On-line Monitoring of VoIP Quality Using IPFIX

Petr MATOUSEK , Martin KMET , Martin BASEL

Department of Information Systems, Faculty of Information Technology, Brno University of Technology,
Bozetechova 2, Brno, Czech Republic

matousp@fit.vutbr.cz, ikmet@fit.vutbr.cz, xbasel02@stud.fit.vutbr.cz

Abstract. *The main goal of VoIP services is to provide a reliable and high-quality voice transmission over packet networks. In order to prove the quality of VoIP transmission, several approaches were designed. In our approach, we are concerned about on-line monitoring of RTP and RTCP traffic. Based on these data, we are able to compute main VoIP quality metrics including jitter, delay, packet loss, and finally R-factor and MOS values. This technique of VoIP quality measuring can be directly incorporated into IPFIX monitoring framework where an IPFIX probe analyses RTP/RTCP packets, computes VoIP quality metrics, and adds these metrics into extended IPFIX flow records. Then, these extended data are stored in a central IPFIX monitoring system called collector where can be used for monitoring purposes. This paper presents a functional implementation of IPFIX plugin for VoIP quality measurement and compares the results with results obtained by other tools.*

Keywords

IPFIX, monitoring, R-factor, VoIP quality.

1. Introduction

Voice over IP (VoIP) is a technology used to transmit the real-time voice over the packet network built upon IP protocol. Today, VoIP is considered as a cheap alternative to the traditional Public Switched Telephone Network (PSTN) with a wide range of additional services including transmission of both audio and voice data, conferencing, voice mail, interconnection with Internet services like Web, directory services, IM, etc.

The most observed feature of VoIP technology is the quality of voice transmission. Since the VoIP traffic is transmitted over packet-based IP networks, the VoIP quality is influenced by a delay, jitter, or packet loss. There are also additional parameters with impact on

the VoIP quality like selection of a voice codec, acoustic echo, quality of input signal, noise, etc.

There are two different approaches for the voice quality measuring: a subjective speech quality assessment and objective speech quality assessment [1]. *The subjective voice quality* tests are carried out by asking people to grade the quality of speech samples under controlled conditions. The methods and procedures for subjective evaluation are defined by ITU-T Rec. P.800 [2]. For listening-opinion tests, the recommended test method is Absolute Category Rating (ACR) in which the *mean opinion score* (MOS) value is obtained by averaging individual opinion scores for a given number of listeners. MOS uses the five-point opinion scale from 5 (for excellent) to 1 (for bad). A major drawback of subjective assessment methods is that these methods cannot be applied in the real-time monitoring.

The objective speech quality assessment includes intrusive and non-intrusive measurement. *The intrusive measurement* is an active method which needs an injection of a reference speech signal into the tested system where predicts speech quality by comparing the reference and the degraded speech signals. Intrusive objective test methods are sometimes called as "full-reference" or "double-ended" since they compare the original signal at sender's side with a signal measured at the output of the transmission network at receiver's side. Examples of such methods are Perceptual Speech Quality Measure (PSQM) [3], Perceptual Evaluation of Speech Quality (PESQ) [4], or Perceptual Objective Listening Quality Assessment (POLQA) [5].

The non-intrusive measurement is a passive method that computes speech quality by analyzing an IP packet header or by analyzing a degraded speech signal itself. It does not require the original signal. It is mainly used for quality monitoring for operational services. One of the non-intrusive measurement methods called *E-Model* is based on a parametric mathematical model. E-Model stands for European Telecommunication Standards Institute (ETSI)Computational Model that was originally described in [6]. The E-Model takes into account all possible impairments for

an end-to-end speech transmission like a quantization noise, talker/listener echo, absolute delay, type of codec, packet loss, or jitter. Computation of E-Model is specified in ITU-T Rec. G.107 [7]. The result of computation is scalar R that describes quality of voice on scale from 100 to 0. Although non-intrusive methods are less accurate than intrusive methods, they are used for a voice quality assessment.

In our work, we focus on an application of a simplified E-Model for on-line voice quality assessment. The input data for E-Model are obtained from Real-Time Transport Protocol (RTP) packets [8]. VoIP systems are mostly based on two types of application protocols: signalization protocols like SIP, H.323, IAX, or SCCP, and transportation protocols like RTP or RTCP [8]. The signalization protocols provide a phone registration, negotiation of call parameters, call establishment, etc. The transport protocols transmit audio and video data between communicating end points. By monitoring RTP packets and observing RTP control protocol (RTCP) packets we are able to evaluate a speech quality of a given RTP stream.

