - I_{e,swb}) \frac{Ppl}{\frac{Ppl}{BurstR} + Bpl}. \quad (8)$$

The Internet Engineering Task Force standardized Opus codec [60] is the default speech codec used by WebRTC speech and audio applications. It is a multifunctional open source codec developed to support both interactive speech and music over the internet with a wide range of bitrates ranging from 6 kbit/s up to 512 kbit/s and supporting both **constant bitrate (CBR)** and **variable bitrate (VBR)** modes. It supports audio bandwidths from NB to fullband with sampling rates ranging from 8 to 48 kHz as well as frame sizes of 2.5 to 60 ms. The support for packet loss concealment and forward error correction makes it suitable for many usage scenarios over the internet. Opus is comprised of two previously known codecs, i.e., SILK [62] and CELT [59]. The two codecs are employed by Opus to provide three operation modes. The operation modes are LP mode (SILK only), MDCT mode (CELT only), [50], and hybrid mode. The LP mode is optimized for low bitrate speech signals and operates from NB to WB bandwidths. The MDCT mode is optimized for higher-quality speech and stereo music at higher bitrates. In the hybrid mode, the Opus coder combines both modes by using LP mode for lower frequencies up to 8 kHz and MDCT mode for higher frequencies and switches dynamically between the two modes in real time. It is worth noting here that a low complexity of the Opus codec makes this codec suitable for mobile phones and low-end embedded devices that usually have limited processing resources [5].

With the emergence of web technologies, there is a growing move toward web-based applications. Instead of relying on software that is based on a specific operating systems and environments, the availability of web browsers in every computer and mobile device makes it easy for a single web application to be accessible by the users anywhere. WebRTC takes advantage of such technology by enabling browser-to-browser **real-time communication (RTC)**. Such applications are mostly deployed in public best-effort networks, and the quality experienced by the users is what led to development of many strategies by researchers in the past years. The Opus codec is primarily used in WebRTC for encoding audio data. The low complexity with a range of bitrates and features makes it suitable for such applications. We take advantage of the functionalities offered by WebRTC to make the most out of the features offered by this codec to improve the quality perceived in WebRTC audio applications.

In this article, we build on earlier work and propose an adaptive codec bitrate switching mechanism for peer-to-peer audio applications using WebRTC. It includes a QoE model that receives as inputs the rate of packet loss at the receiving end. The model uses the NB, WB, and SWB E-model to output the expected degradation for different Opus codec conditions. This then informs a mechanism that selects optimal codec parameters.

The rest of the article is structured as follows. Section 2 reviews related research on the QoE strategies dealing with network impairments including codec switching and WebRTC QoE research. Section 3 details our proposed adaptive algorithm. Section 4 describes the experimental test-bed designed to evaluate the performance of the proposed approach. In Section 5, we summarize the results presented in this article. Finally, Section 6 concludes the article and details the planned future work.

2 RELATED WORK

This section includes a review of the literature for the areas involved in this work. Mainly, it is divided into four parts focusing on **quality of service (QoS)** and QoE strategies, QoE in WebRTC, codec and codec parameter switching strategies, and Opus E-model I_e and B_{pl} parameters.

2.1 QoS and QoE Strategies

The quality of speech in VoIP varies depending on the performance of the underlying IP networks that may introduce impairments like delay and distortions including packet loss, speech coding, and many other factors [22]. To cope with such impairments in packet-switched networks, researchers have been developing different QoS control strategies and QoE methods to improve users' experience [9]. A large body of work has focused on strategies to establish a relationship between QoS metrics and QoE in multimedia applications. A comprehensive review by De Carvalho and De Souza Mota [9] discussed the early strategies used by the researchers deploying source-based adaptive adjustment of the application-layer parameters. Those adjustments include switching the codec type, encoding rate and/or packetization size, and the transmission ends. Some of the strategies rely only on one of the objective models, i.e., intrusive or non-intrusive, and some use hybrid approaches to benefit from both methods. More recent work, by Han and Muntean [18], proposed a hybrid call quality assessment model that uses both an intrusive and a non-intrusive objective evaluation method for the NB Opus codec using the E-model, i.e., the non-intrusive method, in real time and then corrects the quality slowly deploying PESQ, i.e., the intrusive method, by considering packet **Loss Rate (LR)** in the link. The model obtains the network statistics at the start of the call and then repeatedly corrects itself by recording a short period of voice and running an offline assessment using PESQ at each end.

2.2 WebRTC QoE Strategies

There has been an extensive amount of work in the literature dealing with the performance delivered by WebRTC audio and video applications [17]. The performance of the Opus codec in WebRTC can achieve similar results to the quality delivered by the stand-alone Opus coder according to other works [23, 28]. Maruschke et al. [28] have identified a number of Opus key parameters that can be modified by changing the SDP messages to get the best performance out of the Opus codec that is usually limited by the default configuration in WebRTC. The same authors have recently proposed a tool to assess the quality of audio in the secured non-browser-based WebRTC applications [31]. The proposed method is deployed in multiple measurement points to capture and extract the audio payload carried by the Secure Real-Time Transport Protocol (SRTP) to process through POLQA and other objective models with some limitations due to the nature of the SDP protocol when dealing with encrypted multimedia streams.

2.3 Codec and Codec Parameter Switching Strategies

The switching of codec type in VoIP applications is usually done before or during the session in real-time [10, 37]. Aktas et al. [1] compared the quality of standard NB speech codecs under varying network conditions and proposed an adaptive end-to-end based codec switching scheme to adapt to changes in the available bandwidth during the call. This was done by continuous monitoring of the network conditions between two ends and using the re-INVITE message defined in the Session Initiation Protocol standard [51] to modify the session, i.e., switching to the most suitable codec. Assem et al. [6] evaluated the impact of adaptive codecs switching algorithms from an end-user perception point of view and proposed an algorithm to minimize such impacts. Compared to Aktas et al. [1], the study included both NB and WB codecs, i.e., G711, ILBC, GSM, SPEEX, and SILK.

The authors showed that the algorithm can produce a significant improvement to voice quality compared to the codec used at the start of the call only. However, switching codec types would introduce silent gaps that vary in length depending on the prevailing packet loss. Therefore, the number of switching codecs within a time interval should be limited to minimize the degradation in the quality perceived by the end user. Other strategies consider the switching of the encoding rate, i.e., bitrate [33, 49, 64]. Once the re-configuration of the bitrate is made by the encoder, the RTP header carrying the payload can help to inform the decoder about the new bitrate delivered by changing the field containing the payload information. So, the bitrate adaptation/change is seamlessly done by a codec, when the codec supports it. It is worth noting here that the seamless adaptation of codec parameters is supported by the Opus codec [24, 29, 60], the codec deployed in this work, as will be discussed in Section 3.4 of this article. More about codec and codec parameters adaptation strategies can be found in the work of De Carvalho and De Souza Mota [9].

