Jonas Lemberg

Integrating Web-Based Multimedia Technologies with Operator Multimedia Services Core

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 19.6.2017

Thesis supervisor:

Prof. Raimo Kantola

Thesis advisor:

M.Sc. (Tech.) Pekka Mikkonen

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Jonas Lemberg

Title: Integrating Web-Based Multimedia Technologies with Operator Multimedia Services Core

Date: 19.6.2017 Language: English Number of pages: 9+63

Department of Communications and Networking

Professorship: Communications Engineering Code: ELEC3029

Supervisor: Prof. Raimo Kantola

Advisor: M.Sc. (Tech.) Pekka Mikkonen

During the last ten years, the volume of global mobile data traffic has been growing at an increasing pace. The latency and bandwidth of mobile data connections have improved significantly and the volume of mobile data traffic now greatly exceeds the volume of mobile voice traffic. It has therefore become more reasonable to discontinue the circuit switched voice telephony networks. Many operators are therefore deploying packet based core networks for handling voice and video calls. Simultaneously, web browsers are extended to handle both voice and video calls. By combining the properties of both the packet based core network and the web browsers connected to the Internet, operators can benefit from the large amount web browsers acting as potential endpoints in the network. Simultaneously operators have the possibility to e.g. charge for these calls and let customers call using a web browser instead of a mobile phone.

The purpose of this thesis is to evaluate whether the speech quality of this type of solutions is sufficient for commercial use. The speech quality of calls between a web browser and a mobile phone in a packet based network was objectively measured in various simulated environments. The speech quality of a temporary solution in which calls originate from a fully functional external packet based core network to the already existing circuit switched network was measured as well. The results show that speech quality in the packet based calls was superior compared to the circuit switched calls, while the speech quality of the calls breaking out to the circuit switched network was acceptable as well. The post dialing delay of the calls in the temporary implementation was unacceptably long, while the speech quality was on an acceptable level. Another significant finding was that transcoding is currently inevitable when using this technology for voice calls between web browsers and mobile phones.

Keywords: IMS, VoLTE, WebRTC, VoIP, MOS, PESQ, POLQA

AALTO-UNIVERSITETET
HÖGSKOLAN FÖR ELEKTROTEKNIK

SAMMANDRAG AV
DIPLOMARBETET

Utfört av: Jonas Lemberg

Arbetets namn: Integrating Web-Based Multimedia Technologies with Operator Multimedia Services Core

Datum: 19.6.2017 Språk: Engelska Sidantal: 9+63

Institutionen för kommunikations- och nätverksteknik

Huvudämne: Communications Engineering Kod: ELEC3029

Övervakare: Prof. Raimo Kantola

Handledare: DI Pekka Mikkonen

Under de senaste tio åren har volymen av global mobil datatrafik ökat i allt snabbare takt. Latensen och bandbredden på mobila bredbandsanslutningar har förbättrats avsevärt och volymen av mobil datatrafik överskrider nu kraftigt volymen av samtalstrafik. Det har därför blivit allt mer aktuellt att avskaffa de kretskopplade telefonnätverken. Därför installerar nu många operatörer paketbaserade telefonnätverk för hantering av ljud- och videosamtal. Samtidigt utökas webbläsare för att hantera både ljud- och videosamtal. Genom att kombinera egenskaperna av de paketbaserade telefonnätverket samt webbläsarna som är kopplade till internet, kan operatörer dra nytta av den stora mängd webbläsare som kan fungera som potentiella ändpunkter i nätverket. Samtidigt har operatörerna en möjlighet att t.ex. fakturera for dessa samtal och låta kunderna använda en webbläsare istället för en mobiltelefon.

Syftet med detta arbete är att uppskatta om ljudkvaliteten i denna typ av lösningar är tillräcklig för kommersiellt bruk. Ljudkvaliteten på samtal mellan en webbläsare och en mobiltelefon i ett paketbaserat nätverk mättes i diverse simulerade miljöer. Ljudkvaliteten på samtal i en tillfällig lösning där samtal från ett externt paketbaserat nätverk riktades till ett redan existerande kretskopplat telefonnätverk mättes också. Resultaten visar att ljudkvaliteten på samtalen i det paketbaserade nätverket var överlägsen jämfört med ljudkvaliteten på samtalen i det kretskopplade nätverket, medan ljudkvaliteten på samtalen i det kretskopplade nätverket var på acceptabel nivå. Fördröjningen efter uppringning var för lång i den temporära lösningen, medan ljudkvaliteten på samtalen var acceptabel. En annan väsentlig upptäckt var att transkodning för tillfället är oundvikligt då man använder denna teknik för ljudsamtal mellan webbläsare och mobiltelefoner.

Nyckelord: IMS, VoLTE, WebRTC, VoIP, MOS, PESQ, POLQA

Preface

I want to thank Professor Raimo Kantola for his advice and support during the writing process, and for encouraging me to investigate and learn about the benefits of end-to-end speech quality measurements.

I would also like to thank my advisor Pekka Mikkonen at Telia Finland who supported the writing process together with Harri Jokela and colleagues. I'm grateful for getting the opportunity to write my thesis for Telia Finland.

I also owe a special thanks to David Sjöstrand at Ericsson for the very valuable advice regarding speech quality measurements.

Espoo, 19.6.2017

Jonas Lemberg

Contents

Abstract **ii**

Abstract (in Swedish) **iii**

Preface **iv**

Contents **v**

Abbreviations **vii**

1 Introduction **1**
1.1 Research scope 2
1.2 Research questions 2
1.3 Methods 2
1.4 Structure 3
1.5 Related work 3

2 Background **5**
2.1 IP Multimedia Subsystem (IMS) 5
2.1.1 Protocols 5
2.1.2 Architecture 8
2.2 Web Real-Time Communication (WebRTC) 11
2.2.1 Protocols 11
2.2.2 Architecture 14
2.2.3 Browser support 15
2.3 WebRTC access to IMS 17
2.3.1 Architecture 17
2.3.2 Codec support 19
2.3.3 Codec interoperability 23
2.3.4 Codec comparison 24
2.3.5 Market overview 25
2.4 Summary 26

3 Implementation **27**
3.1 Architecture 27
3.1.1 Full IMS environment 27
3.1.2 Hybrid environment 29
3.2 Supported codecs 31

4 Measurements **32**
4.1 Measuring speech quality 32
4.1.1 Perceptual Evaluation of Speech Quality (PESQ) 32
4.1.2 Perceptual Objective Listening Quality Assessment (POLQA) 33
4.2 Measurement setup 35
4.2.1 Test scenarios 35

4.2.2 Recording setup 36
4.2.3 Calibration 38
4.2.4 Probe measurements 39
4.3 Measurement results 39
4.4 Measurement analysis 47
4.5 Summary 48

5 Discussion **50**

6 Conclusion **52**

References **53**

Appendices

A PESQ analysis results **60**

B POLQA analysis results **62**

Abbreviations

3GPP	3rd Generation Partnership Project
A/D	Analog-to-Digital
AAA	Authentication, Authorization, and Accounting
ACR	Absolute Category Rating
ALG	Application Level Gateway
AMR	Adaptive Multi-Rate
API	Application Programming Interface
AS	Application Server
AVC	Advanced Video Coding
BGF	Border Gateway Function
BGCF	Breakout Gateway Control Function
BSC	Base Station Controller
BSS	Base Station Subsystem
BSSAP	BSS Application Part
CBR	Constant Bit Rate
CN	Comfort Noise
CS	Circuit Switched
CSCF	Call Session Control Function
CSS	Cascading Style Sheets
D/A	Digital-to-Analog
DSL	Digital Subscriber Line
DTLS	Datagram Transport Layer Security
eIMS-AGW	IMS Access Gateway enhanced for WebRTC
eP-CSCF	Proxy CSCF enhanced for WebRTC
EVS	Enhanced Voice Services
FB	Fullband
GSM	Global System for Mobile communications
GSM-EFR	GSM Enhanced Full Rate Speech Codec
GW	Gateway
H/V-PCRF	Home/Visited PCRF
HEVC	High Efficiency Video Coding
HSS	Home Subscriber Server
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ICE	Interactive Connectivity Establishment
I-CSCF	Interrogating Call Session Control Function
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IP-CAN	IP Connectivity Access Network
ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
ITU	International Telecommunication Union

ITU-T	Telecommunication Standardization Sector of ITU
JS	JavaScript
LAN	Local Area Network
LTE	Long Term Evolution
MB	Mediumband
MGCF	Media Gateway Control Function
MGW	Media Gateway
MOS	Mean Opinion Score
MRFC	Multimedia Resource Function Controller
MRFP	Multimedia Resource Function Processor
MSS	Mobile Switching Station
MTI	Mandatory To Implement
MTSI	Multimedia Telephony Service for IMS
NAT	Network Address Translation
NB	Narrowband
OTT	Over The Top
P2P	Peer to Peer
PCEF	Policy and Charging Enforcement Function
PCM	Pulse Code Modulation
PCMA	PCM A-law
PCMU	PCM μ-law
PCRF	Policy and Charging Rules Function
PDD	Post Dialing delay
P-CSCF	Proxy CSCF
PESQ	Perceptual Evaluation of Speech Quality
PLC	Packet Loss Concealment
PLMN	Public Land Mobile Network
POLQA	Perceptual Objective Listening Quality Assessment
POTS	Plain Old Telephony System
PS	Packet Switched
PSQM	Perceptual Speech Quality Measure
PSTN	Public Switched Telephony Network
QoS	Quality of Service
RANAP	Radio Access Network Application Part
REST	Representational State Transfer
RFC	Request For Comments
RNC	Radio Network Controller
RTP	Real-Time Transport Protocol
RTCP	RTP Control Protocol
SBC	Session Border Controller
S-CSCF	Serving CSCF
SCTP	Stream Control Transmission Protocol
SDP	Session Description Protocol
SID	Silence Descriptor
SIP	Session Initiation Protocol

SIP-I	SIP with encapsulated ISUP
SIP-T	SIP for Telephones
SIPoWS	SIP over WebSocket
SLF	Subscription Locator Function
SMS	Short Message Service
SRTCP	Secure RTCP
SRTP	Secure RTP
STUN	Session Traversal Utilities for NAT
SWB	Super-Wideband
TCP	Transmission Control Protocol
TURN	Traversal Using Relays around NAT
UDP	User Datagram Protocol
UE	User Equipment
URI	Uniform Resource Identifier
VAD	Voice Activity Detection
VBR	Variable Bit Rate
W3C	World Wide Web Consortium
WAF	WebRTC Authorization Function
WB	Wideband
WCG	Web Communication Gateway
WebRTC	Web Real-Time Communication
WebSDK	Web Software Development Kit
WIC	WebRTC IMS Client
ViLTE	Video over LTE
VoIP	Voice over IP
VoLTE	Voice over LTE
VoWiFi	Voice over WiFi
WWSF	WebRTC Web Server Function
XMPP	Extensible Messaging and Presence Protocol

1 Introduction

During the last ten years the volume of global mobile data traffic has been growing at an increasing pace. By the end of 2009, the volume of mobile data traffic surpassed the volume of mobile voice traffic, and it now greatly exceeds the volume of mobile voice traffic [1]. The measurement results by Ericsson in Figure 1, illustrate this phenomenon.

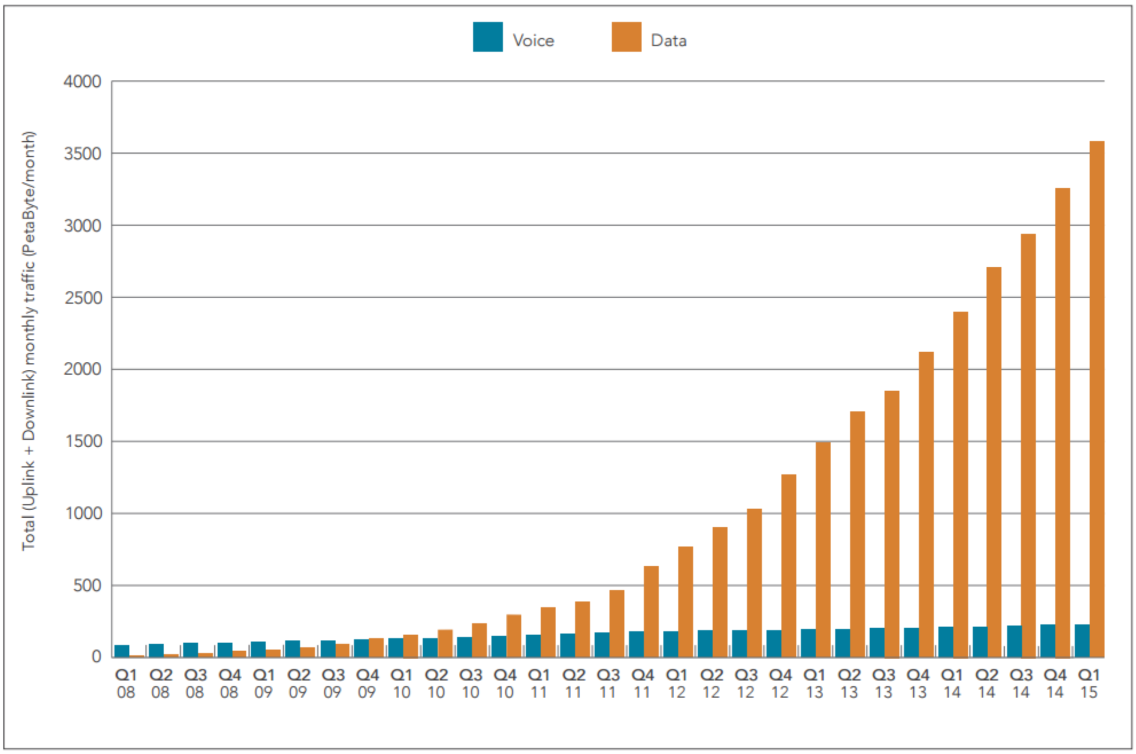

Figure 1: Total monthly global mobile voice and data traffic as measured by Ericsson [1]

One of the reasons for the fast growth is that the latency and bandwidth of mobile data connections have improved significantly during this period. It has therefore become more and more reasonable to discontinue the circuit switched voice telephony networks and start carrying voice over IP networks as packet based. There are both standard and proprietary technologies for Voice over IP (VoIP). Although proprietary, the first truly successful VoIP technology was Skype, which is still widely used today. For setting up and managing multimedia sessions over IP, the Internet Engineering Task Force (IETF) has developed the Session Initiation Protocol (SIP). By using SIP in a particular manner the 3rd Generation Partnership Project (3GPP) then developed the IP Multimedia Subsystem (IMS), which is a standard that aims to assist operators deploying packet based core networks. The latest trend in VoIP is full integration of voice and video in web browsers. As a result, web browsers are now being extended with built-in voice and video support. This technology is called Web Real-Time Communication (WebRTC). With Google being one of the main drivers behind the technology, WebRTC is supported by most browsers.

The major difference between the IMS and WebRTC VoIP approaches is that IMS aims to let mobile operators provide managed voice services in a similar way as classical circuit switched voice services are provided. The idea is to guarantee quality, provide emergency calls and carry the charging model of telephony to VoIP. WebRTC on the other hand will make voice just another Internet application among others, potentially leading to reduced voice revenues for the operators. Quality of voice in WebRTC is based on smart voice encoding and increased capacity of the IP networks, while no guarantees regarding Quality of Service (QoS) can be provided since the Internet works on a best effort model.

Many IMS vendors are developing so called WebRTC gateways (GW), which allow web browsers establish calls over the Internet to the IMS, using WebRTC technology. This enables Internet access to the IMS, allowing calls between regular mobile phones and web browsers. In this way, customers can benefit from the properties of the Internet integrated with traditional telephony services.

1.1 Research scope

There are various ways of using WebRTC technology in an operator network environment. Therefore, the scope of this thesis is limited to voice calls only. Furthermore, the speech quality will be measured end-to-end, one-way in two different environments. One of the environments is a fully functional IMS network, where high definition packet switched voice calls are possible with other clients supporting Multimedia Telephony Service for IMS (MTSI). MTSI refers to the telephony functionality in terminals connected to the IMS, such as mobile phones. The other is a temporary solution where the IMS network has been connected to a circuit switched mobile network, also referred to as Public Land Mobile Network (PLMN), not supporting high definition voice calls. The direction of the call in all the cases will be from the web browser to the mobile phone.

1.2 Research questions

This thesis answers the following questions:

– How do voice calls perform between a web browser and a mobile phone in a PLMN network, using an external IMS?

– How do voice calls perform between a web browser, and a mobile phone with MTSI capability in the same IMS network?

– Is the speech quality in the two environments acceptable for commercial use?

1.3 Methods

The solutions are implemented in a lab environment, after which they are tested and measured using objective intrusive end-to-end measurements methods, specified by the Telecommunication Standardization Sector (ITU-T) of the International Communication Union (ITU).Traffic is injected into the network and the perceived quality is then evaluated with the methods Perceptual Evaluation of Speech Quality (PESQ) and Perceptual Objective Listening Quality Assessment (POLQA). Interference is simulated by adding packet loss, ranging from 0 to 30%. A Mean Opinion Score (MOS) is then derived to represent the perceived speech quality.

1.4 Structure

The thesis is structured in the following way:

Background

This chapter describes the evolution of operator networks and current trends. It also describes technologies such as IMS and WebRTC. Additionally, this chapter describes how WebRTC technology can be utilized when interconnecting the Internet and an operator core network using a WebRTC gateway.

Implementation

This chapter describes how an end-to-end call is made using a web browser at one end and a mobile phone in the PLMN at the other end, utilizing WebRTC technology and a WebRTC gateway. The chapter also describes how end-to-end calls are made in full IMS environments, utilizing the same WebRTC gateway. A high-level design description of the test environments is provided.

Measurement and analysis

The measurement methods used in the thesis are presented. The MOS values of the calls are measured in different simulated network conditions to evaluate the usability of the solutions. This is done by adding packet loss with the tool Netem. The results are presented and analyzed.

Discussion and conclusions

The results are discussed, focusing especially on whether the implementations are suitable for commercial use. Future research areas are suggested as well.

1.5 Related work

In 2013, Boyan Tabakov published a thesis related to the Techno-Economic Feasibility of Web Real-Time Commununications. In his work he evaluated WebRTC from both a technical and economic perspective. He discovered that WebRTC is a techno-economically feasible technology, at least in its basic one-to-one browser-to browser communication on desktop computers. WebRTC is according to Tabakov a natural step in the evolution of the Web. He also pointed out a few challenges related to this technology, such as Apple not supporting WebRTC at the time of writing the thesis [2]. This is still the situation, although Apple has announced they are adding WebRTC support to their WebKit-based browser Safari [3].

In 2016, Chuan Wang published a thesis on Parametric Methods for Determining the Media Plane Quality of Service. The thesis compared the developed parametric method for predicting the speech quality in an IMS environment to results obtained with the methods PESQ and POLQA, which are the same methods that are used in this thesis. The thesis underlines the point that while intrusive objective measurements methods such as PESQ and POLQA are suitable for verifying that an environment provides sufficient speech quality, a parametric method is needed for reliable, flexible and convenient estimation of speech quality in live networks [4].

2 Background

Operators are moving towards packet-based networks, leaving the world of circuit switched calls behind. In order to do this transformation, operators are currently deploying IP Multimedia Subsystems, IMS, to their core network for handling all multimedia sessions, such as voice and video calls [5][6].

Simultaneously, operators are trying to keep up the pace with so called Over The Top (OTT) service providers, such as Facebook and WhatsApp, not to become only Internet service providers for their customers. They are losing call and SMS revenues as third-party vendors are creating content for their networks [7]. Since the mobile Internet connections are becoming increasingly faster, many OTT service providers can offer their customers high-quality calls and messaging services for free. As many operators are moving towards IP based networks, integrating multimedia services such as WebRTC with IMS could be an effective combination to slow down this trend.

OTT service providers do not primarily support cross-platform communication: you can't make a WhatsApp call to a person not having WhatsApp, at least not without paying extra for the breakout feature. Operators have the possibility to offer native calling features over Internet connections, by offering Voice over LTE (VoLTE) and Voice over WiFi (VoWiFi) calling or allowing calls from e.g. web browsers, without requiring the users to install any applications. This is a clear advantage for the operators, and deploying the IMS makes this possible. Another advantage is the end-to-end QoS that IMS can provide, while the Internet works on a best effort basis. In the IMS, the needed resources are reserved according to the users' requirements when negotiating for setting up a multimedia session [8].

2.1 IP Multimedia Subsystem (IMS)

The IMS is an all-IP system designed by the 3GPP. It is a general architecture standard with the purpose to guide mobile network operators in their transition from circuit switched (CS) to packet switched core networks. The IMS architecture is designed to deliver the same kind of flexibility as the Internet [9]. In the book "The IMS : IP multimedia concepts and services", by Miikka Poikselkä et al., the IMS is defined as following:

"IMS is a global, access-independent and standard-based IP connectivity and service control architecture that enables various types of multimedia services to end-users using common Internet-based protocols."

The first release was in 1999, and the standardization is still ongoing. There are various IMS manufacturers, such as Nokia, Ericsson and Huawei [10][11][12].

2.1.1 Protocols

To ensure maximum reusability of Internet standards, 3GPP has decided to use SIP as the standard signaling mechanism for the IMS [9]. SIP is an application layer signaling protocol standardized by the IETF in 1999 [13]. It is used for setting up, modifying and terminating communication sessions, such as Internet telephone calls, multimedia distribution and multimedia conferences. It can be used for both peer-to-peer and multiparty communication. The SIP protocol includes methods for agreeing on compatible media types, routing requests to the correct locations, authentication and authorization [14]. A user is typically identified by its SIP Uniform Resource Identifier or SIP URI, e.g. sip:user@domain.com. SIP is designed to be independent of the underlying transport protocol, which means SIP works over the User Datagram Protocol, Transmission Control Protocol (TCP) and even the Stream Control Transmission Protocol (SCTP) [15].

There are both SIP methods and response codes. In the example in Figure 2, the SIP methods are marked with bold text while the responses are tagged with their corresponding response code number.