However, there is an important issue related with RTP monitoring. RTP streams are transmitted over UDP transport using dynamic ports that can be different for each call. Information about dynamic RTP ports is usually transmitted via signalization protocols (SIP/SDP, H.225.0 CS) so that a receiver knows where audio data should be expected. If we are able to detect a signalization protocol, we can also find out RTP streams. There are situations when a signalization protocol uses a different path through the network than a RTP stream. It that case it is very difficult to detect RTP traffic and many monitoring tools are not able to identify VoIP streams and assess their quality. In order to make our monitoring robust, we developed a technique for on-line detection of RTP packets when signalization is missing [9]. This technique can be easily incorporated in our voice quality monitoring system.

Our work is focused on on-line monitoring of RTP flows on a network using IPFIX framework [10], [11]. IPFIX is a monitoring protocol based on Cisco Netflow [12] that collects statistical information about the traffic going through an observation point. Individual packets are grouped into *flows* that are identified by a source/destination IP address, a source/destination port, IP protocol number, ToS class, and an interface ID. An IPFIX probe checks packet headers and creates flow records for incoming packets. The flow record includes the number of packets and bytes of the flow and timestamps of the first and last packet of the flow.

Standard IPFIX records can be extended by user-defined information that are specified using IPFIX templates. In our case, we add information about the quality of VoIP traffic to every RTP or RTCP flow record so that a network administrator can be informed about the presence of VoIP flows and their quality. This simple and effective solution does not require any additional monitoring devices and provides a single-ended speech quality assessment based on E-Model. Data monitoring can be used to detect and identify possible failures on the network that causes a packet delay, loss and degradation of VoIP transmission.

2.   Contribution

The main contribution of this paper is a design of the system for on-line monitoring of VoIP quality based on RTP detection and analysis. The system measures parameters of RTP packets passing through the monitoring system. It also monitors RTCP statistics that give additional information about RTP transmissions. Using RTP parameters and RTCP statistics we are able to compute an average jitter, packet delay and packet loss. Then a simplified E-Model is computed and R-factor with a corresponding MOS value added into an IPFIX flow record of a given voice stream.

When the flow cache expires, the flow is exported to a IPFIX collector. The paper describes how VoIP quality metrics are computed using RTP/RTCP traffic in a monitoring point only. The system was implemented in C as a plugin for the IPFIX probe. In our study, we show the comparison of our tool with Wireshark, PacketScan, and VoIPmonitor. The tests prove that our approach is viable and can be easily incorporated into common monitoring devices. We also propose an extension of this system that is able to detect RTP streams outside conversation. This enables to monitor VoIP data streams if signalization is missing.

2.1.   Structure of the Paper

The structure of the remainder of this paper is as follows. Section 2 describes current approaches in measuring VoIP quality and their possible deployment for on-line monitoring. Section 3 presents QoS metrics that are to be monitored using RTP and RTCP analysis and shows how R-factor is computed from these values. It also describes an extension of IPFIX flow records that is used for transmission of VoIP quality parameters. Section 4 shows comparison of our tool with three other tools for VoIP quality monitoring. The last section concludes the paper and proposes future work.

3.   Related Work

The area of VoIP quality measuring based on non-intrusive techniques has been researched for many

years. One of the pioneering works was done by Cole and Rosenbluth in [13] where basic transport level parameters as a delay, packet loss and de-jitter buffer were discussed and a reduced E-Model was presented. They implemented their model in Perl as a part of SNMP monitoring [14]. Similar approaches can be found later, see [15], [16]. These approaches differ in a way how to compute E-Model parameters in order to get more accurate results. Jiang and Huang in [16] combine intrusive and non-intrusive methods to calculate average time using ICMP probes. It is an active approach in comparison to previous passive methods.

Many recent works combine E-Model with PESQ measuring in order to receive more accurate results. O'Sullivan et al. in [17] present an improvement of the simplified E-Model using correction coefficients for four common codecs (G.711, G.723, G.726, G.729) to better match PESQ scores. A different approach is presented in [18] where the author replaces the payload of the received RTP packets with the payloads that would these packets contained when they had been used to carry test voice signals according to P.50. However, all these methods need to work with an original and a distorted signal. This is not feasible for on-line monitoring. A new methodology for developing perceptually accurate models based on PESQ and E-Model is presented in [19] that computes predicted MOSc from RTP traffic based on measured MOSc from PESQ and E-Model.