2.4 Opus I_e and B_{pl} Factors

To deploy the E-model in NB and WB networks, ITU-T has standardized a list of the required I_e and B_{pl} values for a number of speech codecs in recommendation G.113 [15]. The recommendation does not provide those factors for some of the most recent speech codecs including the Opus codec. However, to derive the equipment impairment and packet loss robustness factors for any codec condition, the methodologies defined in the ITU-T recommendation in references [40] and [41] can be used to derive such factors for the NB and WB codec conditions, respectively, using the subjective methods. As the objective models have improved over time, they can predict the quality degradation of speech with a closer correlation to the subjective results. The ITU-T standardized a similar methodology in ITU-T Rec. P.834 [42] and ITU-T Rec. P.834.1 [43] to derive the I_e and B_{pl} for new conditions using the objective methods. For the Opus codec, as the main codec considered in this work, we have followed the method recommended in ITU-T Rec. P.834 to derive the $I_{e,nb}$ and B_{pl} factors for the NB bitrates [56]. Similarly in reference [55], we have followed recommendation ITU-T Rec. P.834.1 to derive the $I_{e,wb}$ and B_{pl} for the WB bitrates using the POLQA (v.3) as the instrumental model. There has not been a standardized method to derive the equipment impairment factors packet loss robustness for SWB conditions. However, the work proposed by Mittag et al. [32] can be used to derive $I_{e,swb}$ and B_{pl} for SWB speech codecs. The same method was used to derive the impairment factors for the 3GPP Enhanced Variable Rate (EVS) codec [8], currently listed in the recent amendment to the ITU-T G.113 recommendation [16]. In a recent work, we followed the same method to derive the equipment impairment factors $I_{e,swb}$ and packet loss robustness factors B_{pl} for the SWB Opus codec in other works [2, 54]. A list of all of the derived factors considered in this work can be found in the appendix to this article.

3 THE PROPOSED MODEL

This section details our proposed QoE model, which includes a quality estimation module consisting of the NB, WB, and SWB E-model combined with a database of the required equipment impairment factors for the Opus codec as the default codec deployed in WebRTC. The model benefits from the getStats API offered by WebRTC to extract the relevant QoS metrics taken from the Real-Time Transport Control Protocol sender and receiver reports for each RTC session. With the help of the real-time session re-negotiation functionality offered by WebRTC, we developed an adaptive algorithm that selects the most optimal Opus codec bitrate and operating mode based on the quality scores coming from the E-model. The proposed adaptive algorithm employs an adaptive VoIP architecture by following the concept of a closed feedback loop defined elsewhere [57]. The concept simply considers measurable values in every system state as a variable. The majority

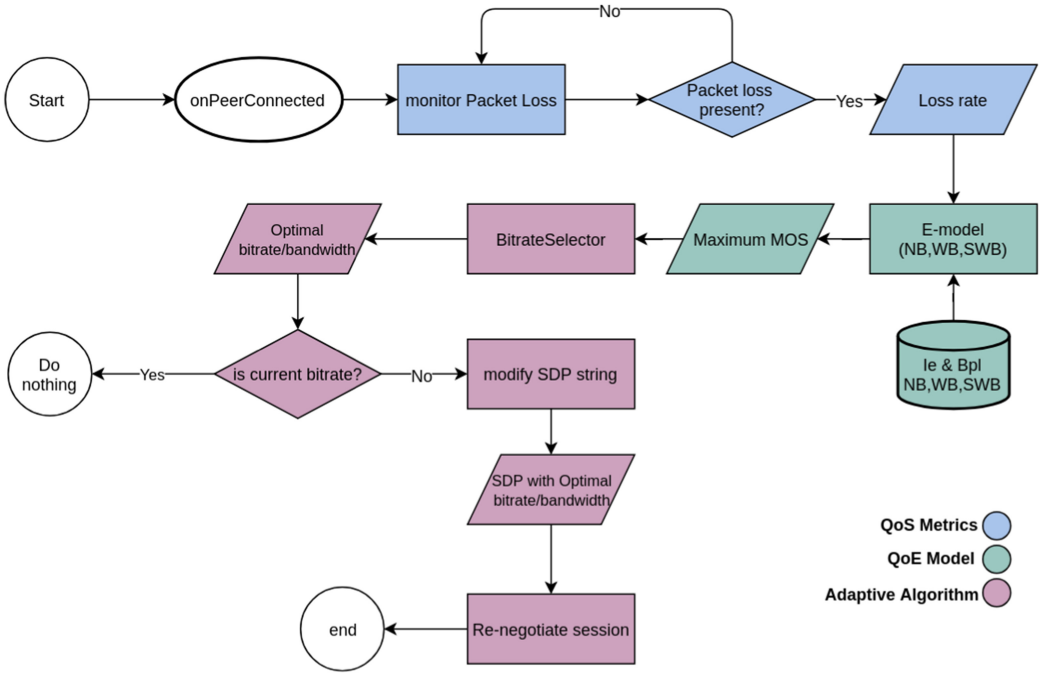


Fig. 2. Flowchart diagram of the adaptive algorithm.

of adaptive implementations in VoIP applications deal with four key variables described in the work of De Carvalho and De Souza Mota [9] as follows:

- *Observation parameters*: Variables that can be measured to assess the performance of the network. QoS parameters like delay, packet loss, and other impairments that are not controlled by the sender and receiver.
- *Decision metrics*: Parameters used to evaluate the characteristics of the audio session over a period of time or a number of samples. This includes QoE parameters like the equipment impairment factors used by the E-model and the resulted MOS, i.e., an indicator of quality perceived by the end user.
- *Performance reference*: This is a variable that can be used as a reference to make decisions by one of the session endpoints. This can be a threshold of a minimum or maximum MOS used to make a decision on how to modify any parameters related to the audio session.
- *Adjustable parameters*: The parameters that can be adjusted based on the three variables mentioned previously. A model can change any codec-related parameters at the sender side or the selection of the Dejitter buffer strategies at the receiver side to improve the quality.

The *observation parameters* in our QoE model are the packet loss metrics and the impact of coding using the Opus codec. The MOS provided by the E-model is considered as the *decision metric*. Opus codec configuration including encoding bitrate and operating mode that results in the maximum MOS based on the LR and type and type is the *performance reference*. Finally, the configurable Opus bitrates and operating mode are considered the *adjustable parameters*. A flow diagram of the higher-level components of the our adaptive algorithm is illustrated in Figure 2. The QoE model can dynamically perform the following functions at a fixed interval in real time:

- (1) Monitor, detect the presence of packet loss, and calculate the resulted LR.
- (2) Evaluate the impact of the loss condition on each of the NB, WB, and SWB Opus bitrates available.
- (3) Compare the resulted MOS from all bitrates available to suggest the most optimal Opus codec condition.
- (4) Re-negotiate the session using the most optimal Opus bitrate and operating mode by modifying the SDP message.

3.1 WebRTC getStats API

The getStats API described in the work of Bergkvist et al. [7] provides all QoS metrics taken from the Real-Time Transport Control Protocol sender's and receiver's report for each peer participating in any WebRTC audio and video session. The metrics provided by the getStats API are built-in Google Chrome and Mozilla Firefox web browsers. For example, Google Chrome offers a dashboard to visualize the QoS metrics for every participant using an audio and/or video WebRTC application. This feature can be accessed by navigating to the following address: `chrome://webrtc-internals`. Similarly, "about:webrtc" can be used in the Mozilla Firefox web browser. The getStats API can also be called by web applications running within the browsers to return specific statistics to monitor the performance of the underlying network for each session. A definition of all metrics provided by the getStats API is described in the work of Alvestrand and Singh [3]. The following are the QoS metrics used by our QoE model in WebRTC deployed in Chrome web browser:

- *packetsLost*: The number of lost RTP packet for each SSRC according to RFC3550.
- *totalSamplesDuration*: The duration in seconds of the session since the first RTP packet was transmitted.
- *bytesReceived*: The total number of bytes including packet overhead of the audio data received.