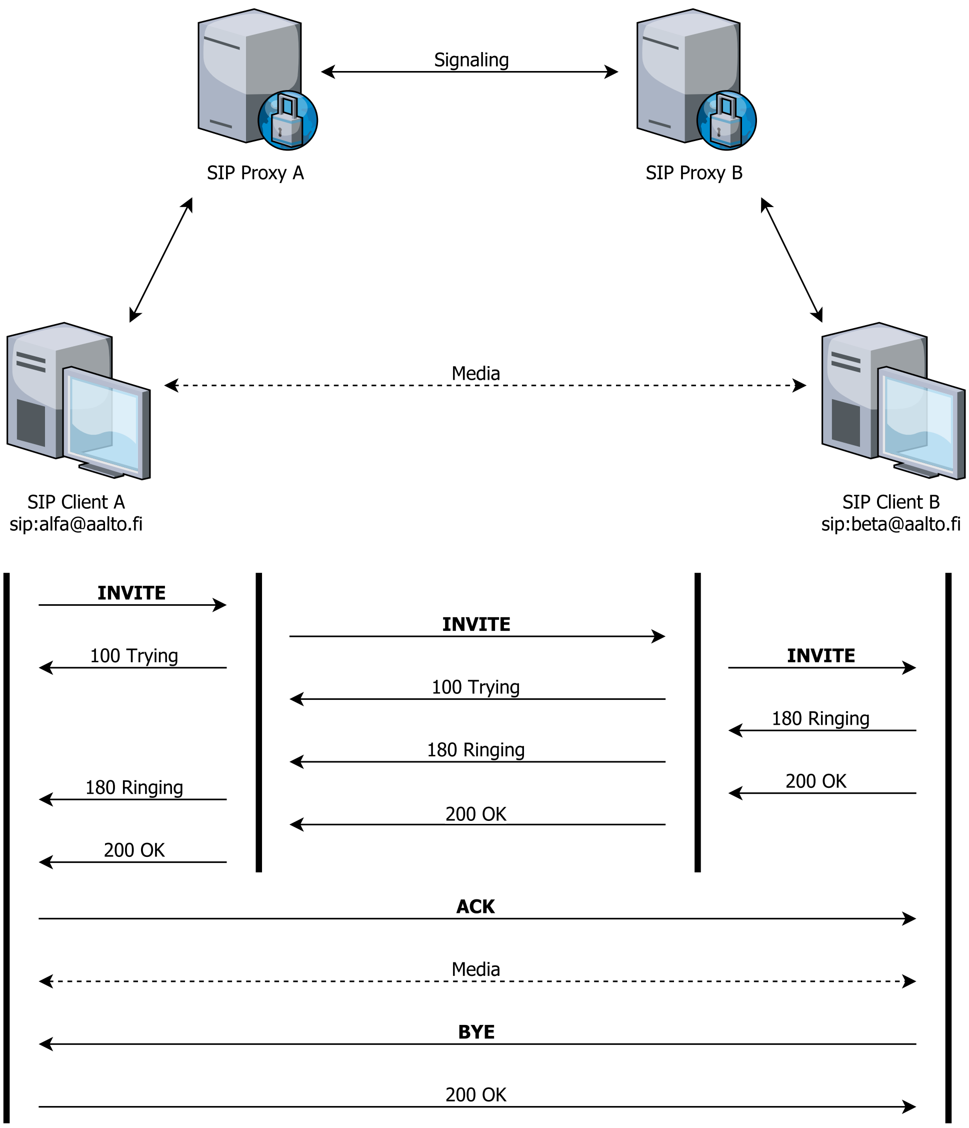

Figure 2: Setting up and terminating a multimedia communication session using the SIP protocol [14]

There are two variations of SIP called SIP for Telephony (SIP-T) and SIP with encapsulated ISUP (SIP-I), ISUP stands for ISDN User Part and ISDN stands for Integrated Services Digital Network. SIP-I is commonly used in legacy CS networks.

SDP

The Session Description Protocol or SDP is a text-based application layer protocol designed for describing multimedia sessions, typically used on top of the SIP protocol. The entity that wants to start the multimedia session, will first send a SDP offer, inside the body of the SIP method INVITE in Figure 2. The offer contains the description of supported media streams and codecs, as well as the IP address and port on which the entity would like to receive the media of the other party. The other party then answers with a SDP answer, which is inside the SIP response code 180 Ringing, indicating whether the media stream is accepted, which codec will be used and on which IP address and port the other party would like to receive the media. When a common media format has been found, the session can be set up. In case e.g. the media format must be changed, SDP can also be used for modifying sessions although the multimedia session is ongoing [8]. The SDP protocol was originally standardized by the IETF in 1998 as RFC2327, which was obsoleted by the present version RFC4566 in 2006 [16].

RTP

When the session has been set up, the actual media stream is then transported over UDP by using the Real-time Transport Protocol (RTP) which is designed for delivering real-time data end-to-end. RTP includes codec identification, sequence numbering, time stamping and delivery monitoring. This allows e.g. identifying reordered packets and estimation of end-to-end latency. QoS monitoring is handled separately by the RTP Control Protocol (RTCP) [8]. When using RTP over the Internet, the stream is often encrypted by using the Secure RTP (SRTP) framework, defined in RFC3711. The SRTP framework also includes support for Secure RTCP (SRTCP).

Diameter

The Diameter protocol is one of the two core protocols used in the IMS, in addition to SIP. The Diameter base protocol provides an Authentication, Authorization, and Accounting (AAA) framework for various applications, such as allowing network access or IP mobility in both local networks and when roaming. The protocol is not bound to a certain application, but instead it focuses on message exchanging functions [17]. The Diameter protocol was standardized by the IETF in 2003 as RFC3588, which was superseded by RFC6733 in 2012 [18].

H.248

The H.248 media gateway control protocol or Megaco is typically used between the Media Gateway Controllers (MGC) and Media Gateways (MGW) to handle signaling and session management during multimedia sessions [8]. The protocol was originally standardized by ITU-T in 2000. The most recent recommendation is ITU-T H.248.1 v3, released in 2013 [19].

2.1.2 Architecture

The 3GPP uses a layered approach to the architectural design, which means that the transport layer is separated from the signaling or control layer. On top of the control layer is the application layer, which provides the services in the IMS.

Application layer
The application or service layer, where application servers (AS) provide the actual services in the IMS.

Control layer
The control or signaling layer, where e.g. decisions whether a user can use a certain service or not are made. This is where sessions are set up and terminated, and where charging functions and other policy related functions are located. Components such as the Call Session Control Functions (CSCF), including Interrogating CSCF (I-CSCF), Serving CSCF (S-CSCF), Proxy CSCF (P-CSCF), the main databases Home Subscriber Server (HSS), Subscription Locator Function (SLF) and other control functions such as Interconnection Border Control Function (IBCF), Breakout Gateway Control Function (BGCF), Media Gateway Control Function (MGCF), Media Resource Broker (MRB) and Multimedia Resource Function Controller (MRFC) are located in this layer.

Transport layer
The transport or media layer, where the actual content of interactions between users is being transported. The media gateways are located in this layer, including the Transit Gateway (TrGW), IP Multimedia Media Gateway (IM-MGW), Multimedia Resource Function Processor (MRFP) and IMS Access Gateway (IMS-AGW).

In the reference architecture by the 3GPP in Figure 3, the network functions are separated into different logical network elements. In reality, the network functions are often combined into the same network element by the vendors implementing the standards. Vendors can combine different network elements in various ways, and those combinations of network elements can be named differently by each vendor. For clarity, only the names used in the standards will be used in the description of the general IMS architecture.

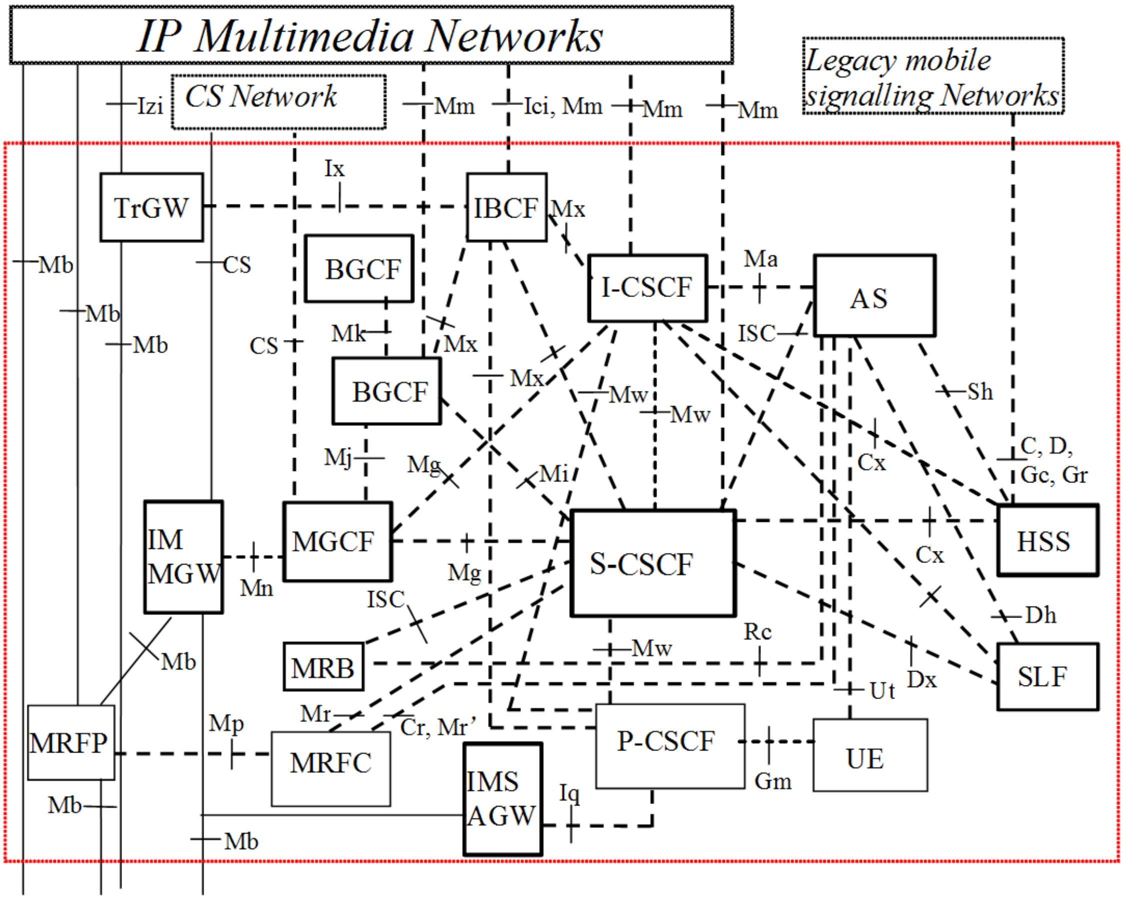

Figure 3: Reference architecture of the IMS, as specified by the 3GPP [20]

Main components [8]

Application Server (AS)
The actual services provided to the end users are located on the application servers. These include functions such as calling, messaging and voice mail.

Call Session Control Function (CSCF)
There are three different Call Session Control Functions (CSCF): Interrogating CSCF, Serving CSCF and Proxy CSCF. They all play an important role during registration and session establishment. They have both separate and shared functions. For instance, both the P-CSCF and S-CSCFs can look into the content of a SDP offer to see whether it contains media allowed for that particular user. If the media isn't allowed, the request is rejected and a SIP error message is sent to the user.

Proxy Call Session Control Function (P-CSCF)
The Proxy CSCF is the first point of contact when accessing the IMS, as this is where the SIP messages from the user are received. The end users also receive

the SIP messages from the P-CSCF. The P-CSCF can e.g. compress the SIP messages before they are sent to the end-user, if needed. The P-CSCF also includes emergency session detection [8].

Interrogating Call Session Control Function (I-CSCF)
The Interrogating CSCF is the contact point for all connections destined to subscribers in that particular network. The I-CSCF asks the Home Subscriber Server (HSS) for the next hop, which is typically either the Serving CSCF or an AS.

Serving Call Session Control Function (S-CSCF)
The Serving CSCF is a central component in the IMS since it handles registration processes, makes routing decisions and maintains session states and sets the service profiles. When a user initializes a registration request, it is routed to the S-CSCF and verified against data from the HSS. Based on this information, the S-CSCF creates a challenge to the end user. If the answer is correct, the S-CSCF starts the registration and the user is authorized to use IMS services. The S-CSCF then retrieves a service profile from the HSS as a part of the registration process.

Home Subscriber Server (HSS)
The HSS is the central database for all subscriber and service-related data in the IMS. In the HSS, a user-specific profile is permanently stored for each unique user. Data related to a specific user can then be requested from the HSS database if needed, e.g. authentication data can be requested by the S-CSCF when a user registers. The HSS also contains a service profile for each user, describing what services that particular user is allowed to use.

Breakout Gateway Control Function (BGCF)
If the S-CSCF decides that a breakout to the CS core network is needed, the S-CSCF contacts the BGCF. If the breakout should occur in the same network in which the BGCF is located, the BGCF selects a Media Gateway Control Function (MGCF) that will handle the call further. If the breakout should happen in another network, the session is forwarded to another BGCF in the selected network.

Media Gateway Control Function (MGCF)
The MGCF handles breakouts to the CS network, and is responsible for standard SIP to SIP-I conversion and for signaling between the CS network and the I-CSCF. The MGCF controls the IP Multimedia Media Gateway (IM-MGW).

IP Multimedia Media Gateway (IM-MGW)
The IM-MGW is the link handling the media between the CS network and the IMS. The MGCF reserves the needed resources from the IM-MGW for handling sessions. The IM-MGW can perform transcoding and generate tones or announcements to users in the CS network if needed.

IMS Access Gateway (IMS-AGW)
Controlled by the P-CSCF, the IMS-AGW handles the media streams between the User Equipment (UE) and the IMS domain. The IMS-AGW is responsible for security, encryption and transcoding [21].

2.2 Web Real-Time Communication (WebRTC)

In May 2011, Google released an open source software package for browser based real-time voice and video communication called WebRTC. Their intention was not only to integrate the functionality into their own browser Chrome, but to create a standard method of communicating over the web supported by all browsers, through a set of standard Application Programming Interfaces (API) [22].

Five years later, WebRTC is supported by major browsers such as Chrome, Firefox, Edge and Opera [23]. This means that there are over 2 billion WebRTC-capable browsers available in the world. Apple is currently developing WebRTC support for Safari [3]. Many companies have adopted the technology for their applications, such as Facebook Messenger, Google Hangouts and Snapchat [24]. The greatest difference to other VoIP technologies is that WebRTC works natively in most browsers without any additional plugins or installed applications. Instead, the needed functionality is offered though a web application typically consisting of Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS).

WebRTC is based on format and protocol specifications defined by the IETF and JavaScript API specifications defined by the World Wide Web Consortium (W3C). The goal of these two specifications is to define an environment where Peer to Peer (P2P) communication services, i.e. transferring messages, audio-video conversation and file transfer is supported. This is typically done with Internet browsers as endpoints [25].

2.2.1 Protocols

Signaling in WebRTC is not defined by the IETF, and it is outside the scope of their specification. This does not mean WebRTC works without signaling. It is up to the implementing entity to decide upon signaling. Figure 4 illustrates a commonly referred model of WebRTC deployment. The media itself is typically transported over a P2P connection between the two parties, while the signaling is handled separately. Commonly referred signaling models are standardized methods such as SIP or Extensible Messaging and Presence Protocol (XMPP) over WebSockets.

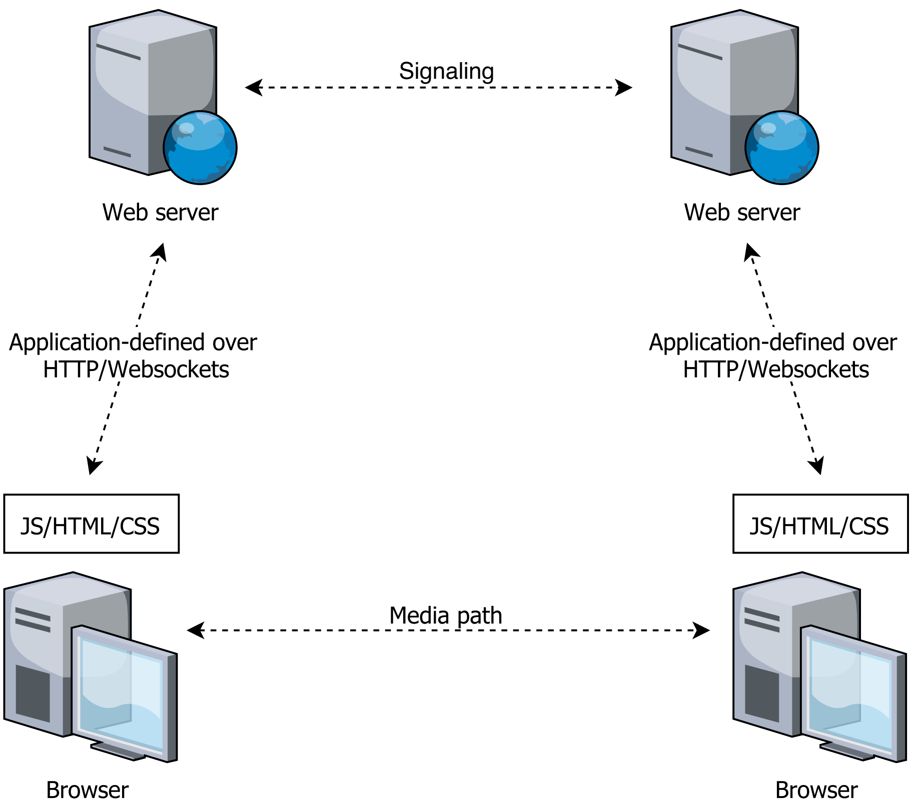

Figure 4: A commonly referred model of WebRTC deployment [26]

Interactive Connectivity Establishment (ICE)

Network Address Translation (NAT) is a commonly used method introduced partly to slow down IPv4 address depletion. The idea of NAT is that many devices use the same public IP address, but the traffic is separated between the different devices by using dedicated ports and port forwarding. NAT is unfortunately problematic when establishing P2P connections, since the clients might not know their public IP addresses when establishing the session [27]. A protocol built to address this problem is called Interactive Connectivity Establishment (ICE), and is often used when establishing WebRTC sessions. Many potential clients might be behind NATs and firewalls, hiding the real IP addresses of the clients or blocking P2P connections. In these cases, Session Traversal Utilities for NAT (STUN) or Traversal Using Relay NAT servers (TURN) are used. Both the STUN and TURN servers are publicly accessible servers, so even though the clients are behind NATs and firewalls, the servers are still accessible. In Figure 5, the components involved in the protocol are illustrated.

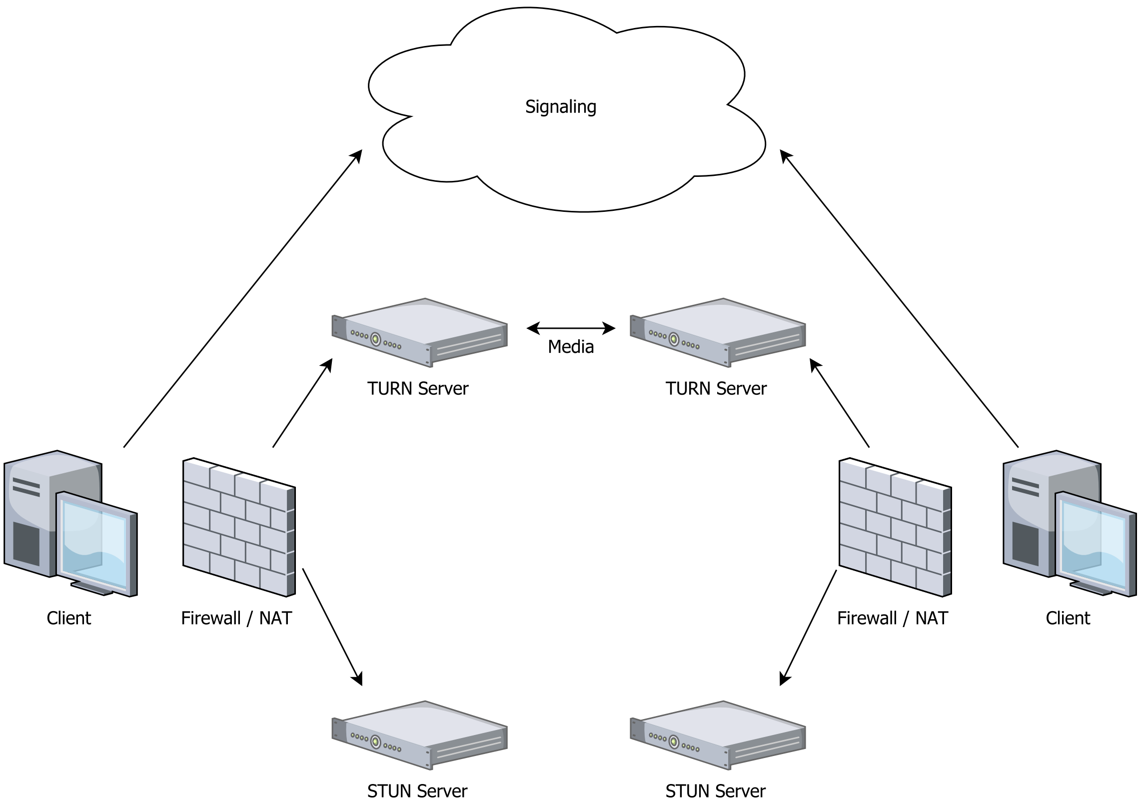

Figure 5: Establishing a WebRTC session utilizing the ICE protocol [28]

When establishing a WebRTC session, the ICE framework checks in the following order whether the session can be established using:

1. The IP addresses known by the clients themselves

2. The IP addresses and ports provided by the STUN server

   When a client contacts the STUN server, the server replies with a message containing the originating IP address and port number of the request. The client thus knows its public IP and port number, and can provide that information to the other client in order to establish the P2P connection.

3. A TURN server as a relay for transferring the media between the peers

   If establishing a session using IP addresses known by the clients or the IP addresses and port numbers provided by the STUN is unsuccessful, the TURN relay server is taken into use [29]. The TURN server works as a media relay server between the clients, when WebRTC sessions need to be established without P2P connections. The TURN server acts as a relay for the media, but not the signaling data.

2.2.2 Architecture

There are two distinct layers in the WebRTC architecture [30]:

The WebRTC C++ API
This layer is more relevant for developers of web browsers, who are interested in incorporating full WebRTC functionality in web browsers.

The WebRTC Web API
This layer is relevant for web developers, who want to create applications and call the API with various functions defined by the W3C [25].

As can be seen in Figure 6, most functionality is hidden from web developers. This makes it easy for web developers to develop applications that utilize WebRTC. It is also common that different Web Software Development Kits (WebSDK), typically JavaScript libraries are built on top of the WebRTC API to simplify the process of using the WebRTC API [31].

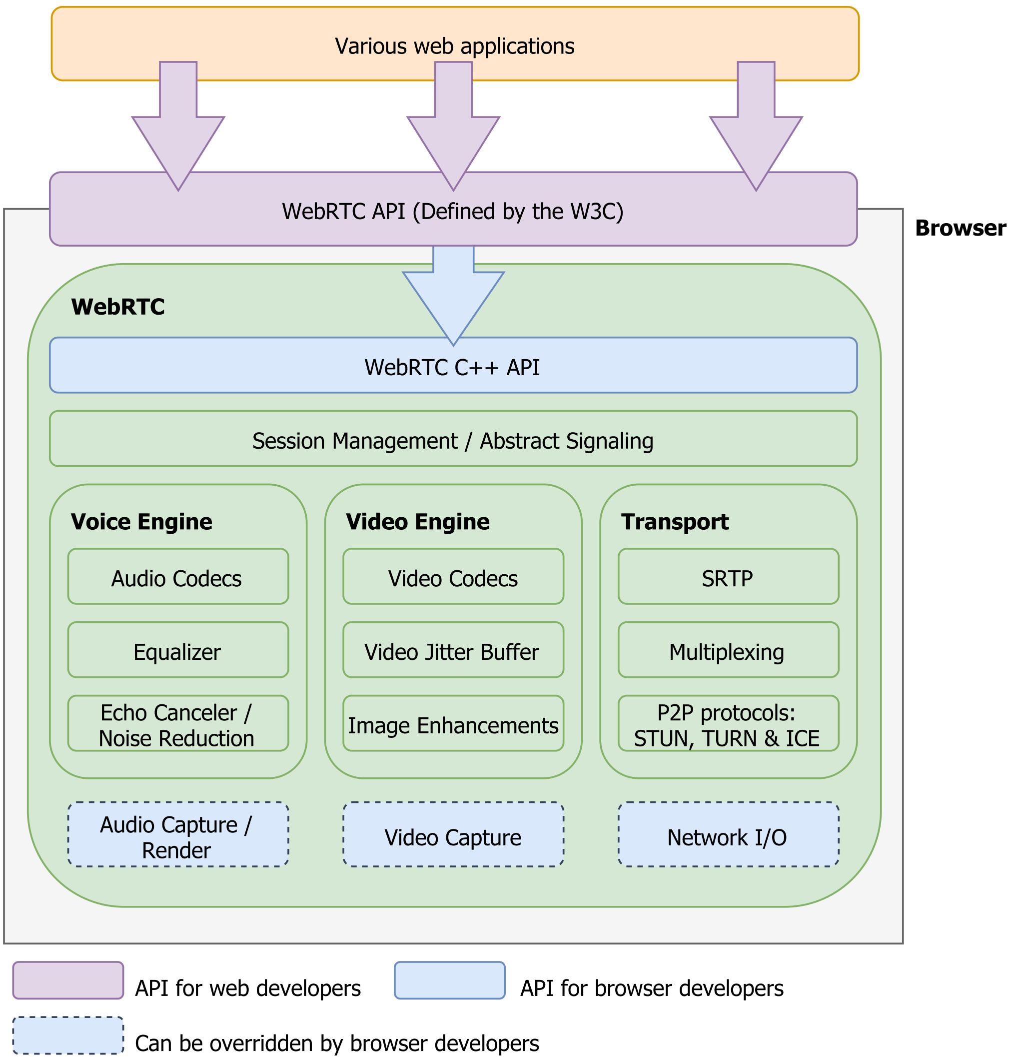

Figure 6: A high-level architectural view of WebRTC [30]

Main components

Transport
Contains the network stack for SRTP, as well as support for the protocols STUN, TURN and ICE. Session Management is an abstract session layer, handling call setup and management. It is up to the application developer to implement the call setup and management layer.

Voice Engine
A framework for the audio media chain, from the end users sound card to the network. Support for various codecs is included, which are described later in this thesis. This component also includes an equalizer for voice, an acoustic echo canceler and noise reduction.

Video Engine
Responsible for the video media chain, from the camera to the network. This component also contains support for video codecs, jitter buffer and image enhancement like removing noise caused by e.g. the web cam.

2.2.3 Browser support

In Figure 7, the current situation regarding WebRTC support in web browsers is illustrated. WebRTC is currently supported by most browsers, including Canary, Chrome, Opera, Nightly, Firefox, Edge and Bowser. Safari and Internet Explorer currently have no support for WebRTC.

This doesn't mean that devices by Apple can't support WebRTC. A common solution is building native applications not requiring a compatible browser, while still acting as a WebRTC endpoint. Bowser, which is a browser originally developed by Ericsson, runs on iOS devices and supports WebRTC as well [32].

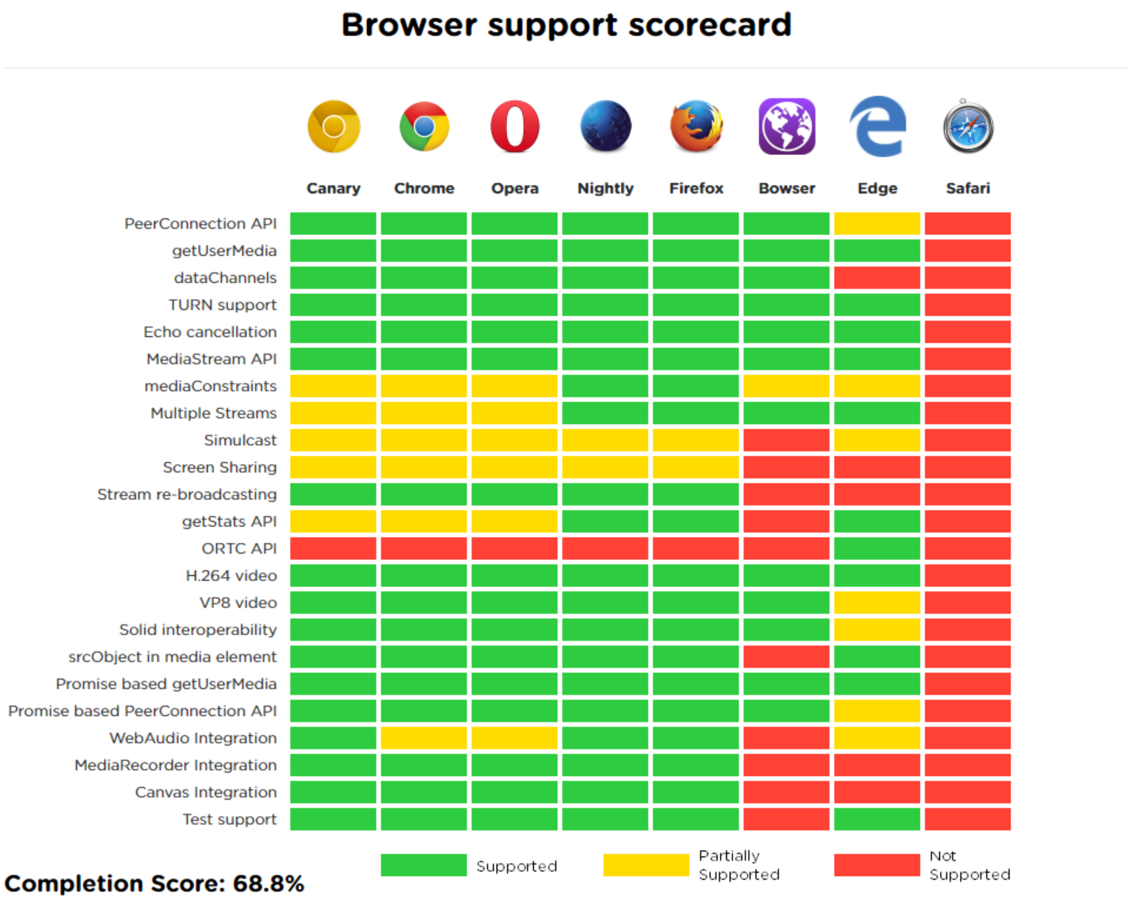

Figure 7: Major browsers support most functions of WebRTC [23]

2.3 WebRTC access to IMS

3GPP has defined a network based architecture model to support WebRTC client access to IMS. This means that a user could access IMS services by using e.g. a web application offered by the operator and running that from a WebRTC-compatible browser. This significantly increases the pool of clients able to access the IMS [33].

2.3.1 Architecture

The architecture and reference model of WebRTC access to IMS, as defined by the 3GPP is described in Figure 8.

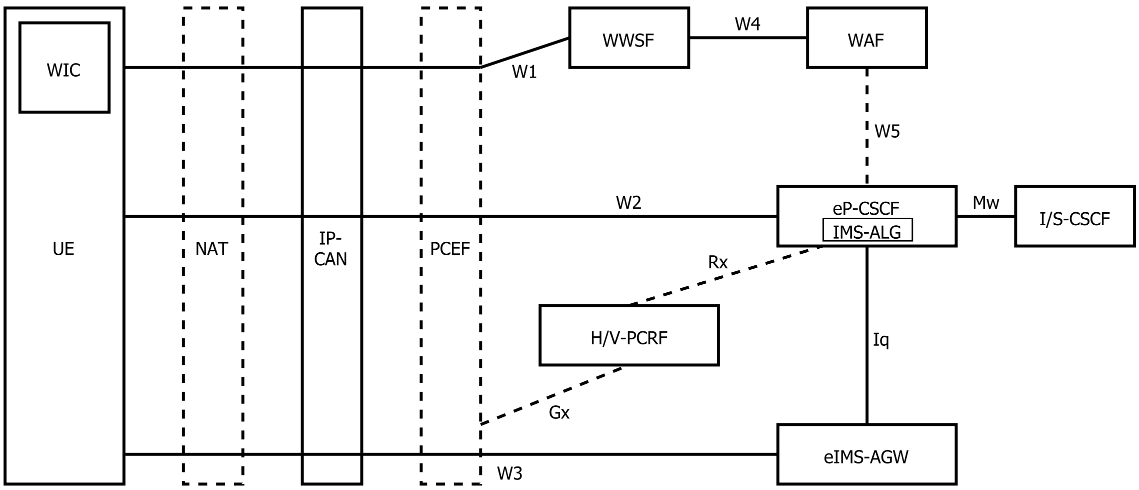

Figure 8: WebRTC IMS architecture and reference model as defined by 3GPP [20]. Note: The objects with dashed borders are optional.

Main components

WebRTC IMS Client (WIC)
A WebRTC application is built according to the instructions in the WebRTC 1.0 document specified by W3C. This is typically a web-application consisting of HTML, CSS and JavaScript. The client is downloaded from the WebRTC Web Server Function (WWSF) to the user equipment and calls the WebRTC API from there.

WebRTC Web Server Function (WWSF)
A web server hosting the potentially browser-based WebRTC IMS Client (WIC). This is from where the WIC is downloaded by the end user, and is the entry page and the initial point of contact to the network. The server can be located either in the operator network or in a third party network, and can also be split across many servers if needed. The WWSF can include the WebRTC Authorization Function (WAF) if located in the same domain.

WebRTC Authorization Function (WAF)
This component is typically included in the WWSF, and handles the authentication of the user. The WAF can be located either in the operator domain or a third party domain.

User Equipment (UE)
The device the WIC will eventually run on, in other words the customer's device. This can be e.g. a mobile phone, computer or smart TV. Since WebRTC runs in the browser, no additional software is required.

IP Connectivity Access Network (IP-CAN)
The IP network used to access the IMS core from the user equipment. When referring to Internet access, the 3GPP refers to any Internet access channel as IP-CAN, which stands for IP Connectivity Access Network. This can in practice be any Internet access method, such as Wireless Local Area Network (WLAN), fixed LAN, 4G or 3G.

P-CSCF enhanced for WebRTC (eP-CSCF)
The point of entry of SIP requests to the IMS system from the WebRTC client. 3GPP recommends SIP over WebSocket (SIPoWS) as the signaling method, defined in RFC 7118 [34], but other signaling methods can be used as well. This is however the point where the signaling used by the end user is converted to standard SIP used in the IMS [35]. The eP-CSCF needs to support at least one WebRTC IMS client-to-network signaling protocol.

IMS-AGW enhanced for WebRTC (eIMS-AGW)
The component the WIC connects to, typically using SRTP to deliver the WebRTC media as defined by the IETF, to the IMS. This is typically also where the transcoding of audio and video is carried out if needed. The eIMS-AGW shall also support NAT traversal using methods such as ICE.

Home/Visited Policy and Charging Rules Function (H/V-PCRF)
The Policy and Charging Rules Function makes policy and charging control decisions based on information obtained from the P-CSCF.

Signaling in Figure 8 explained

W1
The end user downloads the WebRTC client from the web server, typically over Hypertext Transfer Protocol Secure (HTTPS).

W2
The signaling between the WebRTC client and the eP-CSCF. This is typically SIP over WebSocket. Other signaling methods are allowed as well.

W3
The media stream between the end user and the IMS, typically SRTP. This is also referred to as the media plane.

W4
The signaling interface between the WAF and the WWSF.

W5
A non-mandatory signaling interface between the WAF and the eP-CSCF, not specified in the standard as it is implementation specific.

Gx
Exchanges information related to decisions regarding policies, using the Diameter protocol.

Rx
Exchanges information related to policy and charging, using the Diameter protocol.

Mw
Exchanges information between the two CSCF's, using the SIP protocol.

Iq
Exchanges information to the eIMS-AGW related to the media traffic from the WIC, typically using the H.248 protocol.

In commercial solutions, many of the functions listed are typically bundled into one product named differently by each vendor, commonly referred as a WebRTC gateway. Nokia is currently marketing this functionality as a Media Gateway (MGW) [36], while Ericsson is marketing a similar product called Web Communication Gateway (WCG) [37]. The WebRTC functionality for the P-CSCF is typically described as a separate network element. This functionality is referred to as a WebRTC gateway (GW) in this thesis.

2.3.2 Codec support

In order to digitally transmit analog signals such as audio, codecs are required for coding and decoding the signals from analog to digital and vice versa. When a signal is digitalized, a sample of the analog signal is taken at a frequency which is defined as the sampling frequency or rate of the encoder. If the sampling rate of the encoder is e.g. 8000 Hz, a sample of the signal is taken 8000 times every second. The voltage during that particular moment is then converted into a discrete value within the range of the codec resolution. If the resolution of the codec is e.g. 8 bits, there are $2^8 = 256$ different values available for representing the signal voltage digitally. Without any compression the bit rate of this codec would in this case be 8 kHz $\times$ 8 bits = 64 kbit/s.

According to Nyqvist's theorem, the maximum frequency that can be represented in the signal after sampling and decoding is half of the sampling rate used. For this reason, when using low sample rates the signal is typically sent through a low-pass filter before sampling [38]. The Analog-to-Digital (A/D) and Digital-to-Analog (D/A) conversion is naturally a mandatory process no matter what codec is being used. What differentiates codecs from each other is how the signal is processed further after being digitalized, e.g. how the signal is compressed, how the codec handles varying bandwidth and how errors like delayed or corrupt packets are handled when the signal is decoded.

The codecs Adaptive Multi-Rate (AMR), AMR-Wideband (AMR-WB) and (Enhanced Voice Service) EVS are currently specified by the 3GPP as supported audio codecs for MTSI clients in terminals such as mobile phones, in the Technical Specification (TS) 26.114 on Media handling and interaction in IMS [39]. Support for the AMR codec is mandatory, while the codecs AMR-WB and Enhanced Voice Services (EVS) should be supported if the Multimedia Telephony Service for IMS (MTSI) client supports wide (WB), super-wide (SWB) and fullband (FB) speech communication. MTSI clients supporting video communication must support the Advanced Video Coding H.264 (AVC) and should support High Efficiency Video Coding H.265 (HEVC) as well.

IETF has defined two Mandatory To Implement (MTI) codecs for both video and audio that WebRTC endpoints must support. For video, support for codecs VP8 and H.264 (AVC) is mandatory. For audio, Opus is the primary audio codec while G.711 is considered a fallback codec. Both codecs have to be supported in WebRTC endpoints [40].

Additionally, the IETF recommends that the audio codecs AMR, AMR-WB and G.722 should be supported in WebRTC endpoints to avoid transcoding when establishing sessions with e.g. mobile phones [41]. Currently, support for G.722 is implemented in Google Chrome [42], Mozilla Firefox [43] and Microsoft Edge [44]. Opus [45], G.711 [46] and G.722 [47] are royalty-free codecs, while AMR, AMR-WB and EVS are not [48]. It is therefore probable that AMR, AMR-WB or EVS codec support will not be added to browsers in the near future.

Packet Loss Concealment (PLC)

In real-time communication, resending lost packets is generally speaking not feasible since it would typically result in unacceptably long end-to-end delays. This is one of the reasons why real-time communication media is transmitted using the UDP protocol and not the TCP protocol. Since there is no resending of packets, packet loss can occur. Packet Loss Concealment (PLC) is a commonly used technique for masking or hiding errors caused by corrupt, late or lost packets. There are various PLC techniques, such as repeating the already received speech frames or modeling speech to fill the gaps in the packet stream [49]. When referring to PLC in this thesis, it refers to the PLC technique of the codec currently discussed.

Pulse Code Modulation (G.711)

Pulse Code Modulation (PCM) is the classical digital voice encoding method used in Plain Old Telephony Systems (POTS). G.711 is often also referred to as PCM. Standardized in 1972 by ITU-T, G.711 is still a widely used codec and one of the MTI codecs in WebRTC. There are two different versions of the G.711 codec, A-law and μ-law. μ-law is used in Japan and North America, while A-law is used in Europe and the rest of the world. The G.711 A-law is often referred to as PCMA8000, while the μ-law is referred to as PCMU8000. G.711 has a sampling rate of 8 kHz, and a frequency range from 300 - 3400 Hz. The bit rate of the codec is static, 64 kbit/s [50]. Packet Loss Concealment (PLC) functionality was added to G.711 in 1999, in the G.711 Appendix I: A high quality low-complexity algorithm for packet loss concealment with G.711 [51].

G.722

G.722 is a wideband codec standardized by the ITU-T in 1988. It has a bit rate from 48 to 64 kbit/s and a sampling rate of 16 kHz. The frequency range of the codec is 50 - 7000 Hz [52]. PLC support was added to the codec in 2006, in the G.722 Appendix III: A high-quality packet loss concealment algorithm for G.722 [53].

Adaptive Multi-Rate (AMR)

Specified by the 3GPP in 1999, the Adaptive Multi-Rate (AMR) speech codec is still a very common codec used in mobile networks today. The narrowband (NB) AMR codec supports eight source rates or codec modes ranging from 4.75 kbit/s to 12.2 kbit/s when in GSM Enchanced Full Rate mode (GSM-EFR), and is capable of changing the bit rate every 20 ms if needed. The AMR codec features PLC to cope with transmission errors and lost packets, Voice Activity Detection (VAD) for detection silent periods and Comfort Noise (CN) insertion. The bit rate of the Silence Descriptor (SID) frame is 1.8 kbit/s. The sampling rate of the AMR codec is 8 kHz, and the frequency range of the codec is 300 - 3400 Hz [54].

Codec mode	**Bit rate**
AMR_12.20	12.20 kbit/s (GSM EFR)
AMR_10.20	10.20 kbit/s
AMR_7.95	7.95 kbit/s
AMR_7.40	7.40 kbit/s
AMR_6.70	6.70 kbit/s
AMR_5.90	5.90 kbit/s
AMR_5.15	5.15 kbit/s
AMR_4.75	4.75 kbit/s
AMR_SID	1.80 kbit/s

Table 2: Operational modes defined for the AMR-NB codec [54]

The wideband version of AMR, called AMR-WB was introduced in 2001 by the 3GPP to improve the speech clarity. It features nine source rates from 6.60 kbit/s to 23.85 kbit/s, and has a sampling rate of 16 kHz [55]. AMR-WB also features VAD, insertion of SID frames and PLC. The frequency range of AMR-WB is 50 to 7000 Hz. The frame size of both AMR and AMR-WB is 20 ms.

Codec mode	Bit rate
AMR-WB_23.85	23.85 kbit/s
AMR-WB_23.05	23.05 kbit/s
AMR-WB_19.85	19.85 kbit/s
AMR-WB_18.25	18.25 kbit/s
AMR-WB_15.85	15.85 kbit/s
AMR-WB_14.25	14.25 kbit/s
AMR-WB_12.65	12.65 kbit/s
AMR-WB_8.85	8.85 kbit/s
AMR-WB_6.60	6.60 kbit/s
AMR-WB_SID	1.75 kbit/s

Table 3: Operational modes defined for the AMR-WB codec [55]

Enhanced Voice Service (EVS)

Standardized in 2014, Enhanced Voice Service (EVS) is the most advanced audio codec so far specified by 3GPP. It is the successor for AMR-WB. The EVS encoder can switch it's bit rate every 20 ms according to needs. It is designed for both streaming voice and music, and features VAD, CN generation and PLC. The EVS codec is a fullband codec and supports sampling rates from 8 to 48 kHz.

The operational modes for the EVS codec are listed in Table 4.

Bit rate	Bandwidth
5.9 kbit/s	NB, WB
7.2 kbit/s	NB, WB
8.0 kbit/s	NB, WB
9.6 kbit/s	NB, WB, SWB
13.2 kbit/s	(NB), WB, SWB
16.4 kbit/s	NB, WB, SWB, FB
24.4 kbit/s	NB, WB, SWB, FB
32 kbit/s	WB, SWB, FB
48 kbit/s	WB, SWB, FB
64 kbit/s	WB, SWB, FB
96 kbit/s	WB, SWB, FB
128 kbit/s	WB, SWB, FB

Table 4: Operational modes defined for the EVS codec [56]

Opus

Standardized by the IETF in 2012, Opus is one of the two MTI codecs defined for WebRTC. It is a fullband, open source, royalty-free, speech and audio codec designed to handle a wide range of different audio applications. Opus supports several modes, ranging from narrowband speech at 6 kbit/s to bit rates up to 510 kbit/s. The algorithmic delay of the codec ranges from from 5 ms to 66,5 ms, depending on the operational mode [57].

Operational Mode	**Audio Bandwidth**	**Sample Rate (Effective)**
NB	4 kHz	8 kHz
MB	6 kHz	12 kHz
WB	8 kHz	16 kHz
SWB	12 kHz	24 kHz
FB	20 kHz	48 kHz

Table 5: Operational modes defined for the Opus codec

As can be seen from Table 5, the Opus codec does not reproduce audio frequencies over 20 kHz in FB mode although theoretically speaking frequencies up to 24 kHz could be supported due to the sampling rate in FB mode. The reason for this is that 20 kHz is the generally accepted upper audio frequency limit of human hearing. Opus supports both transmitting in mono and stereo, and supports PLC as well. Opus can operate with both Constant Bit Rate (CBR) and Variable Bit Rate (VBR), and can thus adjust its bit rate, bandwidth and frame size to varying network conditions. [45]

Operational Mode	**Bit Rate**	**Recommended application**
NB	8 - 12 kbit/s	Speech
WB	16 - 20 kbit/s	Speech
FB	28 - 40 kbit/s	Speech
FB	48 - 64 kbit/s	Mono music
FB	64 - 128 kbit/s	Stereo music

Table 6: Recommended bit rates by the IETF, referred to as the "sweet spots" [58]

2.3.3 Codec interoperability

The WebRTC endpoints must support both the VP8 and the H.264 video codecs. Since 3GPP has defined H.264 as one of the mandatory codecs supported in MTSI, video-call interoperatibility without transcoding is possible between WebRTC endpoints and non-WebRTC 3GPP-compliant devices, e.g. from browser to mobile phone.

The WebRTC endpoints typically support the audio codecs Opus, G.711 and G.722, of which Opus and G.711 are so called MTI codecs. When calling POTS phones in the Public Switched Telephony Network (PSTN), the G.711 codec can be used in both ends, thus avoiding transcoding. Since 3GPP compliant devices such as mobile phones only support the codecs AMR, AMR-WB or EVS, transcoding is currently inevitable when making voice calls from e.g. a browser to a mobile phone.

Since the G.711 and G.722 codecs do not support VBR, they do not adapt to varying network conditions in the way that the AMR, AMR-WB, EVS or Opus codecs do. Therefore, ensuring a high speech quality can be challenging when using G.711 or G.722 codecs since the Internet works on a best-effort model. Since Opus is the only mandatory WebRTC audio codec capable of adjusting to varying network conditions, transcoding between Opus and AMR, AMR-WB or EVS would ensure the most reliable calls, although transcoding can cause degradation in speech quality [59]. However, transcoding when calling from a web browser to a mobile phone is currently inevitable, since there are no commonly supported codecs. This can be seen in Figure 9, which explains the different codecs supported in various endpoints. In this picture we assume that WebRTC endpoints support G.722.