3.2 Packet Loss Calculation

The calculation of packet LR is done by calling the *packetsLost* metric from the getStats API. We set the frequency of calculating the LR to be done for every 10 seconds to collect enough information regarding the loss presented. As a result, we lower the number of switching of codec configuration that may occur during brief events of packet loss. Computation and detection of bursty packet loss conditions requires the calculation of the average number of consecutively lost packets. However, the average number and loss distribution information are not provided by the current version of the getStats API. Therefore, our adaptive algorithm can not detect bursty loss behaviors in the network. Instead, we calculate the LR and consider loss type as random. The impact of the bursty loss and the performance of the adaptive algorithm under the bursty loss condition is evaluated in Section 4. The experimental test-bed described in Section 4 provides a controlled testing environment. Thus, we can specify the type of loss pattern used, then inform the E-model to operate accordingly.

3.3 E-model Implementation

The quality estimation aspect of our QoE model uses the NB, WB, and SWB E-model versions as described in Section 1. To implement the E-model in our QoE model in WebRTC, a database of the $I_{e,eff}$ and Bpl factors from references [54–56] for Opus codec version 1.2 are used. Table 1 lists all of the bitrates covered by the NB and WB bandwidths, as well as a selected number of bitrates that cover the SWB bandwidth up to 40 kbps, which is considered as the maximum encoding bitrate for speech applications according to Valin et al. [60]. It is worth noting here that Opus can

Table 1. Bitrates and Operating Modes Supported by Opus codec (v. 1.2.1) for Speech Applications

Bandwidth	Bitrate	Mode
SWB	14 kbps	vbr
	15 kbps	vbr
	16 kbps	vbr/cbr
	19 kbps	vbr/cbr
	22 kbps	vbr/cbr
	25 kbps	vbr/cbr
	28 kbps	vbr/cbr
	31 kbps	vbr/cbr
	34 kbps	vbr/cbr
	37 kbps	vbr/cbr
	40 kbps	vbr/cbr
WB	11 kbps	vbr
	12 kbps	vbr/cbr
	13 kbps	vbr/cbr
	14 kbps	cbr
	15 kbps	cbr
NB	6 kbps	vbr/cbr
	7 kbps	vbr/cbr
	8 kbps	vbr/cbr
	9 kbps	vbr/cbr
	10 kbps	cbr
	11 kbps	cbr

encode at a lower bitrates in WB and SWB when operating at VBR mode. This is because VBR mode in the Opus codec is more efficient and can produce a better quality compared to encoding at the same bitrates in CBR mode [60]. The quality estimation part of the QoE model takes the LR as an input. Each implementation of the E-model calculates the $I_{e,eff}$ for every Opus condition listed in Table 1. Equation (3) is used for calculating the $I_{e,eff}$ factor for the NB and WB bitrates. When it comes to SWB mode, the $I_{e,eff,swb}$ and Bpl values are calculated using (8) instead. The NB E-model uses $I_{e,eff,nb}$ and Bpl values to calculate the R-factor R_{nb} using $R_{max} = 94.15$ according to reference [14]. Similarly, the WB E-model calculates R_{wb} using Equation (1) and $R_{max} = 129$ for the WB bitrates. $R_{max} = 148$ is used to calculate R_{swb} for SWB bitrates. The resulted R factors for each implementation of the E-model are then transformed into the MOS scale using Equation (5). The BitRateSelector() module compares all provided MOS and selects the codec condition resulting in the highest MOS as shown in Figure 3. It is worth noting here that as the equations deployed to get the MOS, i.e., Equations (2) through (8), include rather simple mathematical operations and the corresponding comparison, as well as selection processes represent rather simple processes, the computational cost is rather small, i.e., a few milliseconds, making it suitable for real-time applications. To get an exact value of the computational cost, the JavaScript built-in function performance.now() was utilized to evaluate the processing time taken by the algorithm, and the resulted computational time was between 2 and 3 ms when tested on a 2.5-GHz Dual-Core Intel i5 machine. The module checks if the selected codec condition is currently being used; otherwise, if

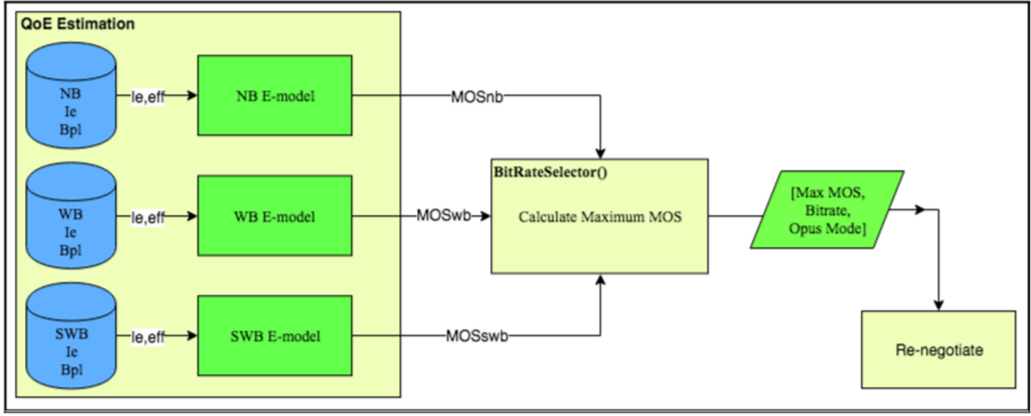


Fig. 3. Implementation of the NB, WB, and SWB E-model.

```

Remote peer's sdp: v=0
o=- 253590174521004296 2 IN IP4 127.0.0.1
s=-
t=0 0
a=group:BUNDLE 0 1
a=msid-semantic: WMS VvUbwpegA7c3AEdd3y1t1FCR0Wnyuo2FHfXq
m=audio 9 UDP/TLS/RTP/SAVPF 111 103 104 9 0 8 105 103 110 112 113 126
c=IN IP4 0.0.0.0
a=rtcp:9 IN IP4 0.0.0.0
a=ice-ufrag:4a0n
a=ice-pwd:07NXXsydF3Roahp93P7hX50d
a=ice-options:trickle
a=fingerprint:sha-256 36:C7:4D:22:B5:31:FB:8A:87:CD:15:D2:66:6A:21:21:FC:05:CE:E3:3B:47:EC:1F:89:5A:EB:03:A6:A4:9E:1C
a=setup:actpass
a=mid:0
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=extmap:2 https://www.ietf.org/id/draft-holme-rmcat-transport-wide-cc-extensions-01
a=extmap:3 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:4 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=extmap:5 urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id
a=sendrecv
a=msid:VvUbwpegA7c3AEdd3y1t1FCR0Wnyuo2FHfXq 51aada74-944b-4b61-9b60-94bae9b3041c
a=rtp-mux
a=rtpmap:111 opus/48000/2
a=rtp-fb:111 transport-cc
a=rtcp:111 minptime=10;useinbandfec=1
a=rtpmap:103 ISAC/16000
a=rtpmap:104 ISAC/32000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMU/8000
a=rtpmap:8 PCM/8000
a=rtpmap:106 CN/32000
a=rtpmap:105 CN/16000
a=rtpmap:13 CN/8000
a=rtpmap:110 telephone-event/48000

```

Fig. 4. Sample of an SDP message for a WebRTC audio session.

another codec condition is suggested, it then outputs the bitrate and the operating mode with the expected MOS for the new codec configuration.