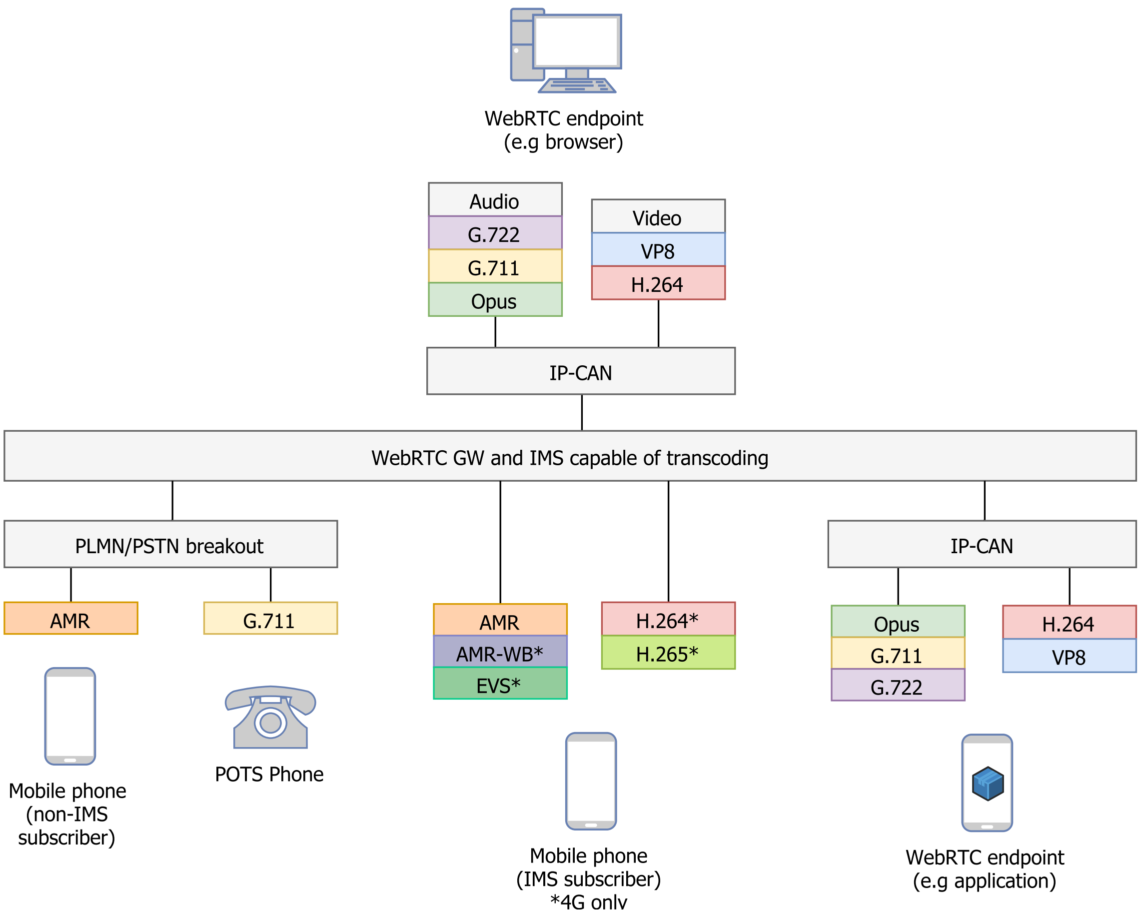

Figure 9: Supported codecs in different endpoints

2.3.4 Codec comparison

Table 7 describes the key features of the codecs presented. As can be seen from Table 7, G.711 and G.722 are not as well suited as the rest of the codecs for use over unstable connections, since they consume about 10 times more bandwidth and do not support VBR.

Codec	Sampling rate	Bandwidth	Bit rate	VBR
G.711	8 kHz	300 - 3400 Hz	64 kbit/s	No
G.722	16 kHz	50 - 7000 Hz	48 - 64 kbit/s	No
AMR	8 kHz	300 - 3400 Hz	4.75 - 12.20 kbit/s	Yes
AMR-WB	16 kHz	50 - 7000 Hz	6.6 - 23.85 kbit/s	Yes
EVS	8-48 kHz	20 - 20000 Hz	5.9 - 128 kbit/s	Yes
Opus	8-48 kHz	20 - 20000 Hz	6 - 510 kbit/s	Yes

Table 7: Comparison of codecs presented

2.3.5 Market overview

Many operators have already started using WebRTC to extend the capabilities of their IMS based networks, exposing their services to a large number of customers and devices. Some operators have created pure OTT services without any interworking with their IMS, while some allow WebRTC access to their IMS [60]. A few examples of different implementations are presented in this chapter.

AT&T Enhanced WebRTC SDK

The North-American operator AT&T has developed a WebRTC SDK that allows web developers to bring telephony functionality to their website with only a few lines of code. It is a JavaScript-based development framework, designed to make developing applications with WebRTC access to IMS easy. The developer does not have to learn about signaling or telecommunication protocols in order to develop a simple telephony application. In Figure 10, the architecture of an application built on top of the AT&T WebRTC SDK is illustrated.

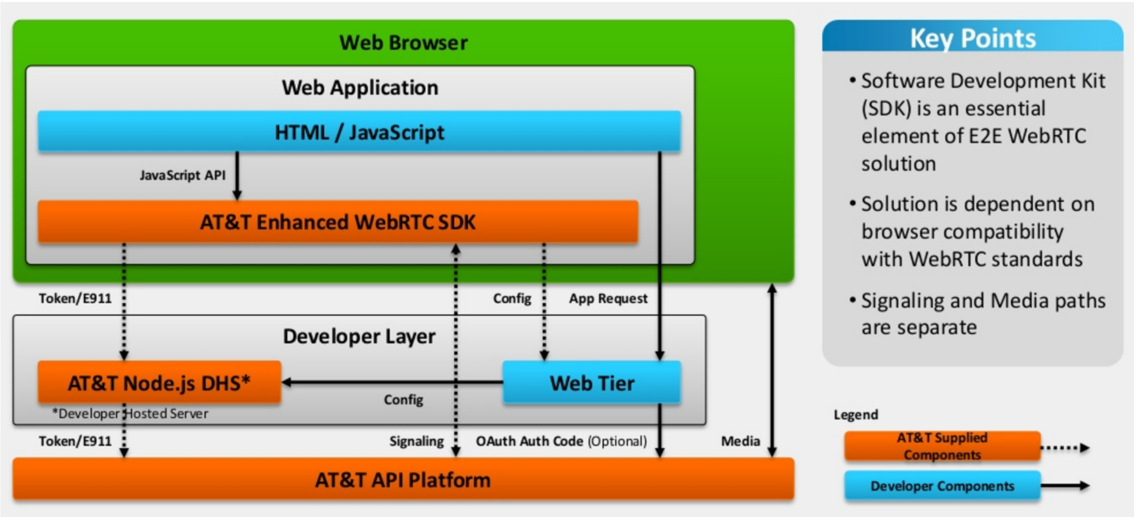

Figure 10: The architecture of an application made on top of the AT&T WebRTC SDK [61]

TU Go

TU Go is a service developed by Telefónica Digital, which is a part of the Spanish operator Telefónica. TU Go is sold under multiple Telefónica operator brands such as Movistar, O_2 and Vivo. The service is currently available in Peru, Argentina, Colombia, UK and Brazil. The service subscriber either downloads an application or uses the service directly through a WebRTC capable browser. The application allows the subscriber to use WiFi for making and receiving calls and text messages, on multiple devices. Since the call is routed over the Internet, the caller can avoid roaming costs in certain cases [62]. The TU Go application has been downloaded more than one million times from the Google Play store [63].

appear.in

A video conferencing service developed by an internal startup in the company Telenor Digital, which is a part of the Norwegian operator Telenor Group. appear.in is a pure OTT service designed for arranging quick conference calls. The service is WebRTC based and works in most browsers without installing plugins or other proprietary software. The user simply accesses the site https://appear.in and enters a preferred room name. After that the conference call starts and anyone aware of that particular room name can join the conversation if having a compatible browser [64].

2.4 Summary

WebRTC is a versatile technology, and combined with other technologies, such as IMS there is a great number of possible use cases. Still, VoIP and video communication has been around for a long time. The only difference is the ease of access and usage that WebRTC brings.

3 Implementation

The speech quality measurements were conducted in two different setups. A fully functional external IMS was used for the speech quality measurements and testing. The two setups were the following:

- A fully functional external IMS, including support for VoLTE. This variation is referred to as the Full IMS environment.
- A temporary setup, where the same external IMS was connected to the existing CS core network, using a SIP trunk. This allowed voice calls between a web browser and a mobile phone in the PLMN over 2G and 3G. This variation is referred to as the Hybrid environment.

An important objective of the measurements was to compare the speech quality of the two variations, in order to learn whether they could be used commercially.

3.1 Architecture

In the following section, the architecture and call flows in the two environments are described. High-level descriptions of the architectures are provided in figures 11-14.

3.1.1 Full IMS environment

In the so called Full IMS environment, an external IMS was used for test calls. High definition calls were possible to conduct to clients in an IP-CAN supporting MTSI. Figure 11 illustrates the call flow in the high-level architecture of the Full IMS environment. After fetching the WIC from a web server, the client connects to the WebRTC GW. The client is then able to make a call from the IMS to either a MTSI in an IP-CAN or an endpoint in the PSTN or PLMN.

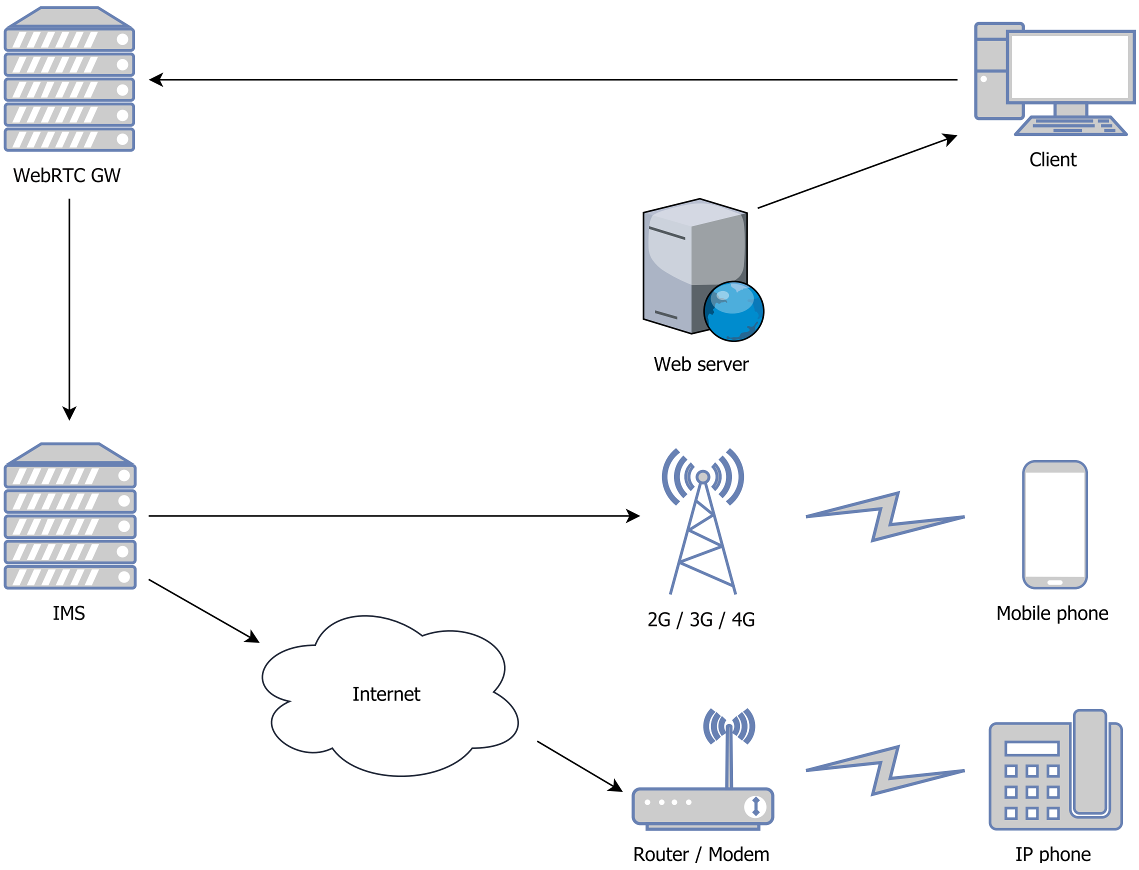

Figure 11: Call flow in the high-level architecture of the Full IMS environment

Figure 12 illustrates the signaling and media flow in the Full IMS environment. The WIC in the UE connects to the WebRTC GW. The signaling from the WIC to the WebRTC GW is handled using a HTTP Representational State Transfer (REST) API in the WebRTC GW, which is then translated by the WebRTC GW to SIP, which is then sent further to the eP-CSCF. The media stream is then set up between the WIC and eIMS-AGW using an SRTP connection, encrypted with the Datagram Transport Layer Security (DTLS) protocol. After the signaling has passed the I/S-CSCF, the AS and HSS are contacted with the Diameter (Dm) protocol. If the receiving party is a MTSI in an IP-CAN, the media stream continues from the eIMS-GW directly to the receiving party. In this case, the G.722 codec is used in the WIC while the eIMS-AGW transcodes that into AMR-WB before passing it further to the MTSI. The signaling passes through the P-CSCF before it reaches the MTSI. If the receiving party is a subscriber in the CS network, the WIC uses the codec G.711 which is transcoded into AMR in the eIMS-AGW. Both the media and signaling then pass the component called Session Border Control (SBC) before passing on to the MCGF and its corresponding Mobile Switching System (MSS) and IM-MGW. From there, both the media and the signaling are passed on to a Radio Network Controller (RNC) or a Base Station Controller (BSC) and from there further to the radio network.

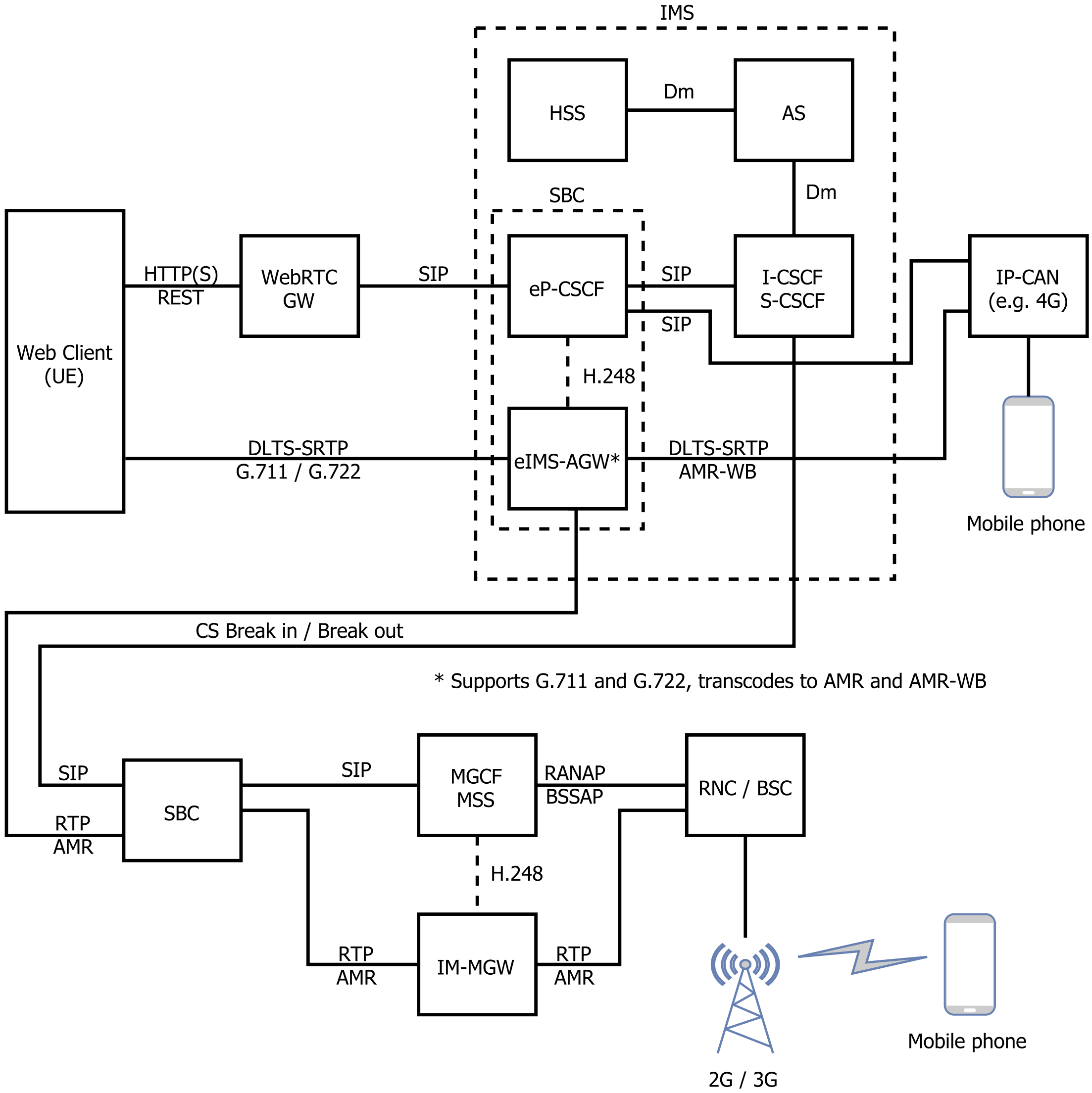

Figure 12: Call flow in the Full IMS environment

3.1.2 Hybrid environment

In the so called Hybrid environment, an external IMS was connected to the PSTN and PLMN using a SIP trunk. The SIP trunk only supported the codec G.711 A-law, with PLC. This limited the bandwidth of the calls to narrowband in this environment. Figure 13 illustrates the call flow in the high-level architecture of the so called Hybrid environment. After fetching the WIC from a web server, the client connects to the WebRTC GW. The client is then able to make a call from the IMS through the SIP trunk to either POTS in the PSTN or mobile phones in the PLMN.

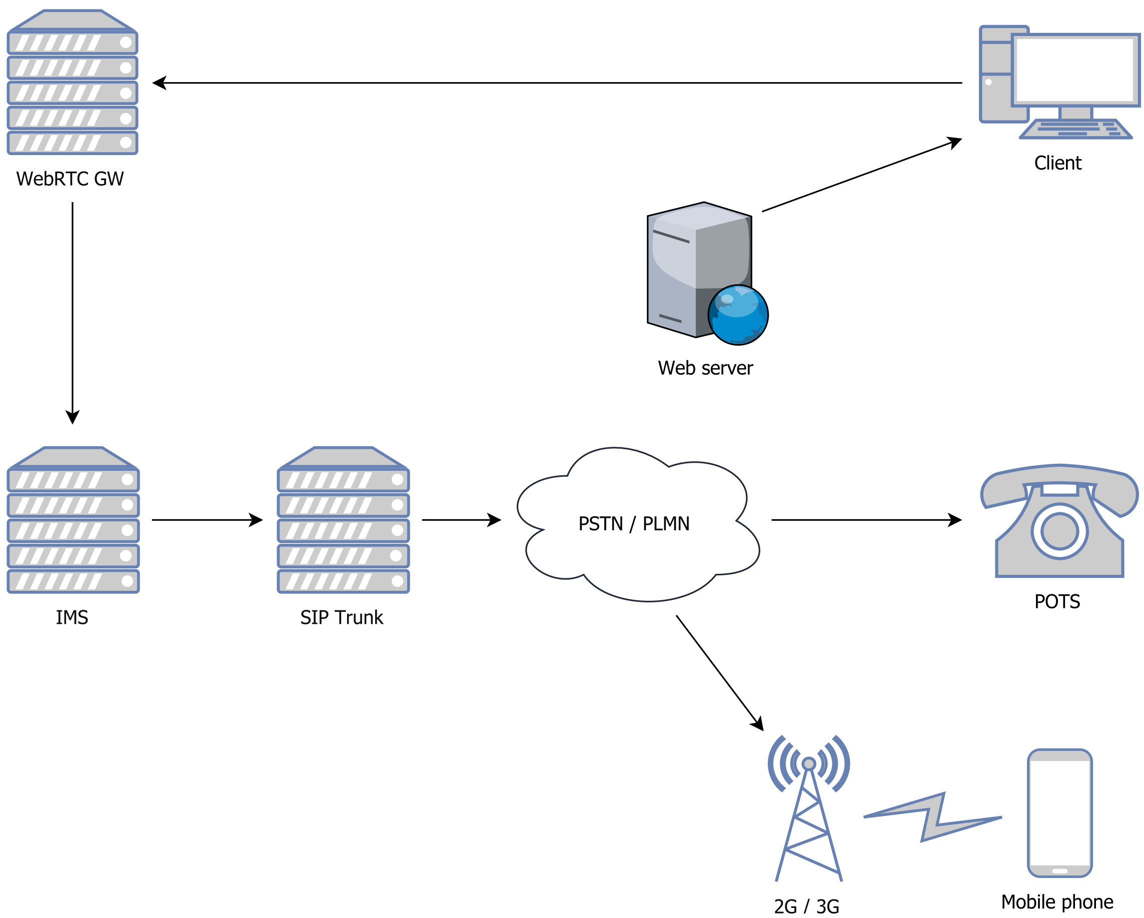

Figure 13: Call flow in the high-level architecture of the Hybrid environment

Figure 14 illustrates the signaling and media flow in the Hybrid environment. The WIC in the UE connects to the WebRTC GW. The signaling from the WIC to the WebRTC GW is handled using a HTTP REST API in the WebRTC GW, which is then translated by the WebRTC GW to SIP, which is then sent further to the eP-CSCF. The media stream is then set up between the WIC and eIMS-AGW using a DTLS-SRTP connection. After the signaling has passed the I/S-CSCF, the AS and HSS are contacted with the Diameter protocol. After this, both the media and signaling then pass the SBC before passing on to the SIP trunk connecting the external IMS to the CS local network. The SIP trunk consists of the logical network elements: A Media Gateway Controller Function (MCGF) and an IP Multimedia Media Gateway (IM-MGW). From MGCF onward, SIP-I is used as signaling when connecting to the MSS and its corresponding MGW. From there, both the media and the signaling are passed on to a RNC or a BSC and from there further to the radio network. Depending on the call case, the media session is either set up between the WIC and the IM-MGW in the SIP trunk or the MSS's corresponding MGW, of which both support G.711 A-law with PLC.