3.4 Session Re-negotiation

As discussed in Section 1, WebRTC peers use the SDP offer/answer inspired by Session Initiation Protocol session negotiation flow. The negotiation of multimedia sessions in WebRTC is done by exchanging the SDP messages between the peers in a text-based format over the signaling channels. An SDP message carries information of the media types and other parameters related to each peer to establish an audio and/or video session. An example of an “audio-only” session exchange message is illustrated in Figure 4.

Each line in the SDP message carries information of the type of media (audio, video), transport protocol (RTP/UDP), and the format of the media type including audio- and video-related parameters. A definition for all parameters used by SDP is described in the work of Handley et al. [19].

SDP munging or SDP modification according to Levent-Levi [27] is used to modify any parameter within the SDP messages before being exchanged between the session endpoints. WebRTC createOffer() or createAnswer() functions use the SDP messages to set up and establish audio/video sessions. The sessions can be re-negotiated while running to add, remove, or change any media-related parameters. Nandakumar and Jennings [36] describe all possible scenarios and SDP formats

Table 2. Example of Switching Bitrate and Mode Using SDP Lines

Bitrate	Mode	SDP Line Example
9 kbps	cbr	a = fmltp:111 maxaveragebitrate = 9216; isCBR = 1
12 kbps	vbr	a = fmltp:111 maxaveragebitrate = 12288; isCBR = 0
32 kbps	cbr	a = fmltp:111 maxaveragebitrate = 32768; isCBR = 1
40 kbps	vbr	a = fmltp:111 maxaveragebitrate = 40960

that can be used in JSEP to set up WebRTC sessions. For the Opus codec, a list of codec-specific parameters defined in other works [58, 60] can be modified before and during codec operation in WebRTC. Spittka et al. [58] detail all parameters that can be modified using the SDP messages for both sender and receiver. In our QoE model, we use the following receiver-only parameters that can be requested by the receiver and can affect one way of the transmission:

- **maxaveragebitrate:** The maximum average coding bitrate supported by the decoder. The receiver can use this parameter to inform the sender, i.e., encoder, to limit the encoding bitrate to save resources. When a frame size of 20 ms is used, Opus supports bitrates from 6,000 bits and up to 40,000 bits for speech and up to 128,000 bits for fullband stereo music. Table 1 lists the maxaveragebitrate for Opus (version 1.2) and the default operating sampling rate based on the findings in other works [2, 54, 55].
- **isCBR:** The Opus codec operates in VBR mode by default. In this mode, the encoder can deliver the highest quality by encoding the audio data depending on the available audio signals at the selected maxaveragebitrate. In CBR mode (VBR), the encoder encodes each packet with a limited number of data regardless of the number of the audio signals. According to the work of Valin et al. [60], the CBR mode delivers lower quality and adds more processing complexity compared to VBR mode.

We have implemented the Re-negotiate() module to handle the adaptive codec parameter negotiation during any active audio session in WebRTC. The SDP line starting with “a = fmltp:111” can be used as described by Spittka et al. [58] to adjust Opus parameters. The parameters “maxaveragebitrate” and “isCBR” are mainly adjusted to change the codec parameters depending on the output of the BitRateSelector() module in our implementation. Once the codec configuration with the highest MOS is returned by the BitRateSelector() module, the Re-negotiate() module modifies the SDP line by adding the recommended Opus bitrate and operating mode to the “a = fmltp:111” line. Table 2 includes examples on how this can be achieved. The required bitrate used by “maxaveragebitrate” is specified in bits per second. The parameter “isCBR” takes a Boolean value, where 1 indicates the use of CBR mode and 0 is used to operate at VBR mode. If the parameter “isCBR” is not specified, the encoder will use the VBR mode by default.

4 EXPERIMENTS

Once the components of our adaptive QoE model were operational, we designed a test-bed to evaluate the performance of the quality estimation by the model and the adaptive codec bitrate switching mechanism as illustrated in Figure 5. A sender and receiver are connected while running an audio-only WebRTC session over an RTP stream. A simple script is deployed at the sender side, which acts as a controller to handle the playing of the reference speech signals at the sender’s side and records the degraded signals at the receiver’s side simultaneously. The QoS metrics are monitored in real time by the model as the RTP packets arrive at the receiving end. The adaptive bitrate switching mechanism can be enabled or disabled to assess the quality resulted from the adaptive

Table 3. Adaptive Algorithm and Packet Loss Experiments

Experiment	Description	No. of Files
1.1	Random loss and AdaptiveOFF	1,024
1.2	Bursty loss and AdaptiveOFF	1,024
1.3	Random loss and AdaptiveON	1,024
1.4	Bursty loss and AdaptiveON	1,024
Total		4,096

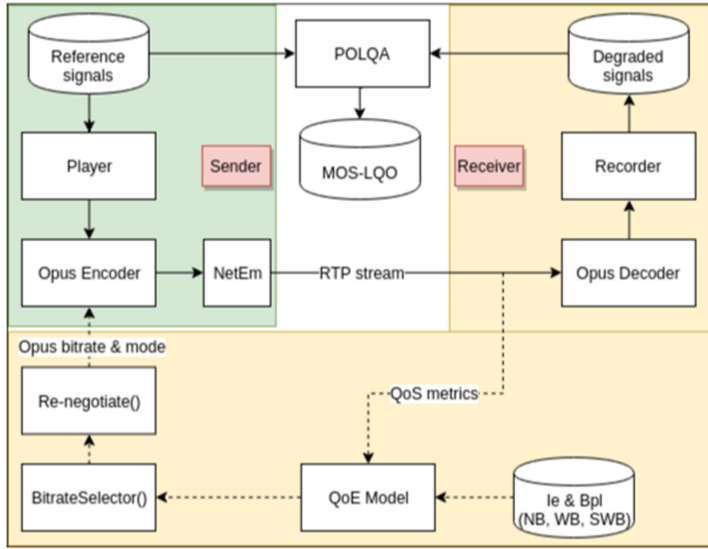


Fig. 5. Experimental Test-bed Architecture.

codec switching independently. The most recent version of POLQA (v.3) was trained to predict the quality degradation for Opus-coded speech signals [53]. In the final phase of each experiment, we obtain the MOS-LOQ quality score from the POLQA model to evaluate the performance of the codec bitrate switching mechanism. The main goals of building this experimental test-bed are as follows:

- To quantify the impact of random and bursty loss conditions on the performance delivered by Opus codec in WebRTC.
- To compare the quality resulting from different packet loss conditions for the proposed adaptive bitrate switching mechanism versus the default WebRTC codec configuration.