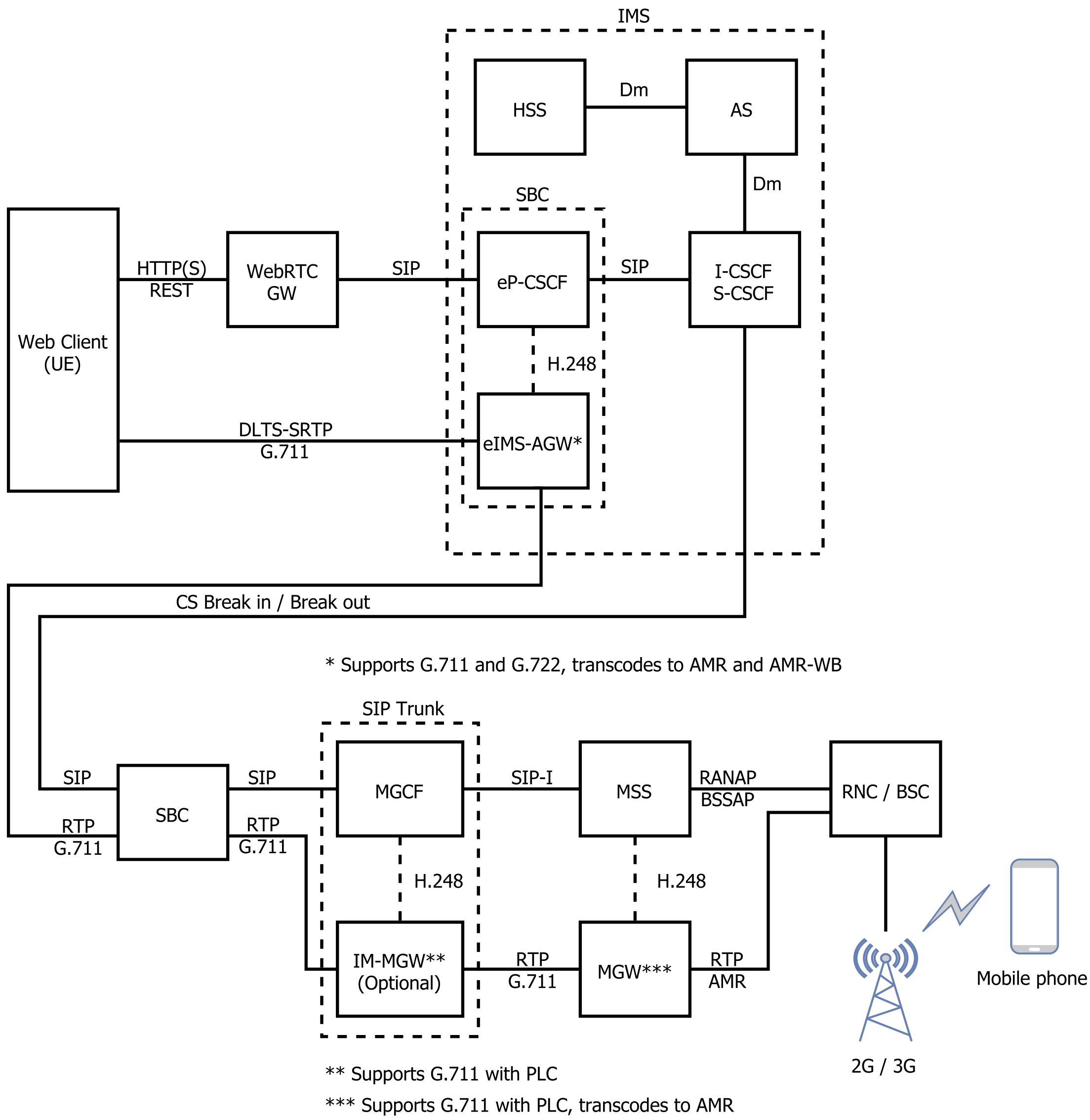

Figure 14: Call flow in the Hybrid environment

3.2 Supported codecs

As stated earlier, there are no commonly supported codecs for both WebRTC endpoints and MTSI or PLMN endpoints, e.g. mobile phones. In this particular implementation, the only codec that could be used in the browser when calling in the Hybrid environment was G.711 A-law, with PLC. The reason for this was that it was the only codec supported by the SIP trunk. The audio was then transcoded to AMR. In the Full IMS environment, the codec G.722 could be used when calling a MTSI. In the MTSI, AMR-WB was used. In case of CS breakout from the Full IMS environment, the codec used in the web browser was G.711 A-law and AMR in the mobile phone.

4 Measurements

In this chapter the measurement methods used in this thesis are presented in greater detail. Furthermore, the measurement results are presented and analyzed. The speech quality of the two test environments is evaluated and compared in different simulated environments, by adding packet-loss using a tool called Netem [65].

4.1 Measuring speech quality

There are generally speaking two approaches to measuring speech quality: Subjective and Objective. When performing subjective speech quality measurements, a person or a group evaluate the speech quality typically using a discrete Absolute Category Rating (ACR) scale:

Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

The test subjects are allowed to choose only one of the alternatives from the scale. From the results, a Mean Opinion Score (MOS) is then calculated to describe the speech quality. A score of 3 and above is considered acceptable speech quality. This method was standardized by the ITU-T as Recommendation P.800 in 1996 [66].

Although subjective measurements are very reliable, they are time-consuming. Therefore, objective methods such as the E-Model, PESQ and POLQA have been developed to produce similar results. The idea is that instead of having actual test subjects evaluating the speech quality, the objective tests are modeled to produce similar results as the subjective speech quality assessments.

Furthermore, measurement methods can be divided into intrusive and non-intrusive methods. Injecting traffic into the network and analyzing that is an intrusive method, while analyzing traffic already existing in the network is a non-intrusive method.

For example, the E-Model ITU-T G.107 predicts the speech quality by analyzing parameters that affect the speech quality, such as packet loss, jitter and latency. The E-Model is therefore suitable for making non-intrusive measurements in live networks [67]. The methods PESQ and POLQA compare the original transmitted signal to the received degraded signal, and then provide a MOS value. These methods are therefore suitable especially for intrusive measurements.

In this particular case, the speech quality is measured using the objective methods PESQ and POLQA, by injecting traffic into the network.

4.1.1 Perceptual Evaluation of Speech Quality (PESQ)

The PESQ method was originally specified by the ITU in 2001, as the ITU-T Recommendation P.862 and is still widely used for evaluating speech quality [68].

It is an objective method for end-to-end speech quality assessment of narrowband codecs with a sampling rate of 8 kHz, later expanded in ITU-T Recommendation P.862.2 to cover wideband codecs with a sampling rate of 16 kHz [69]. The reference signal is compared to the degraded signal that has passed through the network to produce a MOS score of -0.5 to 4.5.

The MOS values the PESQ method produces are called PESQ or RAW MOS and have to be converted to MOS-LQO which stands for MOS Listening Quality Objective in order to be comparable with MOS values obtained using other methods such as POLQA. This is done using the mapping function described below [70].

$$MOS_{LQO} = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.4945x+4.6607}} \tag{1}$$

In the equation, x is the RAW MOS value obtained from the PESQ analysis. The PESQ or RAW MOS to MOS-LQO mapping function is presented in Figure 15.

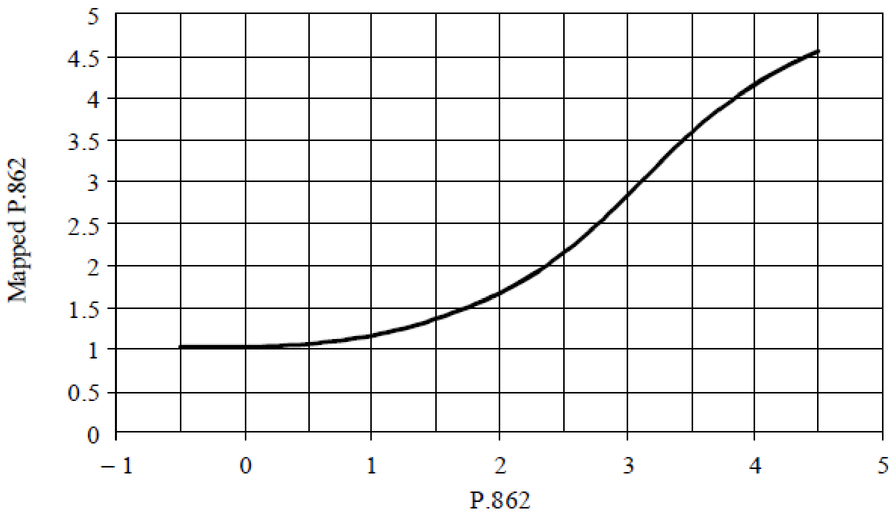

Figure 15: The PESQ or RAW MOS to MOS-LQO mapping function [70]

When analyzing with PESQ in wideband mode, the RAW MOS to MOS-LQO function is defined as follows [69]:

$$MOS_{LQO} = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.3669x+3.8224}} \tag{2}$$

4.1.2 Perceptual Objective Listening Quality Assessment (POLQA)

Even though wideband mode was introduced already in PESQ, a new method called POLQA has been developed for assessing modern super-wideband codecs. POLQA is a method first introduced by ITU in 2011, as Recommendation ITU-T P.863. It is an objective method for predicting speech quality ranging from narrowband to super-wideband.

POLQA is therefore suitable for assessing speech quality of wide and fullband codecs, such as Opus and EVS [71]. The idea is similar to PESQ: The original signal is compared to the degraded signal to produce a predicted MOS. There is no mapping function needed when comparing the results gained when assessing speech quality with the POLQA method, since the model produces MOS-LQO values as results. In Figure 16, the basic philosophy of PESQ and POLQA is presented. Both PESQ and POLQA compare the reference input to the degraded input. The models then predict what the average MOS of a subjective speech quality assessment would be in that case, using a cognitive model. This can be seen in the general block diagram in Figure 16.

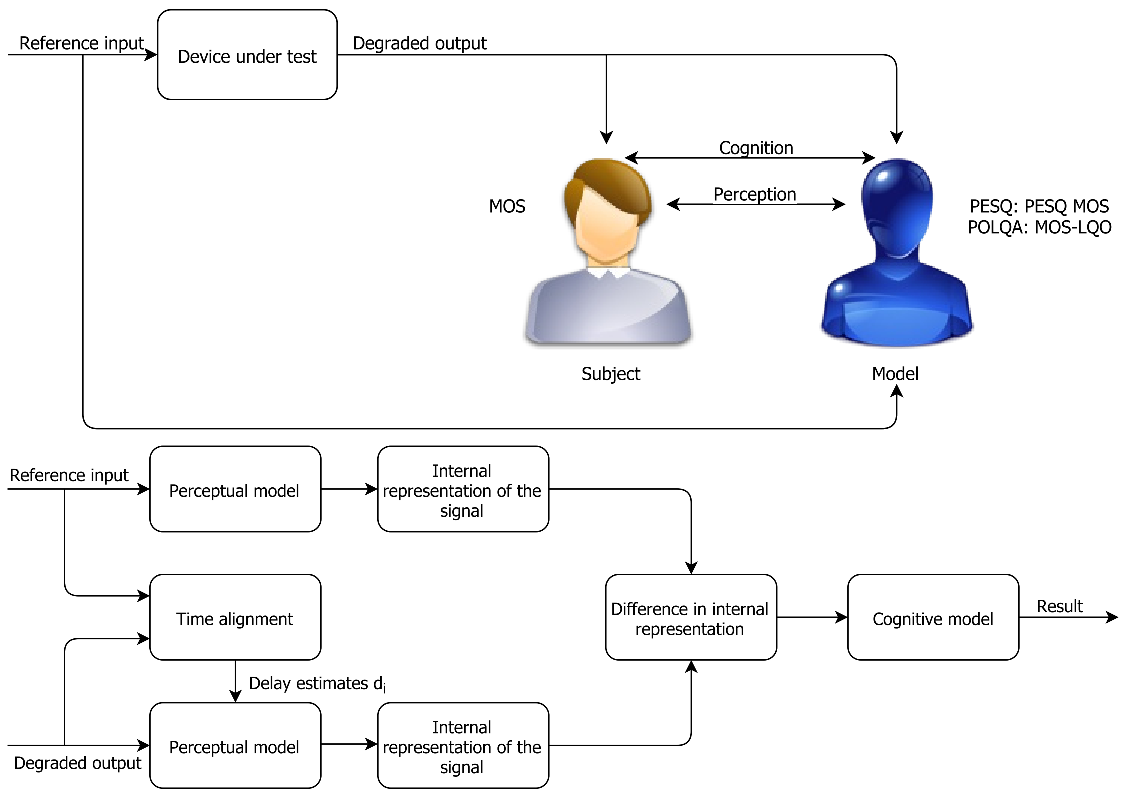

Figure 16: An overview of the basic philosophy used in both PESQ and POLQA [68][71]

Figure 17 describes the evolution of ITU-T recommendations for speech quality testing. The first recommendation by the ITU-T for objective speech quality assessment was published as ITU-T P.861 in 1996, called Perceptual Speech Quality Measure (PSQM). PSQM was then superseded by PESQ that was introduced in 2001, as ITU-T P.862. PESQ was extended to cover wideband codecs in 2005. Introduced in 2011, POLQA is now the most recent recommendation for objective speech quality assessment, and the only recommendation suitable for super-wideband codecs.

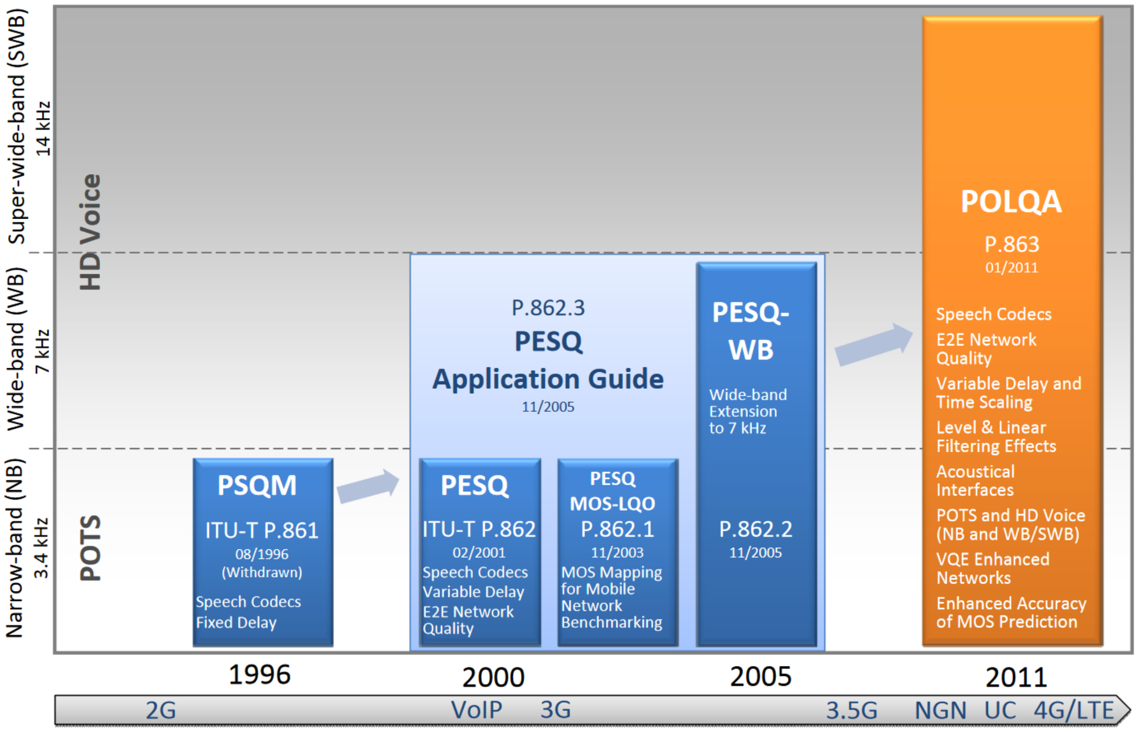

Figure 17: Evolution of ITU-T recommendations for speech quality testing [72]

4.2 Measurement setup

To evaluate and compare the performance of the Hybrid and Full IMS solutions, end-to-end, one-way speech quality measurements were carried out. The transmitted original signal was recorded simultaneously with the degraded received signal as a stereo track, at a sampling rate of 48 kHz. The speech quality was then evaluated using the methods POLQA and PESQ. To simulate various disturbances in the Internet connection, packet loss between 0 - 30% with 3% intervals was added to the transmitting end. For this a tool called Netem was used, which can be installed on Unix-based operating systems. The command used for adding e.g. 3% packet loss was:

```
sudo tc qdisc add dev enp0s25 root netem loss 3%
```

The reason behind the wide packet loss range was to identify trends or patterns not visible when using a more narrow packet loss range.

4.2.1 Test scenarios

The transmitted signal consisted of 32 concatenated speech samples from the ITU P.501 Amendment 2 archive, recorded at a sampling rate of 48 kHz [73]. The samples contained eight different languages, evenly distributed with both male and female voices. This was to ensure that the resulting MOS values would be as realistic and as fair as possible. Additionally, calls containing 96 identical concatenated samples with a Finnish male voice (sample filename: Amendment 2 finnish_m2.85dB.48k.wav) were recorded in each scenario to see whether the results would differ from the test calls with 32 concatenated speech samples.

The calling party and transmitting end was a laptop acting as a WebRTC endpoint running Ubuntu 16.04 LTS, which connected to the WebRTC GW over the Internet. The connection to the WebRTC GW was a stable connection with a latency around 2-5 ms. For the test Chromium 55 was used, which was running a WebRTC IMS client written in HTML and JavaScript. The client authenticated as an IMS subscriber, established a call to a predefined number, played the speech samples defined and then ended the call. The receiving end was a mobile phone connected to the radio network. The mobile phone was forced to connect to either the 2G, 3G or 4G radio network available, depending on the test case. All together 60 different scenarios were simulated, resulting in a total of 2240 samples analyzed. In addition to this, around 100 samples were analyzed for reference.

Architecture	**Network type**	**Packet loss (%)**										
Hybrid	2G 3G	0	3	6	9	12	15	18	21	24	27	30
Full IMS	2G 3G 4G	0	3	6	9	12	15	18	21	24	27	30

Table 8: The test matrix used when carrying out the tests

4.2.2 Recording setup

Since the speech samples provided by ITU are recorded in mono, the stereo signal from the laptop could be split into to two separate but identical analog signals with an 3.5 mm stereo plug to 2 x RCA adapter. One channel of the signal was then connected back to the microphone input of the laptop for transmission, while the other channel was connected to the left channel of a USB sound card for recording. The mobile phone headset jack was connected to the right channel of the USB sound card. In this way it was possible to both record and transmit the original signal as well as record the degraded signal, all simultaneously. The calls in the Hybrid environment were recorded using this setup illustrated in Figure 18.

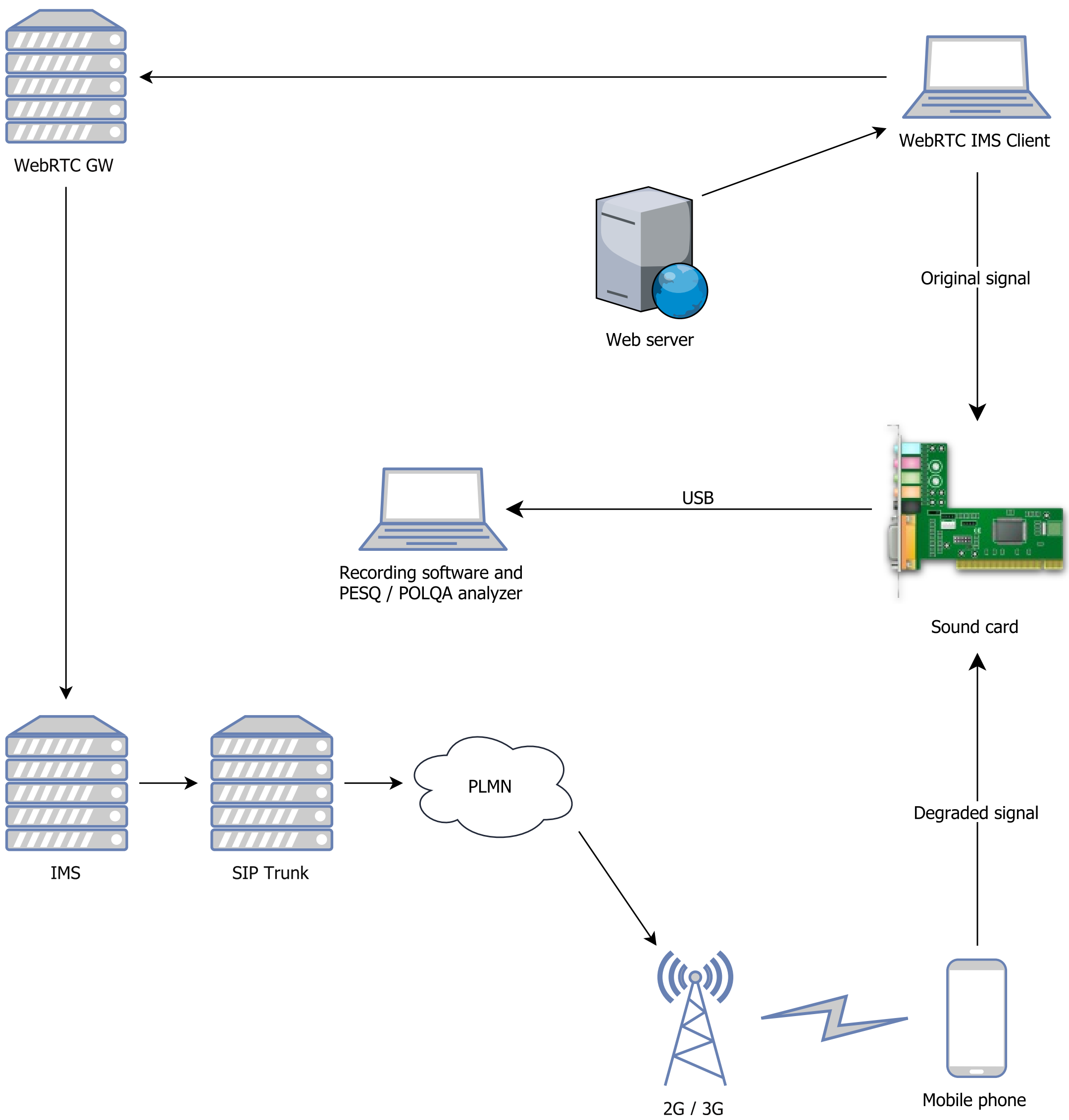

Figure 18: Hybrid recording setup

When measuring the Hybrid solution, speech quality was measured over both 2G and 3G. The reason for this was that there was no support for VoLTE available, and therefore measuring speech quality over 4G was not relevant. The codec used was G.711 A-law with PLC or PCMA8000 which has a sampling rate of 8 kHz, since that was the only codec the SIP trunk in place supported.

The same test calls were then repeated in the Full IMS environment, utilizing the same principle. When analyzing the Full IMS solution, speech quality was measured over both 2G, 3G and 4G (VoLTE). When calling over 2G and 3G, the codec was G.711 A-law with a sampling rate of 8000 Hz. When calling over 4G the codec used was G722 with a sampling rate of 16000 Hz. In the SDP ANSWER from the other party the codec is described as G722/8000 although the sampling rate is 16000 Hz. This is due to an error in the RFC1890 [74] specification and the erroneous sampling rate value has been kept for backward compatibility [75]. The recording setup for the Full IMS (FI) solution is illustrated in Figure 19. The calls were recorded in the same way as in the Hybrid solution.