Table 3 lists all of the experiments conducted to assess the performance of the adaptive algorithm. Experiments 1.1 and 1.2 evaluate the quality of the AdaptiveOFF scenario, i.e., when the default Opus codec configuration under WebRTC is used and the adaptive algorithm is disabled, under random and bursty loss conditions. Experiments 1.3 and 1.4 assess the quality of using AdaptiveON, i.e., when the adaptive algorithm is enabled.

The following sections detail the functionalities of each module implemented in the experimental test-bed to evaluate quality for the goals listed previously.

4.1 Audio Injection and Recording

The test-bed uses the reference-based assessment model POLQA to evaluate the performance of the codec bitrate switching mechanism in our adaptive algorithm. To ensure the reproducibility and eliminate the degradation introduced by input and output equipment, the injection of the prerecorded speech signals is used. ITU-T Rec. P.863.1 describes three methods that can be followed to use the audio injection method [47]:

- **Acoustically:** Injecting the reference signal using an artificial mouth by connecting a head and torso simulator to the input of the end terminal [45].
- **Electrically:** By connecting an audio cable directly from the output device to the input of the end terminal through the 3.5-mm jack.
- **Digitally:** Reference signals are injected locally using software to map the output audio device to the input device and act as the microphone and vice versa.

We choose the digitally injected and recorded signals approach in this experimental test setup. The Linux Pulse Audio Volume Control application *pavucontrol* [25] was used for this purpose. We play the reference audio signals into the sender's WebRTC application, bypassing the microphone, and recording the audio at the receiver's side, bypassing the speakers. A digital player module is implemented at the sender's side to inject the reference signals taken from a database of 32 test signals from ITU-T Rec. P.501 Annex C [4] in eight different languages, i.e., English, German, Italian, French, Finnish, Chinese, Japanese, and Dutch. Each reference speech signal consists of two short sentences separated by a short period of silence, resulting in 8-second-long speech samples as described in reference [44]. The output of the player module feeds the audio signals into WebRTC to be processed by the encoder. On the other end, a recording module is implemented at the receiver's side to digitally record the degraded speech signals to be processed by any objective model described in Section 4.2.

4.2 Packet Loss Module

To focus on the impact of packet loss in a controlled testing environment, the network emulation tool NetEM [21] is deployed at the sender side to introduce degradation using eight packet loss rates, i.e., 1%, 3%, 5%, 10%, 15%, 20%, 25%, and 30%, to the outgoing RTP packets using the Bernoulli model [52] to generate random loss patterns and the Gilbert-Elliott model [20] to generate bursty loss patterns. The packet LR experienced by the end user were selected to cover a broad range of packet LR experienced by the end user in current telecommunication networks, especially in the case of Over-the-Top VoIP services. When it comes to the bursty loss patterns, we used the restricted loss space method described in the work of Varela et al. [61] to create a list of LR and **mean loss burst size (MLBS)** values to ensure the stability of the Gilbert loss model when we generated the loss patterns for the rather short speech signals used in this experiment. The combinations of the used LR and MLBS values are listed in Table 4. It is worth noting here that an impact on the quality depends greatly on the location of the dropped frame in any of the test signals—to be more precise, whether the frame contains silence or voiced data. Therefore, we have repeated the process four times for every test condition. This procedure resulted in 1,024 test cases. Applying bursty loss patterns using NetEm is less straightforward. Although it supports most of the loss models in transmission networks, NetEM requires the additional parameters p and r relating to the bursty loss behavior. It is worth noting here that p is the probability of a transition from state G (the Good state), where no packets are lost, to state B (the Bad state), where all packets are lost. r denotes the probability of going from state B to state G, which can apply the Simple Gilbert loss. As the LR and MLBS are only required to generate bursty loss behavior, we used the Simple Gilbert

Table 4. List of LR and MLBS and their Corresponding p and r Values for NetEm

LR	MLBS	p	r
1%	2	0.51	50
3%	2	1.55	50
5%	4	1.32	25
10%	4	2.8	25
15%	5	2.3	13
20%	6	3.25	12
25%	6	4	12
30%	6	5.15	12

model implementation under NetEM for this purpose. According to the inspection done by Fosser and Nedberg [11] to test the behavior of NetEM when applying bursty loss patterns using different loss models, Equation (6) can be used to apply bursty loss using a combination of the required LR and MLBS values.

$$p = \frac{\pi B * r}{1 - \pi B} \quad (6)$$

For instance, if an LR of 5% and MLBS = 4 is required, p can be calculated as follows:

$$p = \frac{0.05 * 25}{1 - 0.05} = 1.32.$$

Combinations of the used LR and MLBS values along with their corresponding p and r required by NetEM for the LRs used in the validation test-bed experiments are listed in Table 4. It is worth noting here that an impact on the quality under both random and bursty loss depends greatly on the location of the dropped frame in any of the test signals—to be more precise, whether the frame contains silence or voiced data. Therefore, we have repeated the process four times for every test condition. This procedure resulted in 4,096 test cases.

4.3 Perceptual Speech Quality

The degraded and decoded audio data delivered by the RTP stream, recorded using the digital recording module at the receiver's side, was stored into a database to be later used as an input to the instrumental model, i.e., POLQA, to evaluate degradation introduced by the packet loss conditions. The POLQA model can deal with the change of bandwidth during the instantaneous codec bitrate switching [47] that makes it suitable for this experimental setup.

5 EXPERIMENTAL RESULTS

Next, we report results obtained from the evaluation test-bed for the packet loss conditions selected in Section 4.2 for both random loss using the Bernoulli model and the Gilbert-Elliott model for the bursty loss. Figure 6 compares the quality degradation for the MOS-LQO scale under both loss conditions. It is clear that the bursty loss has a lower MOS than the random loss condition at the same LR. This concurs with the results reported in reference [54] for the impact of loss conditions and the resulted impairments factors arising from random and bursty loss patterns.

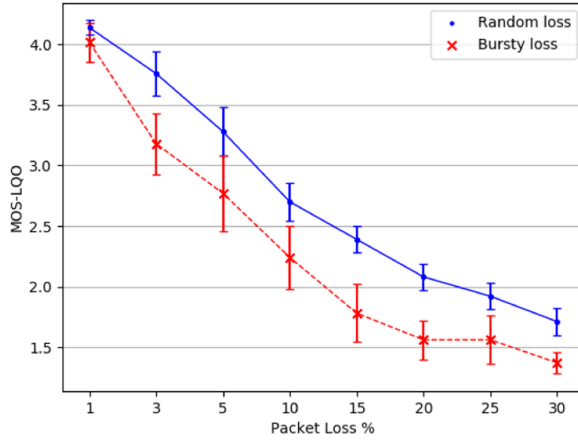


Fig. 6. Random vs. bursty loss MOS quality. The vertical bars show a 95% CI computed over 128 MOS-LQO values.

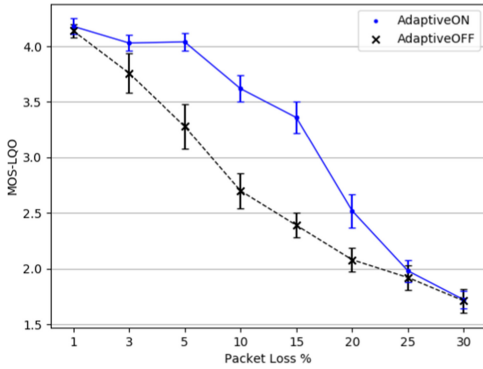


Fig. 7. Adaptive algorithm under random loss. The vertical bars show a 95% CI computed over 128 MOS-LQO values.