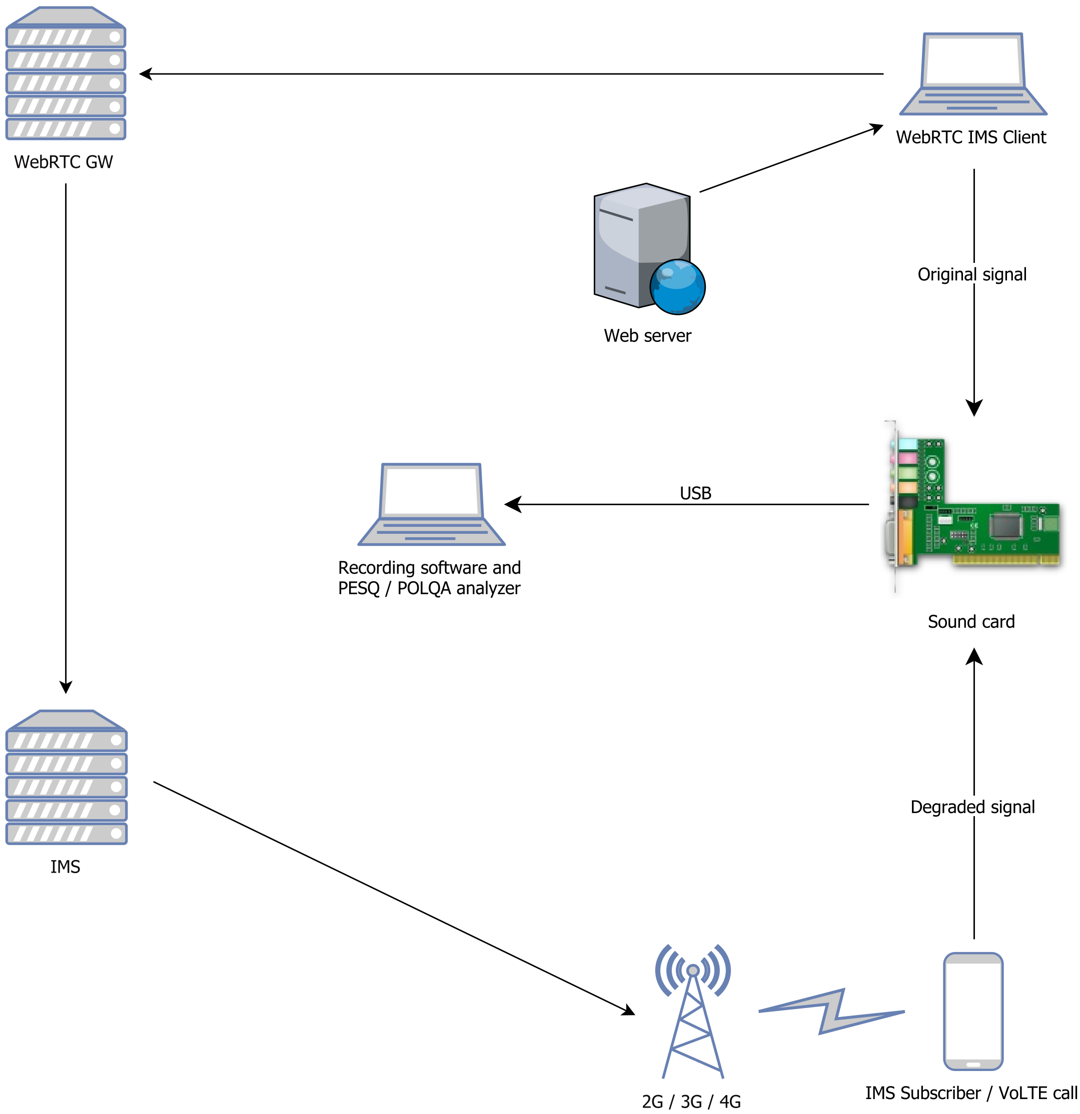

Figure 19: The Full IMS recording setup

4.2.3 Calibration

Analog-to-digital conversions and digital-to-analog conversion typically degrade a signal. To ensure minimum degradation, input and output levels were calibrated on the devices until there was no degradation observed when comparing the recorded original signal before transmission against the original speech samples. This was done using the POLQA method, until all the samples got the maximum POLQA MOS score, which is 4.75. Therefore we can assume that the measurement method in this particular case didn't considerably degrade the signal during transmission or during recording, making the results somewhat comparable with other speech quality measurements.

4.2.4 Probe measurements

In addition to the end-to-end measurements mentioned previously, the traffic from the IMS core to the SIP trunk was mirrored to a measurement device or probe to analyze the packet stream as well. The test setup is illustrated in Figure 20. This provided additional information about the overall packet stream quality and call setup time in the Hybrid solution.

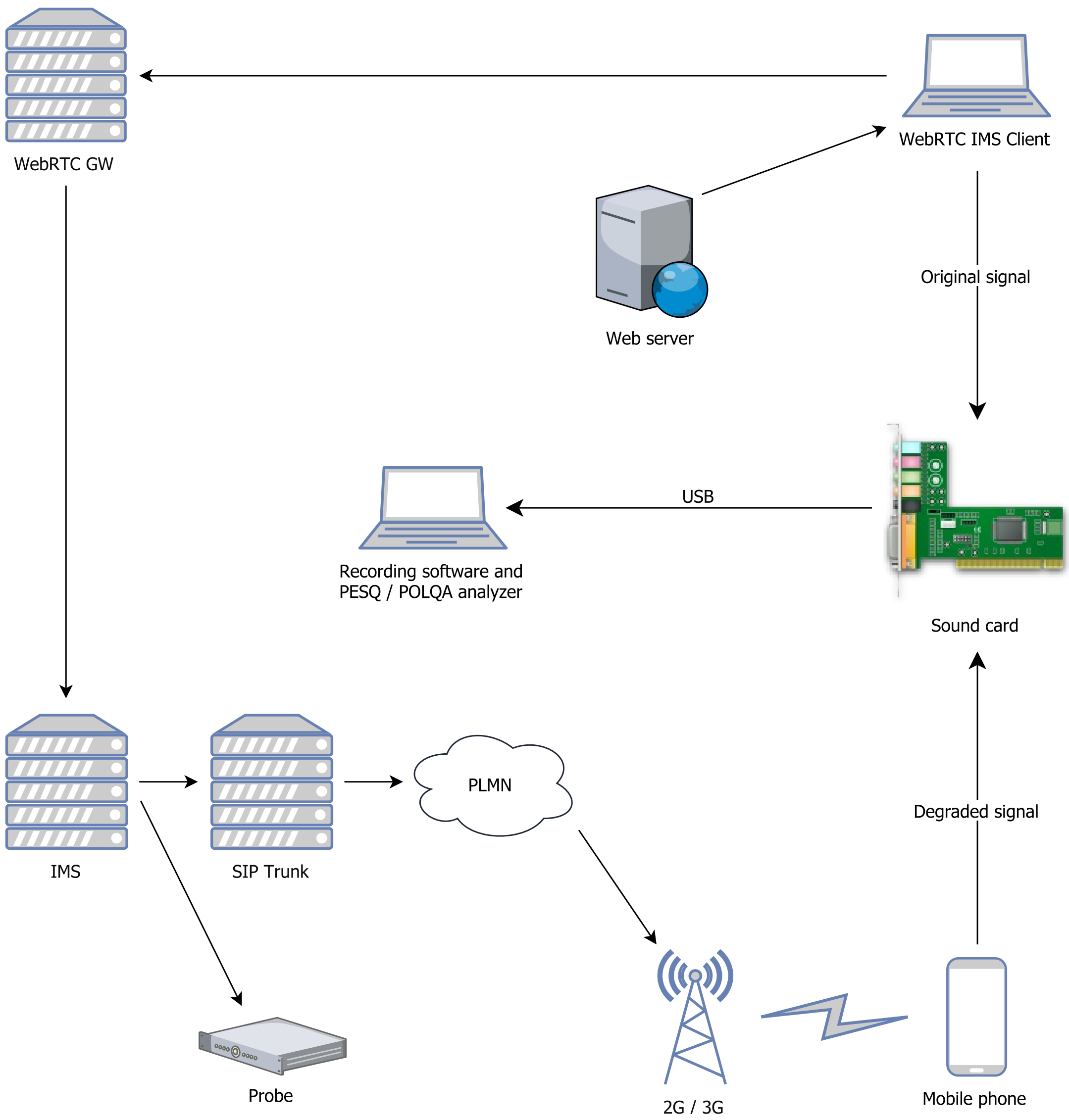

Figure 20: Hybrid recording setup with a measurement device or probe attached

4.3 Measurement results

The attached probe estimated speech quality and latency by analyzing the UDP packet stream, but those values could not be used because of protocol conversion on the way between the WebRTC IMS Client and the SIP trunk, leading to false latency values. In every analyzed call there was a fixed latency of 60 ms, and adding delay with Netem did not affect the value.

When no packet loss was present, the probe estimated that the MOS of the speech quality would be 4.1. This value could still not be used since it did not represent the end-to-end speech quality since the packet stream mirrored to the probe halfway, neither did it represent the end-to-end perceived speech quality as it only analyzed the packet stream. However, attaching the probe allowed us to confirm that the network was working correctly, since there was no packet loss observed from either side of the call, unless added.

Adding jitter or packet loss did not noticeably affect the call setup time. Since the probe was placed close to the radio network, there was never any measurable interference in the packet flow from the radio side. The average call setup time was 6770 milliseconds. In Figure 21, the call setup time in the Hybrid solution is presented in milliseconds. The green line indicates ITU-T recommended PSTN post dialing delay under normal load for local connections, the blue line indicates toll connections and the red line international connections [76].

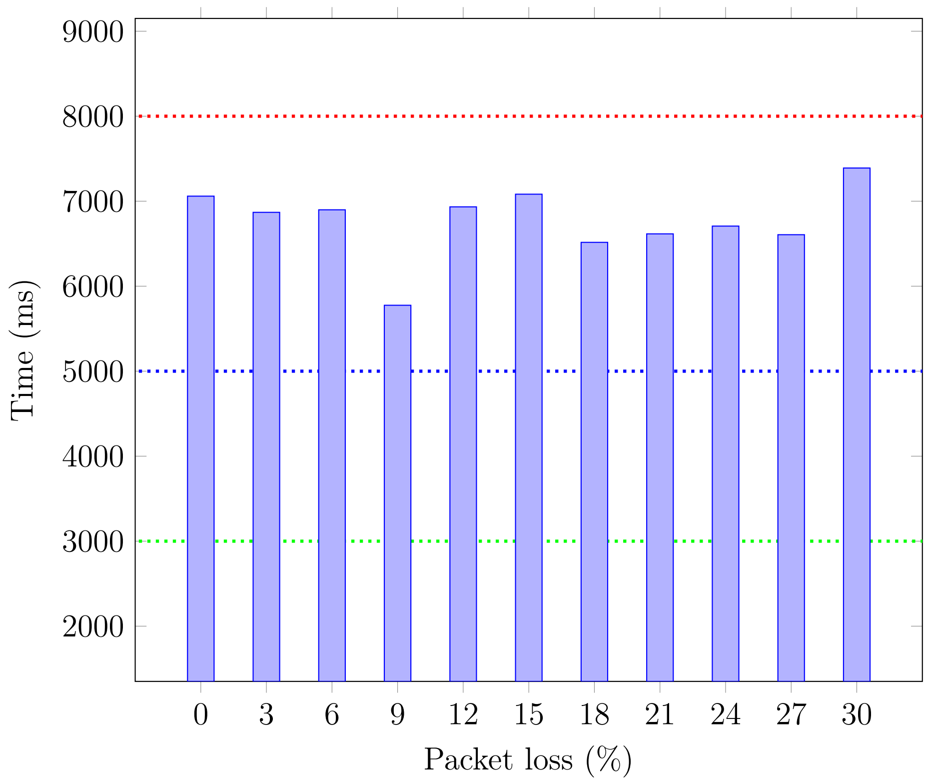

Figure 21: Call setup time (ms) in Hybrid solution with added packet loss

The measurement results when using the PESQ and POLQA methods are presented next. In Figure 22, the average MOS-LQO obtained with the PESQ method is presented. In Figure 23, the median MOS-LQO obtained with the same method is presented. The red dotted line presents the MOS of 3 which is still considered acceptable speech quality. The speech quality was acceptable in both environments when no added packet loss was added. When a 3% packet loss was added, only the speech quality in the Hybrid environment was acceptable. An explanation of the test cases is provided below.

Legend explanation

2G Hybrid
A call from a web browser to a mobile phone, using the SIP trunk connection between the external IMS and the local CS network. The mobile phone was forced to use the available 2G radio network. The browser was using the G.711 A-law codec with PLC (sampling rate 8 kHz), the other end AMR-NB.

3G Hybrid
A call from a web browser to a mobile phone, using the SIP trunk connection between the external IMS and the local CS network. The mobile phone was forced to use the available 3G radio network. The browser was using the G.711 A-law codec with PLC (sampling rate 8 kHz), the other end AMR-NB.

2G FI
A call from a web browser to a mobile phone, which was an IMS subscriber in the same network. The mobile phone was forced to use the available 2G radio network, leading to a CS breakout. The browser was using the G.711 A-law codec (sampling rate 8 kHz), the other end AMR-NB.

3G FI
A call from a web browser to a mobile phone, which was an IMS subscriber in the same network. The mobile phone was forced to use the available 3G radio network, leading to a CS breakout. The browser was using the G.711 A-law codec (sampling rate 8 kHz), the other end AMR-NB.

4G FI
A call from a web browser to a mobile phone. The mobile phone, which was an IMS subscriber was forced to use the available 4G radio network. The browser was using the G.722 codec (sampling rate 16 kHz), the other end AMR-WB.

4G VoLTE
An end-to-end VoLTE call using two mobile phones, which were IMS subscribers in the same network. The codec used was AMR-WB, no transcoding was required. Recorded for reference.

4G Opus
An end-to-end Opus browser call with a laptop running Chrome and a mobile phone running Chrome and using a 4G mobile data connection. The codec was Opus end-to-end without transcoding (sampling rate 48 kHz). Recorded for reference.

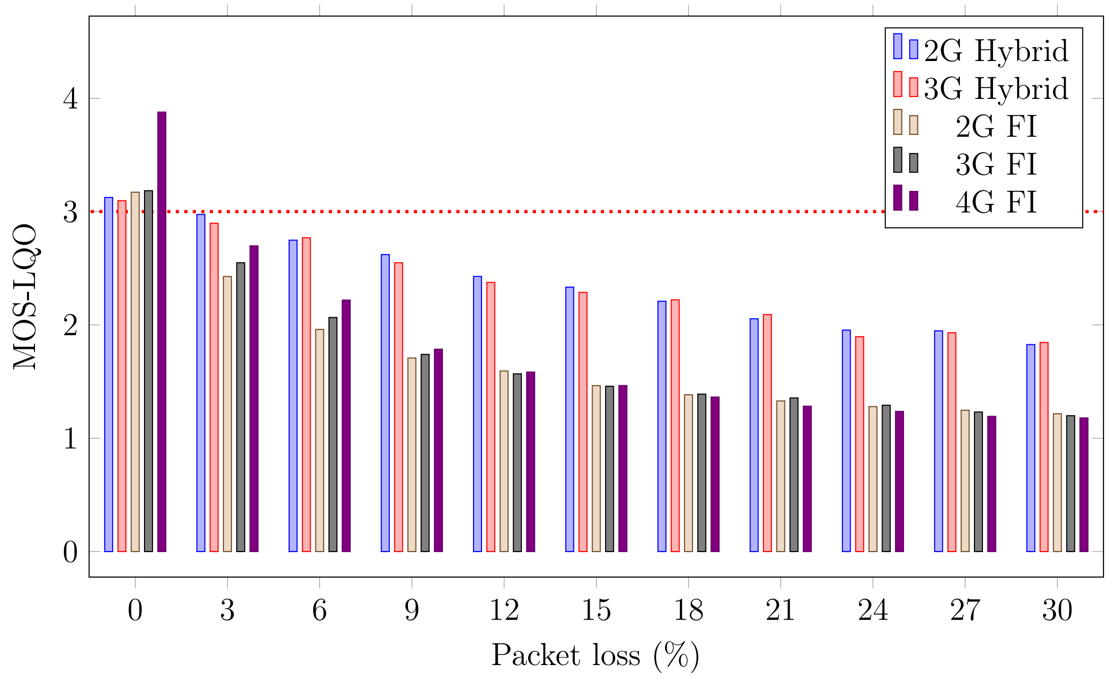

Figure 22: Average MOS-LQO obtained with the PESQ method

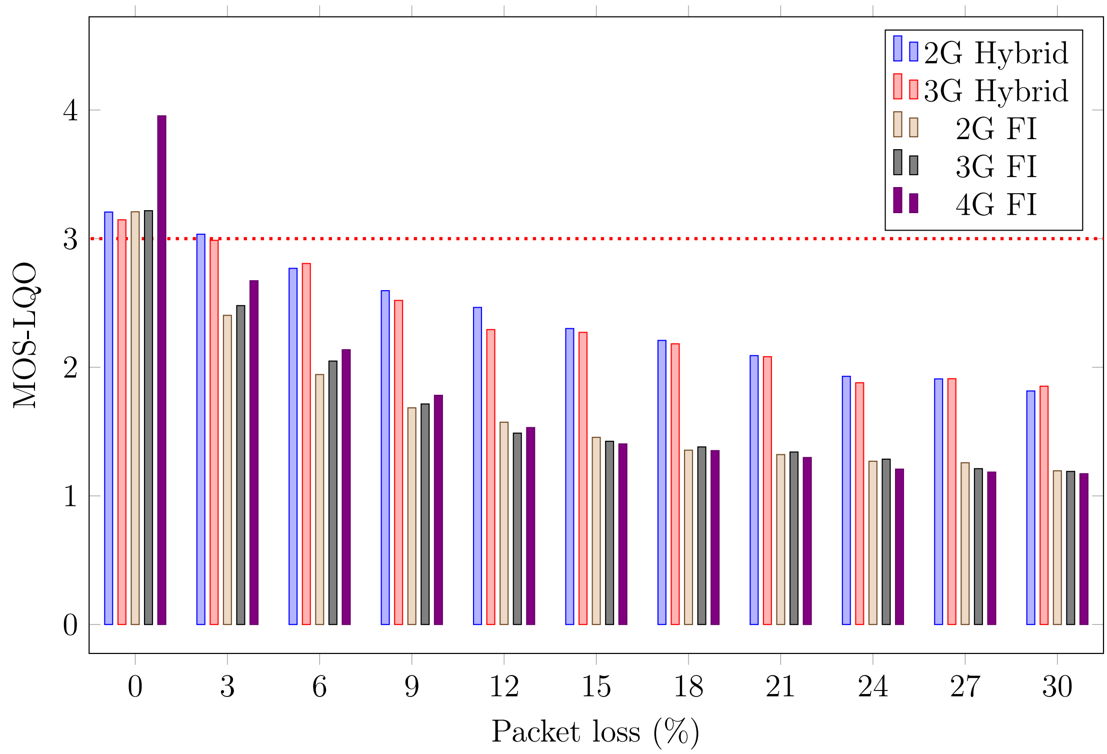

Figure 23: Median MOS-LQO obtained with the PESQ method

In Figure 24, the average MOS-LQO obtained with the POLQA method is presented. The red dotted line presents the MOS of 3 which is still considered acceptable speech quality. In Figure 25, the median MOS-LQO obtained with the same method is presented. The red dotted line presents the MOS of 3 which is still considered acceptable speech quality. Again, the speech quality is acceptable in both environments when no packet loss is added. When 3% packet loss is added, the speech quality is acceptable only in the Hybrid environment as well as in the Full IMS environment, when the end user has 4G connectivity.

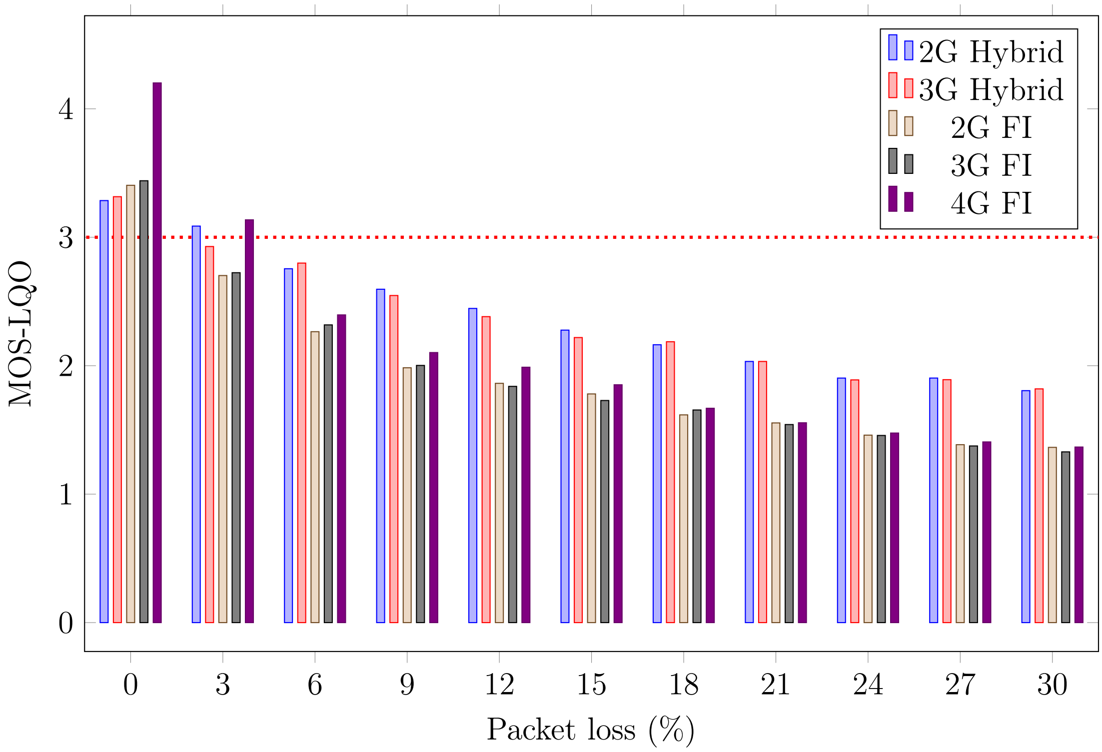

Figure 24: Average MOS-LQO obtained with the POLQA method

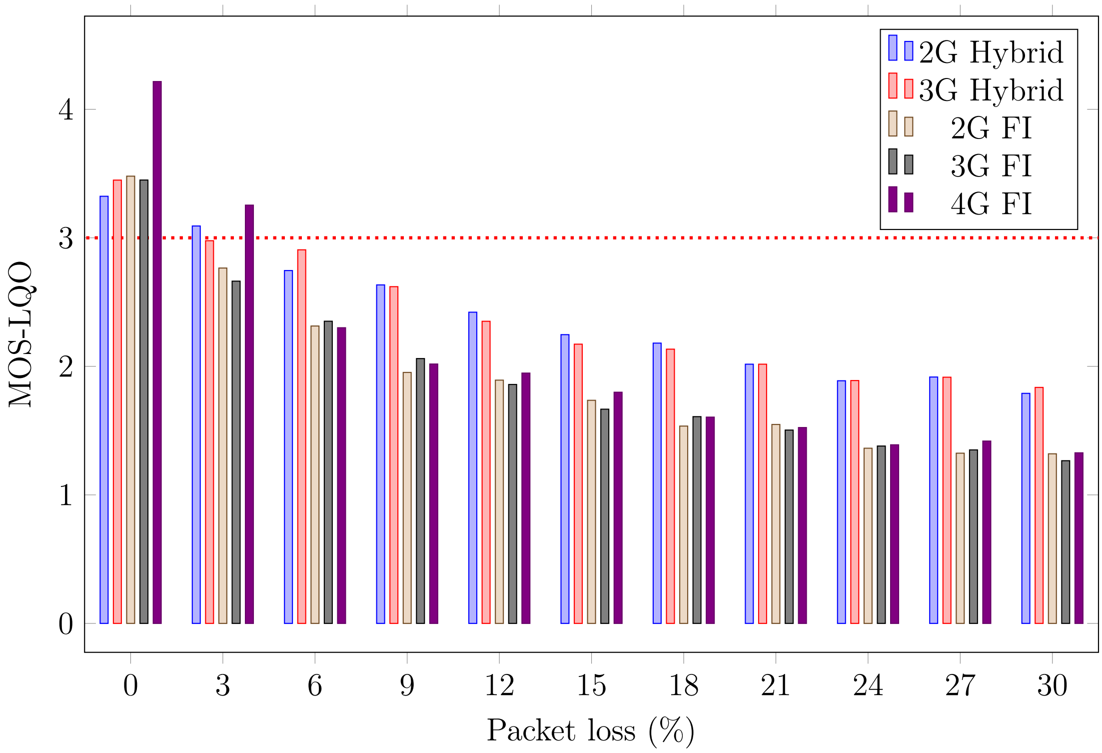

Figure 25: Median MOS-LQO obtained with the POLQA method

In Figure 26, the average mouth-to-ear latency obtained in POLQA analysis is presented. In Figure 27, the median mouth-to-ear latency obtained using the same method is presented. The green and red lines describe limits for latency. According to the ITU-T Recommendation G.114, latency should ideally be kept under 150 ms (green line), although latency up to 400 can be accepted depending on the context (red line) [76].