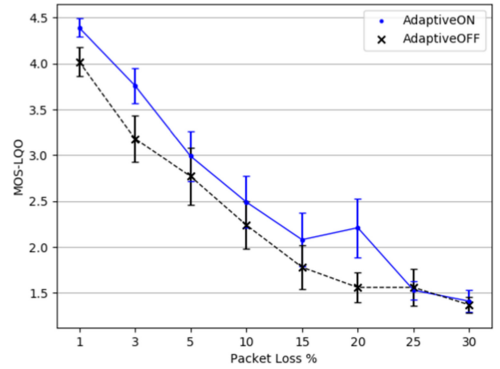


Fig. 8. Adaptive algorithm under bursty loss. The vertical bars show a 95% CI computed over 128 MOS-LQO values.

Figure 7 shows the quality degradation on the MOS-LQO scale for AdaptiveON, i.e., when the adaptive algorithm is enabled (solid blue line) against the quality resulted for AdaptiveOFF for when the adaptive algorithm is disabled (dashed black line). AdaptiveOFF is considered as the default WebRTC configuration for the Opus codec. Detailed MOS and their confidence intervals (CI 95) are listed in Table 5. According to the MOS reported, AdaptiveON has a clear advantage over AdaptiveOFF. There is an improvement of the MOS starting from 3% to 20% for AdaptiveON; however, it is unlikely for users to tolerate a quality degradation for scores below an MOS of 3, which makes the adaptive algorithm usable for LRs up to 15%. In Figure 8, the MOS for AdaptiveON (solid blue line) shows an improvement over the default Opus configuration used in WebRTC, i.e., AdaptiveOFF for 1% and 3% LRs in particular. There is no significant improvement for AdaptiveON for LRs beyond 3%. Even though an improvement of the MOS can be seen at 20%, the actual MOS is 2.21 as listed in Table 6, which is considered poor and most likely will not improve the quality perceived by the users.

Table 5. MOS and CI 95 Under Random Loss

LR	AdaptiveOFF		AdaptiveON	
	MOS	CI 95	MOS	CI 95
1%	4.14	0.06	4.18	0.07
3%	3.76	0.18	4.03	0.07
5%	3.28	0.20	4.04	0.08
10%	2.70	0.16	3.62	0.12
15%	2.39	0.11	3.36	0.14
20%	2.08	0.11	2.52	0.15
25%	1.92	0.11	1.98	0.10
30%	1.71	0.11	1.72	0.08

Table 6. MOS and CI 95 Under Bursty Loss

LR	AdaptiveOFF		AdaptiveON	
	MOS	CI 95	MOS	CI 95
1%	4.02	0.16	4.39	0.10
3%	3.18	0.25	3.76	0.19
5%	2.77	0.31	2.99	0.27
10%	2.24	0.26	2.49	0.28
15%	1.78	0.24	2.08	0.29
20%	1.56	0.16	2.21	0.32
25%	1.56	0.20	1.53	0.1
30%	1.37	0.09	1.41	0.12

It is worth noting here that the results listed in Tables 5 and 6 still show a higher degradation introduced by the bursty loss conditions for both AdaptiveON and AdaptiveOFF test cases.

6 CONCLUSION AND FUTURE WORK

In this article, an adaptive codec bitrate switching algorithm is proposed for speech applications in WebRTC. The adaptive algorithm employs a QoE model deploying the NB, WB, and SWB E-model to assess the combined impact of speech coding and packet loss. The implemented E-model utilizes a database containing the equipment impairment factors that had already been derived instrumentally for all bitrates supported by the Opus codec for speech applications. Based on the calculated LR and the estimated quality degradation, the model then informs a codec parameter switching mechanism to select the most optimal bitrate and operating mode using the Opus codec. The experimental results show that the proposed approach can effectively lower the impact of random packet loss and, to some extent, bursty loss conditions. In the future, we would like to extend the implementation of the proposed adaptive approach by adding the following functionalities:

- Include the I_d factor in Equation (1) to consider the impact of end-to-end delay when calculating the R factor to estimate a conversational quality experienced by the end user. The following QoS metrics are returned by the getStats API parameters and can be used for this purpose:
 - *googCurrentDelayMs*: Delay time at the receiving side in milliseconds.
 - *googJitterBufferMs*: The size of the jitter buffer measured in milliseconds.
- When dealing with the issue addressed in Section 3.2 regarding the automatic detection of the packet loss type, i.e., random and bursty, we consider using third-party tools like tcpdump or Tshark. Such tools are non-web browser based, and therefore, we will be implementing an additional script to bridge the QoS statistics taken from such tools to make a decision regarding the type of loss presented in the network.

APPENDIX

A OPUS CODEC I_e AND B_{pl} FACTORS

The following sections of this appendix list all of the equipment impairment factors used by the QoE model in the proposed adaptive algorithm. The presented factors cover the I_e and B_{pl} for the narrowband, wideband, and super-wideband Opus codec as discussed at the end of Section 2.4. The factors cover both VBR and CBR operating modes by Opus codec for mono speech.

A.1 SWB

Table 7. $I_{e,SWB}$ for VBR and CBR SWB Opus

Bitrate	VBR $I_{e,SWB}$	CBR $I_{e,SWB}$
14	38.1	–
15	34.1	–
16	30.16	36.88
19	23.80	29.58
22	20.22	23.38
25	20.20	22.64
28	18.48	23.65
31	16.82	21.74
34	14.76	20.62
37	12.08	16.23
40	10.67	15.90

Table 8. Bpl for VBR and CBR SWB Opus Under Random and Bursty Packet Loss

VBR Bpl			CBR Bpl		
Bitrate	Random	Bursty	Bitrate	Random	Bursty
14	16.51	9.13	14	–	–
15	15	7.9	15	–	–
16	11.83	10.06	16	13.30	10.80
19	10.67	8.92	19	10.74	8.93
22	10.47	8.57	22	10.01	8.46
25	11.28	9.18	25	10.57	8.74
28	10.63	8.91	28	11.67	9.98
31	10.44	8.82	31	11.51	9.58
34	10.12	8.54	34	11.78	9.69
37	10.38	8.30	37	10.62	8.99
40	9.79	8.04	40	11.35	9.37

A.2 WB

Table 9. $I_{e,WB}$ for VBR and CBR WB Opus

Bitrate	VBR $I_{e,WB}$	CBR $I_{e,WB}$
11	28.4	–
12	23.3	29.8
13	20	26.2
14	–	20.8
15	–	–

Table 10. B_{pl} for VBR and CBR WB Opus Under Random and Bursty Packet Loss

VBR B_{pl}			CBR B_{pl}		
Bitrate	Random	Bursty	Bitrate	Random	Bursty
11	23.9	20.1	11	–	–
12	22.2	18.9	12	24.8	20.8
13	19.5	16.9	13	23.0	19.4
14	–	–	14	17.6	15.8
15	–	–	15	18.7	16.1

A.3 NB

Table 11. $I_{e,NB}$ for VBR and CBR NB Opus Modes

Bitrate	VBR $I_{e,NB}$	CBR $I_{e,NB}$
6	23	46.3
7	21.3	31.7
8	16	19.2
9	11.7	13.2
10	–	6.2
11	–	2

Table 12. B_{pl} for VBR and CBR NB Opus Under Random and Bursty Packet Loss

VBR B_{pl}			CBR B_{pl}		
Bitrate	Random	Bursty	Bitrate	Random	Bursty
6	24.6	15.4	6	8.9	5.7
7	22.5	14.4	7	18.2	11.8
8	20.2	13.5	8	15.1	10.1
9	18.9	12.4	9	17.2	11.3
10	–	–	10	15.3	10.5
11	–	–	11	16	10.7

REFERENCES

- [1] Ismet Aktas, Florian Schmidt, Elias Weingärtner, Cai-Julian Schnelke, and Klaus Wehrle. 2012. An adaptive codec switching scheme for SIP-Based VoIP. In *Internet of Things, Smart Spaces, and Next Generation Networking*. Springer, 347–358.
- [2] Mohannad ALAhmadi, Peter Pocta, and Hugh Melvin. 2019. Instrumental estimation of e-model equipment impairment factor parameters for super-wideband Opus codec. In *Proceedings of the 2019 30th Irish Signals and Systems Conference (ISSC'19)*. IEEE, Los Alamitos, CA, 1–6.
- [3] Harald Alvestrand and Varun Singh. 2018. Identifiers for WebRTC's Statistics API. Retrieved September 15, 2021 from <https://www.w3.org/TR/2018/CR-webrtc-stats-20180703/>.
- [4] ITU-T P.501 amendment 1. 2017. Test signals for use in telephonometry. *ITU-T Recommendation* (2017).
- [5] Amit Ashara. 2016. *Implementing Opus Voice Code for TM4C129x device*. Application Report. Texas Instruments.
- [6] Haytham Assem, Mohamed Adel, Brendan Jennings, David Malone, Jonathan Dunne, and Pat O'Sullivan. 2013. A generic algorithm for mid-call audio codec switching. In *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM'13)*. IEEE, Los Alamitos, CA, 1276–1281.

- [7] Adam Bergkvist, Daniel Burnett, Cullen Jennings, Anant Narayanan, Bernard Aboba, Taylor Brandstetter, and Jan-ivar Bruaroey. 2018. WebRTC 1.0: Real-time Communication Between Browsers. Retrieved September 15, 2021 from <https://www.w3.org/TR/2018/CR-webrtc-20180927/>.
- [8] Stefan Bruhn, Harald Pöbloth, Markus Schnell, Bernhard Grill, Jon Gibbs, Lei Miao, Kari Järvinen, et al. 2015. Standardization of the new 3GPP EVS codec. In *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'15)*. IEEE, Los Alamitos, CA, 5703–5707.
- [9] Leandro Silva Galvão De Carvalho and Edjair De Souza Mota. 2013. Survey on application-layer mechanisms for speech quality adaptation in VoIP. *ACM Computing Surveys* 45, 3 (2013), 36.
- [10] Claudio Casetti, Juan Carlos De Martin, and Michela Meo. 2000. A framework for the analysis of adaptive voice over IP. In *Proceedings of the 2000 IEEE International Conference on Communications (ICC'00). Global Convergence Through Communications. Conference Record, Vol. 2*. IEEE, Los Alamitos, CA, 821–826.
- [11] Eirik Fosser and Lars Olav D. Nedberg. 2016. *Quality of Experience of WebRTC Based Video Communication*. Master's Thesis. NTNU.
- [12] ITU-T G.107. 2009. The E-model: A computational model for use in transmission planning. *ITU-T Recommendation* (2009).
- [13] ITU-T G.107.1. 2019. Wideband e-model. *ITU-T Recommendation* (2019).
- [14] ITU-T G.108. 1999. Application of the e-model: A planning guide. *ITU-T Recommendation* (1999).
- [15] ITU-T G.113. 2007. Transmission impairments due to speech processing. *ITU-T Recommendation* (2007).
- [16] ITU-T G.113.2. 2019. New appendix v—provisional planning values for the fullband equipment impairment factor and the fullband packet loss robustness factor. *ITU-T Recommendation* (2019).
- [17] Boni García, Micael Gallego, Francisco Gortázar, and Antonia Bertolino. 2018. Understanding and estimating quality of experience in WebRTC applications. *Computing* 101 (2018), 1585–1607.
- [18] Yi Han and Gabriel-Miro Muntean. 2015. Hybrid real-time quality assessment model for voice over IP. In *Proceedings of the 2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. IEEE, Los Alamitos, CA, 1–6.
- [19] Mark Handley, Van Jacobson, and Colin Perkins. 2006. SDP: Session description protocol. *IETF RFC4566* (2006).
- [20] Gerhard Haßlinger and Oliver Hohlfeld. 2008. The Gilbert-Elliott model for packet loss in real time services on the internet. In *Proceedings of the 14th GI/ITG Conference-Measurement, Modelling, and Evaluation of Computer and Communication Systems*. 1–15.
- [21] Stephen Hemminger. 2005. Network emulation with NetEm. In *Proceedings of the Linux Conference*. 18–23.
- [22] Jan Janssen, Danny De Vleeschauwer, Maarten Buchli, and Guido H. Petit. 2002. Assessing voice quality in packet-based telephony. *IEEE Internet Computing* 6, 3 (2002), 48–56.
- [23] Oliver Jokisch, Michael Maruschke, Martin Meszaros, and Viktor Iaroshenko. 2016. Audio and speech quality survey of the Opus codec in web real-time communication. In *Proceedings of the 27th Conference of Electronic Signal Processing (ESSV'16)*. 254–262.
- [24] Oliver Jokisch, Michael Maruschke, Martin Meszaros, and Viktor Iaroshenko. 2016. Audio and speech quality survey of the Opus codec in Web real-time communication. In *Proceedings of the 27th Conference of Electronic Signal Processing (ESSV'16)*. 254–262.
- [25] Tanu Kaskinen. 2019. PulseAudio Volume Control 4.0. Retrieved September 15, 2021 from <https://freedesktop.org/software/pulseaudio/pavucontrol/>.
- [26] Doh-Suk Kim and Ahmed Tarraf. 2007. ANIQUE+: A new American national standard for non-intrusive estimation of narrowband speech quality. *Bell Labs Technical Journal* 12, 1 (2007), 221–236.
- [27] Tashi Levent-Levi. 2019. Identifiers for WebRTC's Statistics API. Retrieved September 15, 2021 from <https://webrtcglossary.com/sdp-munging/>.
- [28] Michael Maruschke, Oliver Jokisch, Martin Meszaros, and Viktor Iaroshenko. 2015. Review of the Opus codec in a WebRTC scenario for audio and speech communication. In *Proceedings of the International Conference on Speech and Computer*. 348–355.
- [29] Michael Maruschke, Oliver Jokisch, Martin Meszaros, and Viktor Iaroshenko. 2015. Review of the Opus codec in a WebRTC scenario for audio and speech communication. In *Proceedings of the International Conference on Speech and Computer*. 348–355.
- [30] James William McGowan. 2005. Burst ratio: A measure of bursty loss on packet-based networks. US Patent 6,931,017.
- [31] Martin Meszaros, Franziska Trojahn, Michael Maruschke, and Oliver Jokisch. 2018. QuARTCS: A tool enabling end-to-any speech quality assessment of WebRTC-based calls. In *Proceedings of the International Conference on Speech and Computer*. 408–418.
- [32] Gabriel Mittag, Sebastian Möller, Vincent Barriac, and Stéphane Ragot. 2018. Quantifying quality degradation of the EVS super-wideband speech codec. In *Proceedings of the 2018 10th International Conference on Quality of Multimedia Experience (QoMEX'18)*. IEEE, Los Alamitos, CA, 1–6.