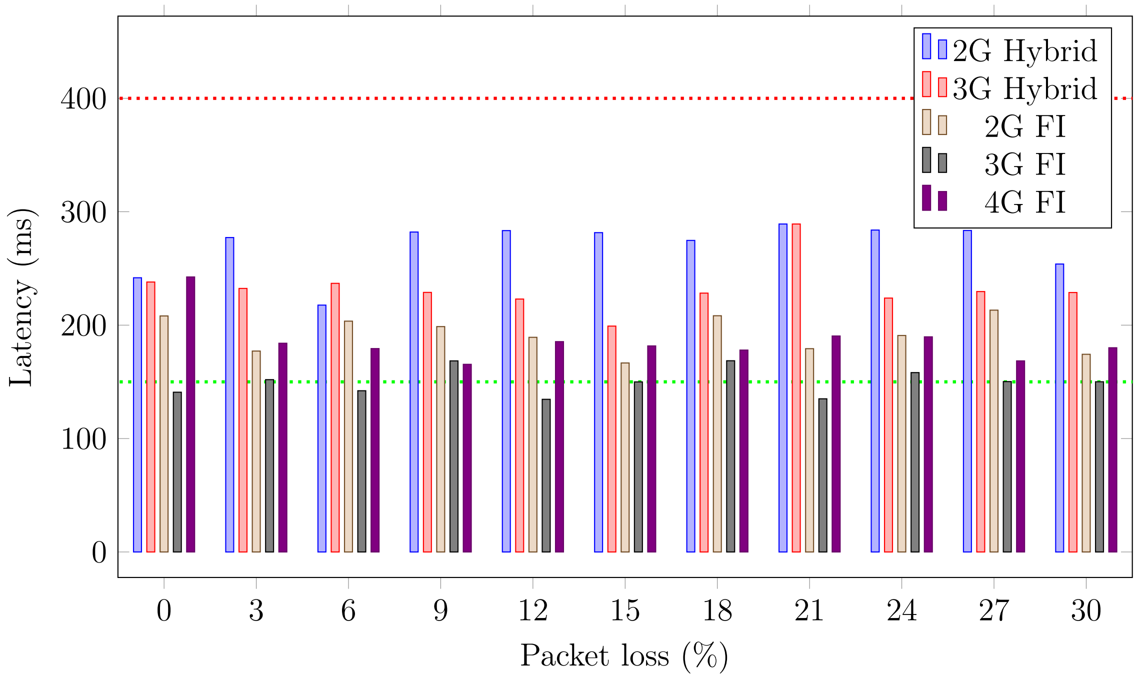

Figure 26: Average mouth-to-ear latency obtained with the POLQA method

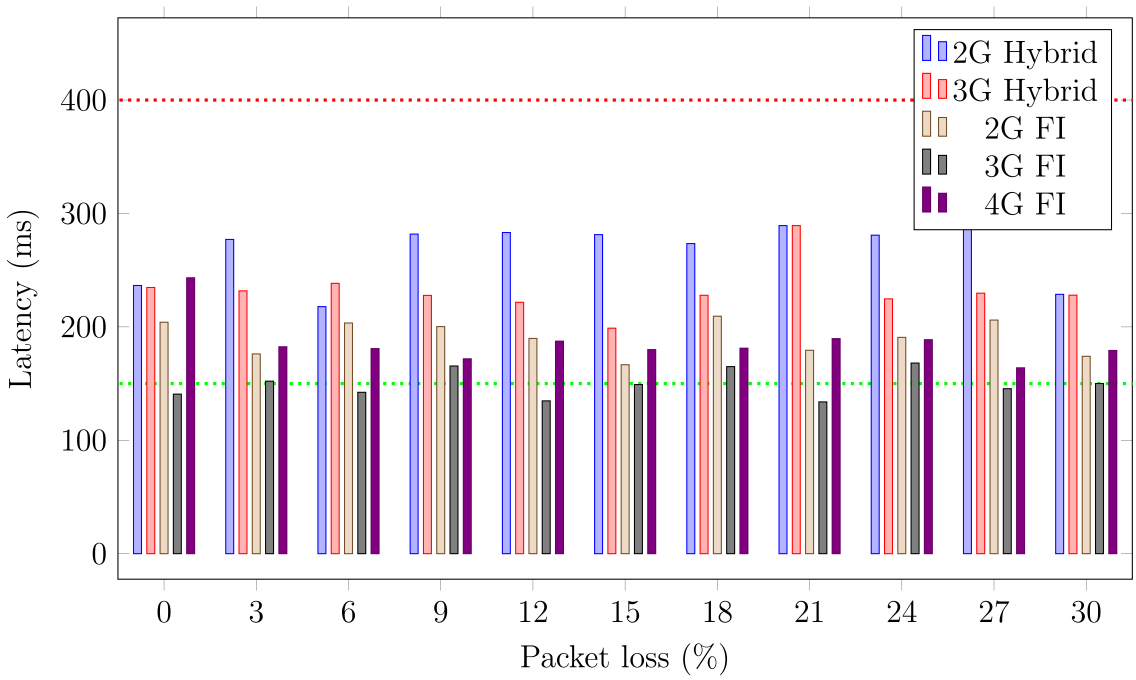

Figure 27: Median mouth-to-ear latency obtained with the POLQA method

In Figure 28, the average and median MOS results of the different environments are compared. The same file, called Amendment 2 finnish_m2.85dB.48k.wav was played back 96 times during the call, and no packet loss was added. Similar to the earlier results, speech quality is acceptable in all cases. The speech quality of the high definition calls is superior compared to the calls breaking out to the CS network. For reference, VoLTE calls and Opus browser to browser calls between two mobile phones were analyzed as well.

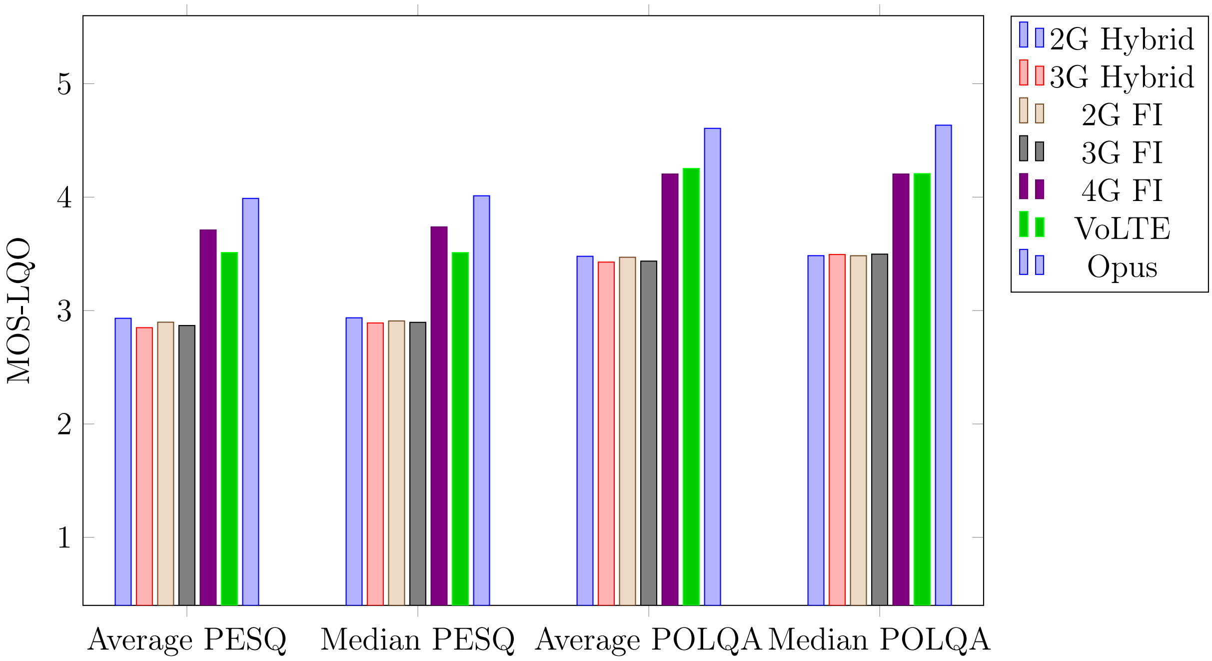

Figure 28: MOS-LQO values obtained with PESQ and POLQA analysis with 0% packet loss and using the same sample file Amendment 2 finnish_m2.85dB.48k.wav, 96 times

In Figure 29, the average and median mouth-to-ear latency of the different setups is compared. The same file, called Amendment 2 finnish_m2.85dB.48k.wav was played back 96 times during the call, and no packet loss was added. The green and red lines describe limits for latency. Latency should ideally be kept under 150 ms (green line), although latency up to 400 can be accepted depending on the context (red line). In this case, only the 3G FI and VoLTE call cases are close to the limit of 150 ms, while the rest exceed this limit.

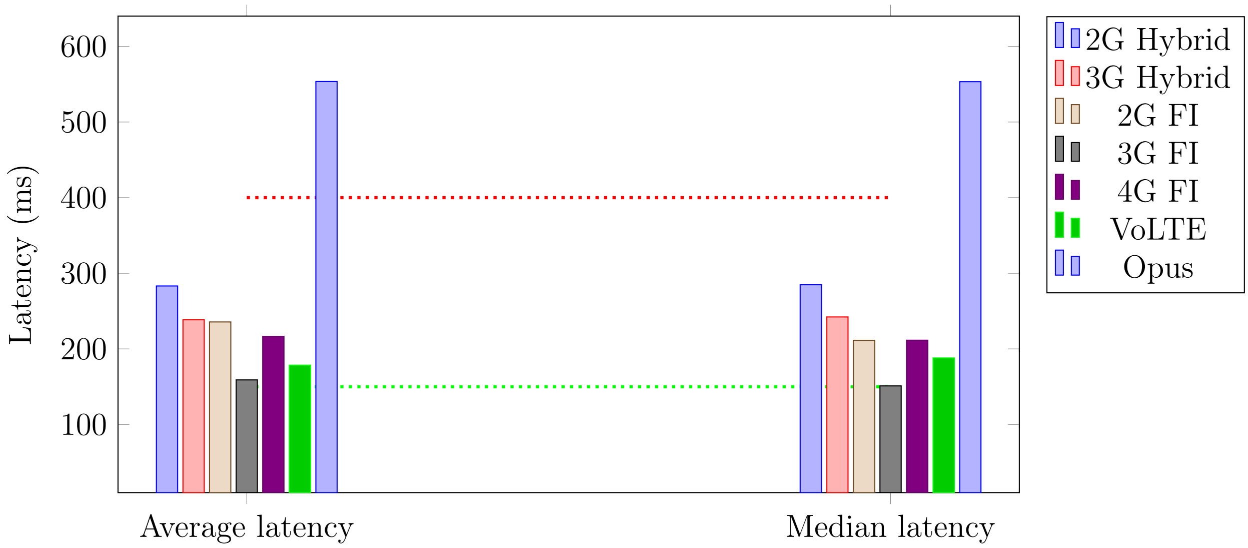

Figure 29: Mouth-to-ear latency obtained in POLQA analysis with 0% packet loss and using the same sample file Amendment 2 finnish_m2.85dB.48k.wav, 96 times

4.4 Measurement analysis

When comparing the speech quality of the calls made in the Hybrid solution to the Full IMS solution, packet loss degraded the speech quality more in the Full IMS solution, around 0,5 MOS with packet loss ranging from 6 to 30 percent. The reason for this is that the Hybrid solution had a built-in PLC (Packet Loss Concealment) mechanism, while the Full IMS solution lacked this functionality. The eIMS-AGW component which handled transcoding in this particular case did not support PLC with G.711 or G.722. The MGW in the Hybrid solution supported PLC with G.711.

The latency was higher in the Hybrid solution, around 60-80 ms. According to the ITU-T Recommendation G.114, latency should ideally be kept under 150 ms, although delays up to 400 can be accepted depending on the context. Figure 30 illustrates how latency affects the user satisfaction (R value) in the E-model [77].

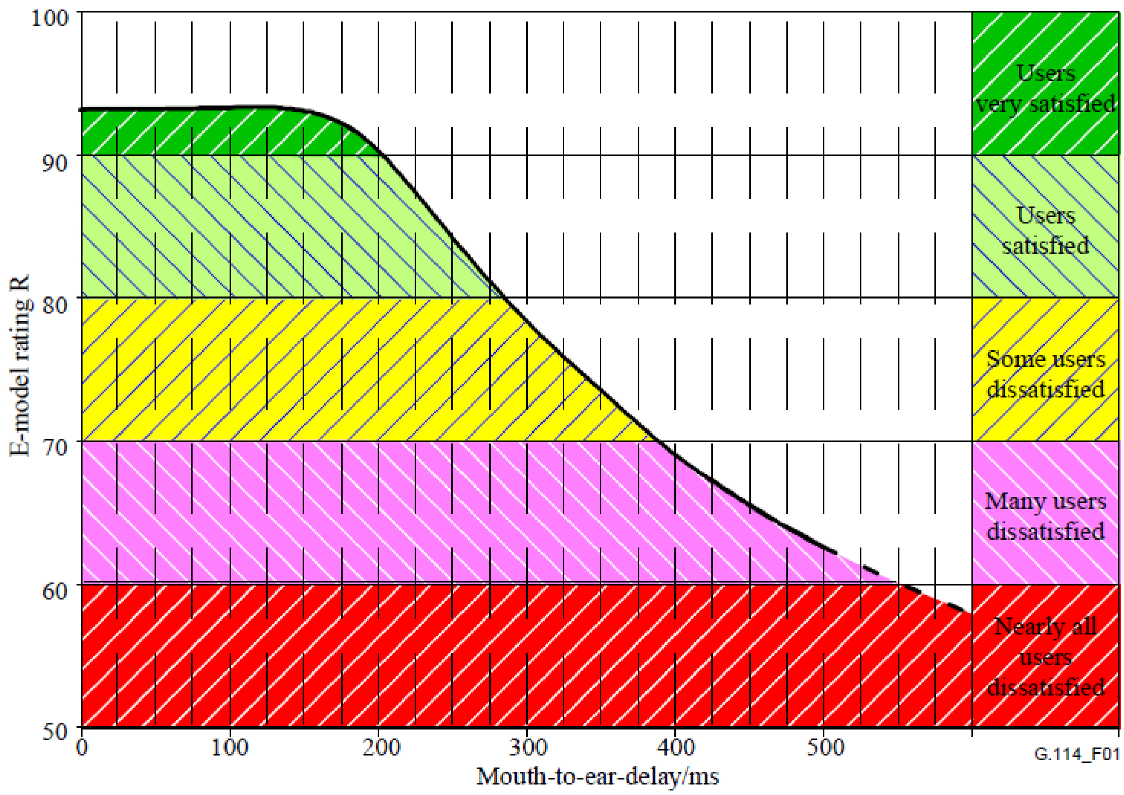

Figure 30: Figure of how the mouth-to-ear delay affects the user satisfaction rating (R value) in the E-model [77]

When making a browser-to-browser call using Opus without transcoding, the mouth-to-ear latency was around 550 ms. The reason for such high latency is partly the algorithmic delay of Opus, which is 66,6 ms in fullband mode. The algorithmic delay thus stood for 133 ms of the total delay in the call. Another reason might be that the browser Chrome used in the mobile phone simply did not process the audio fast enough. Since Opus is a complex codec, decoding the audio takes a lot of resources.

The speech quality was superior in VoLTE calls, which was expected. The latency when calling from the browser to a IMS subscriber was a bit higher than when calling to 2G or 3G. This is probably because the complexity of the codecs increase when using high definition codecs, which also makes the transcoding process slower.

The post dialing delay (PDD) was too long in the Hybrid solution, around 7000 ms. This should be investigated further.

4.5 Summary

The speech quality was acceptable in all call cases when no packet loss was added. The only case where speech quality wasn't acceptable was when calling from browser to browser, since the latency was over 500 ms using the Opus codec.

In the Hybrid solution, the speech quality of all calls over 2G and 3G were still acceptable with a 3 percent packet loss ratio. In the Full IMS solution, the speech quality for the high definition calls over 4G was still acceptable with a 3 percent packet loss ratio, while the speech quality was not acceptable for the narrowband calls over 2G and 3G.

5 Discussion

The results of the measurements show that this was only a scratching on the surface, and that there are a lot more interesting phenomena to investigate. In this thesis only voice calls were investigated, but obviously this could have been extended to video as well.

More extensive measurements could have been done to investigate the behavior of VoLTE calls and end-to-end Opus calls to see how they behave in different situations. Decreasing bandwidth and adding packet loss when measuring end-to-end Opus calls would have shown us how the Opus codec copes with interference.

Looking deeper into the reasons behind the unacceptably long mouth-to-ear delay in the browser-to-browser calls, when using the Opus audio codec would have been interesting. Could it be that the hardware in the devices used wasn't capable of processing the audio streams fast enough or could the latency have decreased if we would have used an application optimized for voice calls with Opus, such as WhatsApp? In the test, the Opus was used in fullband mode, which probably also affected the results due to the algorithmic delay of Opus. Using Opus in narrowband mode would probably have resulted in lower mouth-to-ear latency.

One might ask how it is possible that the Opus voice calls gained such high MOS values, considering the high end-to-end latency. The reason for this is that the methods POLQA and PESQ do not simply take static latency into account when assessing speech quality, since observing static latency in a listening test is not possible. Variations in latency are obviously taken into account.

Currently, there is no common audio codec that would be supported both in WebRTC and mobile phones. This could be solved by adding the Opus codec as one of the mandatory codecs in the MTSI. In this way, both parties could enjoy high quality voice calls without any transcoding. Another option would be implementing AMR, AMR-WB or EVS support in WebRTC endpoints, but as the codecs are not royalty-free, this will probably not happen in the near future unless the license of one of these codecs is changed in a way that browser makers could implement the support.

There is a commonly supported video codec, H.264 both in WebRTC endpoints and in mobile phones. Still, the next generation of video codecs, VP9 and H.265 are already available which makes it possible to achieve the same video quality with about 40 percent savings in bit rate [78]. Youtube is already using the royalty-free codec VP9 for streaming videos [79], and H.265 hardware support is already included in most mobile phones. Qualcomm also has mobile chipsets with VP9 hardware support [80]. Still, IETF has not decided upon the next MTI video codec for WebRTC, and at this point it is hard to say whether VP9 or H.265 will be one of the next MTI video codecs. Again, H.265 is not a royalty-free codec, which is an obstacle for the manufacturers of browsers to include codec support. Recently, a collaboration project called the Alliance of Open Media was founded. Among the members of this alliance are well known companies such as Amazon, ARM, Cisco, Google, Intel, Microsoft, Mozilla, Netflix and NVIDIA. The goal of the alliance is to develop open high-quality video codecs for streaming video over the web. Therefore it is possible that the next MTI codec for WebRTC is a codec by the Alliance of Open Media [81].

6 Conclusion

This study shows that acceptable speech quality can be achieved in voice calls between a web browser and a mobile phone utilizing a WebRTC GW in an IMS environment, also when using an external IMS connected to the legacy network using a SIP trunk. An added packet loss of 3% still resulted in acceptable speech quality when PLC was used, which means acceptable speech quality can be achieved even when the user is experiencing heavy packet loss. This study also shows that 4G connectivity and VoLTE support is required for superior quality in this particular solution.

The recommendation is not to use the temporary solution until the post dialing delay has been shortened. Also, the recommendation is to add PLC support to the full IMS solution before offering this as a commercial product. Since most browsers already support WebRTC, there is no reason for postponing the deployment of this service after these issues have been fixed. Adding support for transcoding Opus to AMR and AMR-WB and vice versa is recommended. Fine tuning the Opus codec for lower latency in the browser to browser calls is necessary for a satisfying user experience. This can probably be done forcing the client to use the Opus codec with a lower sample and bit rate.

In the current implementation, authentication is done by logging in using a SIP URI and a password. To prevent fraud, an enforced strong password policy, two-step verification or strong authentication is recommended.

However, treating WebRTC only as a gateway to an operator network is an underestimation of its potential. Limiting the WebRTC access to voice only is not preferable, since video functionality is natively supported as well. WebRTC technology combined with an operator network can provide many useful functions, but WebRTC does not need a telephony network, it works well without it. Still, adding hardware support for the Opus codec would be a interesting move by mobile and IMS vendors.

An IMS will partly offer services the Internet has offered for more than ten years. Hence, setting up an IMS with WebRTC access will not end the struggle between operators and OTT service providers.

References

[1] D. B. et al. (2015, Jun.) Akamai's state of the internet Q1 2015 report. akamai-state-of-the-internet-report-q1-2015.pdf. Akamai. 150 Broadway, Cambridge, MA 02142. [Online]. Available: https://www.akamai.com/es/es/multimedia/documents/state-of-the-internet/akamai-state-of-the-internet-report-q1-2015.pdf

[2] B. Tabakov, "Techno-Economic Feasibility of Web Real-Time Communications," Master's thesis, Aalto University, Jan. 2014. [Online]. Available: https://aaltodoc.aalto.fi/bitstream/handle/123456789/12710/master_Tabakov_Boyan_2014.pdf

[3] "WebKit Feature Status," WebKit, May 2017, official website. [Online]. Available: https://webkit.org/status/#specification-webrtc

[4] C. Wang, "Parametric Methods for Determining the Media Plane Quality of Service," Master's thesis, Aalto University, Jun. 2016.

[5] "DNA tuo ensimmäisenä operaattorina Suomessa matkaviestinverkkoonsa 4G-puhelut," Press release, DNA Oyj, Mar. 2016, (in Finnish). [Online]. Available: https://corporate.dna.fi/lehdistotiedotteet?type=stt2&id=42408150

[6] "Elisa ottaa käyttöön sisäkuuluvuutta merkittävästi parantavat wifi-puhelut," Press release, Elisa Oyj, Sep. 2016, (in Finnish). [Online]. Available: https://palsta.elisa.fi/elisan-tiedotteet-2/elisa-ottaa-kayttoon-sisakuuluvuutta-merkittavasti-parantavat-wifi-puhelut-502546

[7] "Facebook's Phone Company: WhatsApp Goes To The Next Level With Its Voice Calling Service," Forbes, Apr. 2015, news article. [Online]. Available: https://www.forbes.com/sites/parmyolson/2015/04/07/facebooks-whatsapp-voice-calling/#70e569d11388

[8] Miikka Poikselkä et al., *The IMS : IP multimedia concepts and services*, 2nd ed. John Wiley & Sons, Ltd, 2007, no. 9780470031834.

[9] "IP-Multimedia Subsystem," 3GPP, May 2017, official website. [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/109-ims

[10] "IP Multimedia Subsystem (IMS) Core products," Nokia Oyj, May 2017, official website. [Online]. Available: https://networks.nokia.com/products/ip-multimedia-subsystem-ims-core-products

[11] "Cloud Core Network Solutions," Huawei Technologies Co., Ltd., May 2017, official website. [Online]. Available: http://carrier.huawei.com/en/products/core-network/cs-ims/ims

[12] "Ericsson Telecom Core," Ericsson Ab, May 2017, official website. [Online]. Available: https://www.ericsson.com/ourportfolio/products/telecom-core

[13] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," RFC2543, Mar. 1999.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC3261, Jun. 2002.

[15] G. Camarillo, R. Kantola, and H. Schulzrinne, "Evaluation of Transport Protocols for the Session Initiation Protocol," *IEEE Network*, vol. 17, no. 5, pp. 40–46, Sep. 2003. [Online]. Available: https://aaltodoc.aalto.fi/bitstream/handle/123456789/5113/publication3.pdf

[16] M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC4566, Jul. 2006.

[17] J. Liu, S. Jiang, and H. Lin, "Introduction to Diameter," IBM Corporation, Jan. 2006. [Online]. Available: http://www.ibm.com/developerworks/library/wi-diameter

[18] J. Arkko, J. Loughney, G. Zorn, and V. Fajardo, "Diameter Base Protocol," RFC6733, Oct. 2012.

[19] "Gateway control protocol: Version 3," Recommendation H.248.1, ITU-T, Mar. 2013. [Online]. Available: https://www.itu.int/rec/T-REC-H.248.1-201303-I/en

[20] "IP Multimedia Subsystem (IMS); Stage 2," TS 23.228 V14.3.0, 3GPP, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=821

[21] "IMS Application Level Gateway (IMS-ALG) – IMS Access Gateway (IMS-AGW); Iq Interface; Stage 3," TS 29.334 V14.2.0, 3GPP, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1710

[22] H. Alvestrand, "Google release of WebRTC source code," Google, Jun. 2011, email archive. [Online]. Available: http://lists.w3.org/Archives/Public/public-webrtc/2011May/0022.html

[23] "Is WebRTC ready yet?" &yet, May 2017, open source website on GitHub. [Online]. Available: http://iswebrtcreadyyet.com

[24] H. Alvestrand, "WebRTC @5," Google, Jun. 2016, email archive. [Online]. Available: https://groups.google.com/forum/#!topic/discuss-webrtc/I0GqzwfKJfQ

[25] "WebRTC 1.0: Real-time Communication Between Browsers," Working Draft, W3C, May 2017. [Online]. Available: https://www.w3.org/TR/webrtc

[26] H. Alvestrand, "Overview: Real Time Protocols for Browser-based Applications," Internet-Draft, IETF, Mar. 2017. [Online]. Available: https://tools.ietf.org/html/draft-ietf-rtcweb-overview-18

[27] V. Andersson, "Network Address Translator Traversal for the Peer-to-Peer Session Initiation Protocol on Mobile Phones," Master's thesis, Aalto University, May 2010. [Online]. Available: https://aaltodoc.aalto.fi/bitstream/handle/123456789/3223/urn100205.pdf

[28] S. Dutton, "WebRTC in the real world: STUN, TURN and signaling," Google, Nov. 2013. [Online]. Available: http://www.html5rocks.com/en/tutorials/webrtc/infrastructure

[29] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," RFC5245, Apr. 2010.

[30] "WebRTC architecture," Google, May 2017, official website. [Online]. Available: https://webrtc.org/architecture

[31] R. Stidolph, "What's in a WebRTC JavaScript Library?" webrtcH4cKS, Apr. 2014. [Online]. Available: https://webrtchacks.com/whats-in-a-webrtc-javascript-library

[32] "Bowser - WebRTC on iOS," OpenWebRTC, May 2017, official website. [Online]. Available: https://www.openwebrtc.org/bowser/

[33] "Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); Stage 1," TS 22.228 V15.0.0, 3GPP, Sep. 2016. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=629

[34] I. B. Castillo, J. M. Villegas, and V. Pascual, "The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP)," RFC7118, Jan. 2014.

[35] V. Pascual, "The IMS approach to WebRTC," webrtcH4cKS, Mar. 2014. [Online]. Available: https://webrtchacks.com/ims-approach-webrtc

[36] "Media Gateway," Nokia Oyj, May 2017, official website. [Online]. Available: https://networks.nokia.com/products/media-gateway

[37] "Web Communication Gateway," Ericsson Ab, May 2017, official website. [Online]. Available: https://www.ericsson.com/ourportfolio/products/web-communication-gateway

[38] Li Tan, *Digital signal processing : fundamentals and applications.* Academic Press, Elsevier Inc., 2008, no. 9780123740908.

[39] "Media handling and interaction," TS 26.114 V14.3.0, 3GPP, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1404

[40] J. Valin and C. Bran, "WebRTC Audio Codec and Processing Requirements," RFC7874, May 2016.

[41] S. Proust, "Additional WebRTC Audio Codecs for Interoperability," RFC7875, May 2016.

[42] "Chrome M40 WebRTC Release Notes," Chrome release notes, Google, Dec. 2014. [Online]. Available: https://groups.google.com/forum/#!topic/discuss-webrtc/vGW4O3QOyLM/discussion

[43] "Media formats supported by the HTML audio and video elements," Official website, Mozilla, May 2017. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats

[44] "Microsoft Edge Codec Capabilities," Official website, Microsoft, May 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/mt599587(v=vs.85).aspx

[45] "Opus Codec," Xiph.org Foundation, May 2017, official website. [Online]. Available: https://opus-codec.org

[46] ITU-T Recommendation declared patent(s) - G.711, ITU-T Patent Information database, May 2017. [Online]. Available: http://www.itu.int/ITU-T/recommendations/related_ps.aspx?id_prod=911

[47] ITU-T Recommendation declared patent(s) - G.722, ITU-T Patent Information database, May 2017. [Online]. Available: http://www.itu.int/ITU-T/recommendations/related_ps.aspx?id_prod=919

[48] Official website, VoiceAge Corporation, May 2017. [Online]. Available: http://www.voiceage.com/Overview-lic.html

[49] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, no. 5, pp. 40–48, Oct. 1998. [Online]. Available: http://ieeexplore.ieee.org/document/730750/

[50] "Pulse Code Modulation (PCM) Of Voice Frequencies," Recommendation G.711, ITU-T, Nov. 1988. [Online]. Available: https://www.itu.int/rec/T-REC-G.711-198811-I/en

[51] "A high quality low-complexity algorithm for packet loss concealment with G.711," Recommendation G.711 - Appendix I, ITU-T, Sep. 1999. [Online]. Available: https://www.itu.int/rec/T-REC-G.711-199909-I!AppI/en

[52] "7 kHz audio-coding within 64 kbit/s," Recommendation G.722, ITU-T, Sep. 2012. [Online]. Available: https://www.itu.int/rec/T-REC-G.722-201209-I/en

[53] "A high-quality packet loss concealment algorithm for G.722," Recommendation G.722 - Appendix III, ITU-T, Nov. 2006. [Online]. Available: https://www.itu.int/rec/T-REC-G.722-200611-S!AppIII/en

[54] "Mandatory speech CODEC speech processing functions; AMR speech Codec; General description," TS 26.071 V14.0.0, 3GPP, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1386

[55] "Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description," TS 26.171 V14.0.0, 3GPP, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1420

[56] "Codec for Enhanced Voice Services (EVS); General overview," TS 26.441 V14.0.0, 3GPP, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1463

[57] "Definition of the Opus Audio Codec - RFC Errata," IETF, Nov. 2016. [Online]. Available: https://www.rfc-editor.org/errata_search.php?rfc=6716

[58] J. Valin, K. Vos, and T. Terriberry, "Definition of the Opus Audio Codec," RFC6716, Sep. 2012.

[59] M. Mikulec, M. Fajkus, M. Voznak, and O. Johansson, "Multiple Transcoding Impact on Speech Quality in Ideal Network Conditions," *AEEE Information and Communication Technologies and Services*, vol. 13, no. 5, pp. 552–557, Dec. 2015. [Online]. Available: http://advances.utc.sk/index.php/AEEE/article/viewFile/1519/1117

[60] I. Lajtos, D. O'Byrne, and E. Ersoz, "WebRTC to complement IP Communication Services," White Paper, GSM Association, Feb. 2016. [Online]. Available: http://www.gsma.com/network2020/wp-content/uploads/2016/02/WebRTC_to_complement_IP_Communication_Services_v1.0.pdf

[61] "AT&T Enhanced WebRTC API Overview," Presentation, AT&T, Feb. 2015. [Online]. Available: https://www.slideshare.net/Attdev/enhanced-web-rtcoverviewfinal

[62] "TU: Call and text over wifi with O2 & Movistar," Official website, Telefónica, May 2017. [Online]. Available: https://tu.com/en

[63] "TU (Movistar/O2/Vivo)," Play Store, Google, May 2017. [Online]. Available: https://play.google.com/store/apps/details?id=es.tid.gconnect

[64] "appear.in – one click video conversations," Official website, Telenor Digital, May 2017. [Online]. Available: https://appear.in/information/team

[65] "networking:netem [Linux Foundation Wiki]," Official website, Linux Foundation, May 2017. [Online]. Available: https://wiki.linuxfoundation.org/networking/netem

[66] "Methods for subjective determination of transmission quality," Recommendation P.800, ITU-T, Aug. 1996. [Online]. Available: https://www.itu.int/rec/T-REC-P.800/en

[67] "The E-model: a computational model for use in transmission planning," Recommendation G.107, ITU-T, Jun. 2015. [Online]. Available: https://www.itu.int/rec/T-REC-G.107

[68] "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Recommendation P.862, ITU-T, Feb. 2001. [Online]. Available: http://www.itu.int/rec/T-REC-P.862/en

[69] "Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs," Recommendation P.862.2, ITU-T, Nov. 2007. [Online]. Available: http://www.itu.int/rec/T-REC-P.862.2-200711-I/en

[70] "Mapping function for transforming p.862 raw result scores to mos-lqo," Recommendation P.862.1, ITU-T, Nov. 2003. [Online]. Available: https://www.itu.int/rec/T-REC-P.862.1-200311-I/en

[71] "Perceptual objective listening quality assessment," Recommendation P.863, ITU-T, Sep. 2014. [Online]. Available: https://www.itu.int/rec/T-REC-P.863-201409-I/en

[72] "POLQA - The Next-Generation Mobile Voice Quality Testing Standard," Official website, OPTICOM, SwissQual and TNO, May 2017. [Online]. Available: http://www.polqa.info

[73] "New Annex C - Speech files prepared for use with ITU-T P.800 conformant applications and perceptual-based objective speech quality prediction," Recommendation P.501 Amendment 2, ITU-T, Oct. 2014. [Online]. Available: https://www.itu.int/rec/T-REC-P.501-201410-S!Amd2/en

[74] H. Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC1890, p. 14, Jan. 1996.

[75] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC3551, p. 13, Jul. 2003.

[76] "Network grade of service parameters and target values for circuit-switched services in the evolving ISDN," Recommendation E.721, ITU-T, May 1999. [Online]. Available: https://www.itu.int/rec/T-REC-E.721-199905-I/en

[77] “One-way transmission time,” Recommendation G.114, ITU-T, May 2003. [Online]. Available: https://www.itu.int/rec/T-REC-G.114/en

[78] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, “Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders,” *Picture Coding Symposium (PCS), 2013*, no. 30, Dec. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6737766/

[79] “VP9: Faster, better, buffer-free YouTube videos,” YouTube Engineering and Developers Blog, Google Inc., Apr. 2015. [Online]. Available: https://youtube-eng.googleblog.com/2015/04/vp9-faster-better-buffer-free-youtube.html

[80] “Qualcomm Snapdragon 835 Mobile Platform,” Product description, Qualcomm, Mar. 2017. [Online]. Available: https://www.qualcomm.com/media/documents/files/snapdragon-835-mobile-platform-product-brief.pdf

[81] “Alliance for Open Media Established to Deliver Next-Generation Open Media Formats,” Press release, Alliance for Open Media, Sep. 2015. [Online]. Available: http://aomedia.org/press-releases/alliance-to-deliver-next-generation-open-media-formats

A PESQ analysis results

PESQ analysis results with average and median MOS and delay

samplefile	AvgMOS	MedMOS	AvgDelay	MedDelay
hybrid_2g_0loss.wav	3,124509375	3,2068	240,625	235
hybrid_2g_0loss_rep.wav	2,937371875	2,9339	320,25	332
hybrid_2g_0loss_rep_2.wav	2,9260625	2,93585	289,625	284
hybrid_2g_0loss_rep_3.wav	2,930896875	2,9412	237,25	232
hybrid_2g_3loss.wav	2,97480625	3,03445	276,6875	276
hybrid_2g_6loss.wav	2,74780625	2,76915	217,1875	218
hybrid_2g_9loss.wav	2,620853125	2,59555	281,5625	282
hybrid_2g_12loss.wav	2,4277375	2,4652	282,5625	282
hybrid_2g_15loss.wav	2,33331875	2,30105	281,0625	282
hybrid_2g_18loss.wav	2,20885	2,20885	272,875	272
hybrid_2g_21loss.wav	2,054075	2,0908	288	288
hybrid_2g_24loss.wav	1,954453125	1,92955	281,0625	280
hybrid_2g_27loss.wav	1,947571875	1,9094	281,3125	286
hybrid_2g_30loss.wav	1,8267	1,81575	259,9375	228
hybrid_3g_0loss.wav	3,096334375	3,14665	236,4375	234
hybrid_3g_0loss_rep.wav	2,844996875	2,86235	242,125	252
hybrid_3g_0loss_rep_2.wav	2,825475	2,89035	235,125	245
hybrid_3g_0loss_rep_3.wav	2,87605625	2,903	235,9375	232
hybrid_3g_3loss.wav	2,8971625	2,987	231	231
hybrid_3g_6loss.wav	2,7689375	2,80675	236	238
hybrid_3g_9loss.wav	2,54848125	2,5198	226,9375	227
hybrid_3g_12loss.wav	2,3748	2,29295	222,25	220
hybrid_3g_15loss.wav	2,287165625	2,2713	198,25	198
hybrid_3g_18loss.wav	2,221246875	2,1823	227,4375	228
hybrid_3g_21loss.wav	2,091578125	2,0821	230,75	230
hybrid_3g_24loss.wav	1,8958625	1,87945	222,3125	224
hybrid_3g_27loss.wav	1,930546875	1,91115	229,125	229
hybrid_3g_30loss.wav	1,84475	1,8528	227,375	228
ims_2g_0loss.wav	3,171059375	3,2089	207,8125	204
ims_2g_0loss_rep.wav	2,895103125	2,8883	210,75	210
ims_2g_0loss_rep_2.wav	2,892996875	2,9082	205,375	206
ims_2g_0loss_rep_3.wav	2,9032875	2,9275	288,8125	288
ims_2g_3loss.wav	2,427065625	2,4038	177,25	176
ims_2g_6loss.wav	1,960240625	1,9434	204,125	204
ims_2g_9loss.wav	1,70810625	1,6846	199,375	202
ims_2g_12loss.wav	1,593240625	1,57275	191,125	190
ims_2g_15loss.wav	1,464621875	1,45505	167,375	168
ims_2g_18loss.wav	1,383565625	1,35535	209,75	210
ims_2g_21loss.wav	1,329253125	1,3211	180,0625	180
ims_2g_24loss.wav	1,279375	1,26925	191,625	192
ims_2g_27loss.wav	1,247143333	1,25745	211,2	208

ims_2g_30loss.wav	1,215846875	1,19485	174,875	174
ims_3g_0loss.wav	3,184309375	3,21735	140,1875	140
ims_3g_0loss_rep.wav	2,837509375	2,88385	175,75	170
ims_3g_0loss_rep_2.wav	2,90338125	2,91085	150,8125	150
ims_3g_0loss_rep_3.wav	2,8625625	2,8957	148,5625	148
ims_3g_3loss.wav	2,5488875	2,47935	152,75	152
ims_3g_6loss.wav	2,065575	2,0482	143	144
ims_3g_9loss.wav	1,739596875	1,71455	169,0625	166
ims_3g_12loss.wav	1,568296875	1,4879	135,5	136
ims_3g_15loss.wav	1,458559375	1,42455	150,8125	148
ims_3g_18loss.wav	1,38918125	1,38065	170	164
ims_3g_21loss.wav	1,355421875	1,341	135,875	134
ims_3g_24loss.wav	1,291328125	1,2855	158,125	166
ims_3g_27loss.wav	1,2316	1,2116	150,5	146
ims_3g_30loss.wav	1,19888125	1,1901	152	150
ims_4g_0loss.wav	3,863653125	3,9334	250,5	248
ims_4g_0loss_2.wav	3,873940625	3,95505	229,6875	234
ims_4g_0loss_3.wav	3,896115625	3,97255	253,4375	256
ims_4g_0loss_rep.wav	3,735678125	3,77895	240,625	242
ims_4g_0loss_rep_2.wav	3,708615625	3,7196	198,875	206
ims_4g_0loss_rep_3.wav	3,6867	3,7373	210,5	217
ims_4g_3loss.wav	2,697475	2,6727	187,125	190
ims_4g_6loss.wav	2,21920625	2,1364	179,625	184
ims_4g_9loss.wav	1,78580625	1,7819	171,875	179
ims_4g_12loss.wav	1,584215625	1,5312	187,25	196
ims_4g_15loss.wav	1,46504375	1,4039	180,25	186
ims_4g_18loss.wav	1,363553125	1,35195	179,0625	184
ims_4g_21loss.wav	1,282903125	1,29825	196,8125	196
ims_4g_24loss.wav	1,23691875	1,2081	191,5	191
ims_4g_27loss.wav	1,193471875	1,185	165,9375	166
ims_4g_30loss.wav	1,17946875	1,1723	185,4375	188
opus_4g_0loss_rep.wav	3,996475	4,0117	551,6875	552
opus_4g_0loss_rep_2.wav	4,018915625	4,04605	551,8125	552
opus_4g_0loss_rep_3.wav	3,95034375	4,00065	554,125	554
ims_volte_trimmed.wav	3,51108	3,5105	191,6	198

B POLQA analysis results

POLQA analysis results with average and median MOS and delay

samplefile	AvgMOS	MedMOS	AvgDelay	MedDelay
hybrid_2g_0loss.wav	3,28568125	3,32325	241,778125	236,5
hybrid_2g_0loss_rep.wav	3,506896875	3,49845	321,5	332,9
hybrid_2g_0loss_rep_2.wav	3,444971875	3,47365	290,31875	284,8
hybrid_2g_0loss_rep_3.wav	3,48265	3,4839	237,8	233,5
hybrid_2g_3loss.wav	3,0868875	3,0921	277,25	277,1
hybrid_2g_6loss.wav	2,754959375	2,74545	217,665625	217,85
hybrid_2g_9loss.wav	2,595328125	2,63375	282,10625	281,8
hybrid_2g_12loss.wav	2,445946875	2,42135	283,38125	283,2
hybrid_2g_15loss.wav	2,277046875	2,2467	281,575	281,35
hybrid_2g_18loss.wav	2,163153125	2,18085	274,70625	273,45
hybrid_2g_21loss.wav	2,033084375	2,0168	289,178125	289,3
hybrid_2g_24loss.wav	1,9037	1,8879	283,846875	280,85
hybrid_2g_27loss.wav	1,90384375	1,9171	283,465625	286,7
hybrid_2g_30loss.wav	1,80581875	1,78985	253,8875	228,7
hybrid_3g_0loss.wav	3,315840625	3,44905	238,028125	234,75
hybrid_3g_0loss_rep.wav	3,462565625	3,49775	242,584375	251,85
hybrid_3g_0loss_rep_2.wav	3,369471875	3,4942	236,046875	242,25
hybrid_3g_0loss_rep_3.wav	3,448853125	3,49005	236,871875	233
hybrid_3g_3loss.wav	2,92795625	2,9775	232,365625	231,75
hybrid_3g_6loss.wav	2,7989375	2,90675	236,85	238,4
hybrid_3g_9loss.wav	2,546528125	2,62	228,86875	227,75
hybrid_3g_12loss.wav	2,382003125	2,35065	223,009375	221,7
hybrid_3g_15loss.wav	2,21926875	2,17265	199,1625	198,85
hybrid_3g_18loss.wav	2,18655625	2,13375	228,240625	227,85
hybrid_2g_21loss.wav	2,033084375	2,0168	289,178125	289,3
hybrid_3g_24loss.wav	1,88949375	1,88965	223,8625	224,7
hybrid_3g_27loss.wav	1,891228125	1,9155	229,61875	229,7
hybrid_3g_30loss.wav	1,81956875	1,83605	228,765625	227,95
ims_2g_0loss.wav	3,40446875	3,4797	208,11875	204,1
ims_2g_0loss_rep.wav	3,5138375	3,5215	211,371875	211,35
ims_2g_0loss_rep_2.wav	3,4251625	3,4314	206,096875	206
ims_2g_0loss_rep_3.wav	3,470240625	3,4833	289,496875	289,45
ims_2g_3loss.wav	2,702053125	2,765	177,153125	176,15
ims_2g_6loss.wav	2,264921875	2,31395	203,525	203,4
ims_2g_9loss.wav	1,983909375	1,9526	198,728125	200,25
ims_2g_12loss.wav	1,8629	1,89275	189,265625	189,8
ims_2g_15loss.wav	1,780284375	1,73575	166,653125	166,6
ims_2g_18loss.wav	1,61723125	1,53535	208,29375	209,4
ims_2g_21loss.wav	1,554584375	1,5478	179,253125	179,35
ims_2g_24loss.wav	1,459425	1,363	190,878125	190,75
ims_2g_27loss.wav	1,385127586	1,3244	213,2206897	206

ims_2g_30loss.wav	1,36435	1,3196	174,3	174,05
ims_3g_0loss.wav	3,439928125	3,44965	140,865625	140,7
ims_3g_0loss_rep.wav	3,40725	3,4877	176,5625	171,4
ims_3g_0loss_rep_2.wav	3,460709375	3,50095	151,40625	151,1
ims_3g_0loss_rep_3.wav	3,43925	3,4973	149,115625	149
ims_3g_3loss.wav	2,724028125	2,6629	151,940625	152,1
ims_3g_6loss.wav	2,317709375	2,35115	142,15625	142,3
ims_3g_9loss.wav	2,002190625	2,06055	168,4875	165,5
ims_3g_12loss.wav	1,83918125	1,85905	134,6	134,75
ims_3g_15loss.wav	1,729196875	1,6668	149,946875	149,15
ims_3g_18loss.wav	1,65526875	1,60875	168,603125	164,95
ims_3g_21loss.wav	1,54193125	1,5043	135,05	133,85
ims_3g_24loss.wav	1,456978125	1,38015	158,159375	168,1
ims_3g_27loss.wav	1,37523125	1,34995	150,225	145,5
ims_3g_30loss.wav	1,329190625	1,2659	150,065625	150,05
ims_4g_0loss.wav	4,20043125	4,18185	250,13125	243,3
ims_4g_0loss_2.wav	4,219459375	4,22425	228,609375	232,3
ims_4g_0loss_3.wav	4,18509375	4,21585	248,775	252,65
ims_4g_0loss_rep.wav	4,209421875	4,20345	237,0625	238,55
ims_4g_0loss_rep_2.wav	4,215415625	4,21335	203,215625	203,3
ims_4g_0loss_rep_3.wav	4,18565625	4,2023	209,340625	211,5
ims_4g_3loss.wav	3,13564375	3,2544	184,046875	182,45
ims_4g_6loss.wav	2,3955125	2,3005	179,3375	180,85
ims_4g_9loss.wav	2,102371875	2,01825	165,453125	171,85
ims_4g_12loss.wav	1,988234375	1,94775	185,465625	187,4
ims_4g_15loss.wav	1,851496875	1,79875	181,684375	179,95
ims_4g_18loss.wav	1,668728125	1,6051	178,025	181,2
ims_4g_21loss.wav	1,55555625	1,5239	190,44375	189,65
ims_4g_24loss.wav	1,4758125	1,38965	189,7	188,7
ims_4g_27loss.wav	1,40666875	1,41885	168,490625	163,9
ims_4g_30loss.wav	1,366890625	1,32735	180,065625	179,2
opus_4g_0loss_rep.wav	4,63010625	4,6343	552,596875	552,95
opus_4g_0loss_rep_2.wav	4,6364875	4,6409	552,9625	553,35
opus_4g_0loss_rep_3.wav	4,552525	4,6046	555,09375	555,2
ims_volte.wav	4,25266	4,2068	178,44	188