- [33] Is-Haka Mkwawa, Emmanuel Jammeh, Lingfen Sun, and Emmanuel Ifeakor. 2010. Feedback-free early VoIP quality adaptation scheme in next generation networks. In *Proceedings of the 2010 IEEE Global Telecommunications Conference (GLOBECOM'10)*. IEEE, Los Alamitos, CA, 1–5.
- [34] Sebastian Möller, Nicolas Côté, Valérie Gautier-Turbin, Nobuhiko Kitawaki, and Akira Takahashi. 2010. Instrumental estimation of E-model parameters for wideband speech codecs. *EURASIP Journal on Audio, Speech, and Music Processing* 2010, 1 (2010), 782731.
- [35] Sebastian Moller, Alexander Raake, Nobuhiko Kitawaki, Akira Takahashi, and Marcel Waltermann. 2006. Impairment factor framework for wide-band speech codecs. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 6 (2006), 1969–1976.
- [36] Suhas Nandakumar and Cullen Jennings. 2018. Annotated example SDP for WebRTC draft-ietf-rtcweb-sdp-11. *IETF* (2018).
- [37] See Leng Ng, Simon Hoh, and Devinder Singh. 2005. Effectiveness of adaptive codec switching VoIP application over heterogeneous networks. In *Proceedings of the 2005 2nd Asia Pacific Conference on Mobile Technology, Applications, and Systems*.
- [38] ITU-T P.563. 2004. Single-ended method for objective speech quality assessment in narrow-band telephony applications. *ITU-T Recommendation* (2004).
- [39] ITU-T P.800. 1996. Methods for subjective determination of transmission quality. *ITU-T Recommendation* (1996).
- [40] ITU-T P.833. 2001. Methodology for derivation of equipment impairment factors from subjective listening-only tests. *ITU-T Recommendation* (2001).
- [41] ITU-T P.833.1. 2009. Methodology for the derivation of equipment impairment factors from subjective listening-only tests for wideband speech codecs. *ITU-T Recommendation* (2009).
- [42] ITU-T P.834. 2015. Methodology for the derivation of equipment impairment factors from instrumental models. *ITU-T Recommendation* (2015).
- [43] ITU-T P.834.1. 2015. Extension of the methodology for the derivation of equipment impairment factors from instrumental models for wideband speech codecs. *ITU-T Recommendation* (2015).
- [44] ITU-T P.862. 2001. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *ITU-T Recommendation* (2001).
- [45] ITU-T P.862.3. 2007. Application guide for objective quality measurement based on recommendations P.862, P.862.1 and P.862.2. *ITU-T Recommendation* (2007).
- [46] ITU-T P.863. 2011. Perceptual objective listening quality assessment POLQA. *ITU-T Recommendation* (2011).
- [47] ITU-T P.863.1. 2019. Application guide for recommendation ITU-T P.863. *ITU-T Recommendation* (2019).
- [48] Alexander Raake, Sebastian Möller, Marcel Waltermann, Nicolas Cote, and J.-P. Ramirez. 2010. Parameter-based prediction of speech quality in listening context—Towards a WB e-model. In *Proceedings of the 2010 2nd International Workshop on Quality of Multimedia Experience (QoMEX'10)*. IEEE, Los Alamitos, CA, 182–187.
- [49] Abdel Rabassa, Marc St-Hilaire, Chung-Horng Lung, Ioannis Lambadaris, Nishith Goel, and Marzia Zaman. 2010. New speech traffic background simulation models for realistic VoIP network planning. In *Proceedings of the 2010 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'10)*. IEEE, Los Alamitos, CA, 364–371.
- [50] Anssi Rämö and Henri Toukoma. 2011. Voice quality characterization of IETF Opus codec. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*.
- [51] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan B. Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. 2002. SIP: Session initiation protocol. *IETF RFC3261* (2002).
- [52] Stefano Salsano, Fabio Ludovici, Alessandro Ordine, and D. Giannuzzi. 2012. Definition of a general and intuitive loss model for packet networks and its implementation in the Netem module in the Linux kernel. *University of Rome «Tor Vergata»*. Version 3 (2012).
- [53] ITU-T SG12-C.22. 2017. A subjective ACR LOT testing fullband speech coding and prediction by P.863. *CH-Geneva* (2017).
- [54] ITU-T SG12-C334. 2019. Instrumental estimation of e-model equipment impairment factor parameters for super-wideband Opus codec. *CH-Geneva* (2019).
- [55] ITU-T SG12-C335. 2019. Instrumental estimation of e-model equipment impairment factor parameters for wideband Opus codec. *CH-Geneva* (2019).
- [56] ITU-T SG12-C336. 2019. Instrumental estimation of e-model equipment impairment factor parameters for narrowband Opus codec. *CH-Geneva* (2019).
- [57] Mary Shaw and Beyond Objects. 1995. Software design paradigm based on process control. *ACM Software Engineering Notes* 20, 1 (1995), 27–39.
- [58] Julian Spittka, Koen Vos, and Jean-Marc Valin. 2015. RTP payload format for the Opus speech and audio codec. *IETF RFC7587* (2015).

- [59] Jean-Marc Valin, Timothy B. Terriberry, Gregory Maxwell, and Christopher Montgomery. 2010. Constrained-energy lapped transform (CELT) Codec draft-valin-celt-codec-02. *IETF* (2010).
- [60] Jean-Marc Valin, Koen Vos, and Timothy B. Terriberry. 2012. Definition of the Opus audio codec. *IETF RFC6716* (2012).
- [61] Martin Varela, Ian Marsh, and Björn Grönvall. 2006. A systematic study of PESQ's behavior (from a networking perspective). In *Proceedings of Measurement of Speech and Audio Quality in Networks* (2006).
- [62] Koen Vos, Soeren Skak Jensen, and Karsten Vandborg Soerensen. 2010. SILK speech codec draft-vos-silk-02. *IETF* (2010).
- [63] Marcel Wältermann, Izabela Tucker, Alexander Raake, and Sebastian Möller. 2010. Extension of the e-model towards super-wideband speech transmission. In *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, Los Alamitos, CA, 4654–4657.
- [64] Juejia Zhou, Xiaoming She, and Lan Chen. 2010. Source and channel coding adaptation for optimizing VoIP quality of experience in cellular systems. In *Proceedings of the 2010 IEEE Wireless Communication and Networking Conference*. IEEE, Los Alamitos, CA, 1–6.

Received September 2019; revised April 2020; accepted March 2021