There are also works based on ITU-T Rec. P.563 [20] that propose a single-ended method for objective speech quality assessment. Its computation is very complex and includes a reconstruction of a voice stream, signal pre-processing, etc. that cannot be done in a monitoring device. Works like [21] or [22] are mostly focused on a precise quality assessment using a received speech signal rather than on on-line quality measuring.

Monitoring tools like Wireshark observe IP, UDP, and RTP headers. Based on header values only, they compute VoIP statistical data like an end-to-end delay, inter-arrival jitter or cumulative packet loss [8]. This method gives interesting information about the quality of transmission. However, it does not take into account voice features like a codec type, one-way delay, etc.

Our method is also based on a simplified E-Model where its parameters are extracted from RTP headers. R-factor is computed on-the-fly during RTP stream processing. Computation of a simplified E-Model combines several published approaches in order to make it fast and accurate enough for an on-line monitoring using IPFIX architecture. Unlike of work [15], [23] our system includes automatic RTP identification using a set of features and it is fully incorporated into the standard IPFIX monitoring architecture.

4. Monitoring of VoIP Quality

This paper deals with VoIP quality measuring on the packet network using analysis of RTP streams that transmit encoded voice calls. An IPFIX monitoring probe can be connected in any place between communicating parties, see Fig. 1. The probe analyses incoming data, extracts and processes RTP packets, and finally computes R-factor value using the simplified E-Model. Voice quality metrics are added into IPFIX flow records of RTP or RTCP packets and exported via IPFIX protocol to an IPFIX collector.
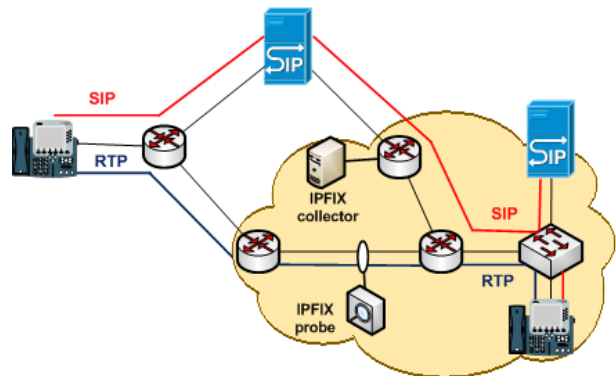

Fig. 1: Architecture of IPFIX monitoring.

Our system is able to monitor packets loss, cumulative jitter, and delay. R-factor is calculated using an Eq. 1 according to ITU-T Rec. G.107 [7]:

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

where R_0 is signal-to-noise ratio, I_S is simultaneous impairment factor, I_d is delay-related impairment, I_{e-eff} is equipment-related impairment, and A is advantage factor. The score obtained from E-Model can be converted to MOS-CQE (MOS conversational quality estimated) according to ITU-T Rec. G.107 [7, Ann. B].

4.1. On-line Computation of E-Model

In our approach, we are more focused on transmission quality of VoIP packets so we consider only wired connections (A=0), standard room or circuit noises and standard impairments on the voice signal (parameter I_S) in our computation. Thus, we use the simplified E-Model with default values recommended in [7, Sec. 7.7]. So, Eq. 1 can be simplified as follows [1]:

$$R = 93.2 - I_d - I_{e-eff}. \qquad (2)$$

In the following text, we show how I_d and I_{e-eff} can be computed on-the-fly.

1)   Computing I_d

As the computational process to obtain I_d according to G.107 is too complicated, a simplified Eq. 3 was proposed in [13]. According to the authors, this function fits the values of I_d within the range of 0–400 ms:

$$I_d = 0.024d + 0.11\,(d - 117.3)\,H\,(d - 117.3)\,, \quad (3)$$

where d is the one-way delay and $H(x)$ is the Heavyside (or step) function defined as $H(x) = 0$ for $x < 0$ or $H(x) = 1$ if $x \geq 0$. By application of linear regression [15], we get modified Eq. 4:

$$I_d = \begin{cases} 0.0267d & d \leq 175 \text{ ms} \\ 0.1194d - 15.876 & 175 \leq d \leq 400 \text{ ms} \end{cases}. \quad (4)$$

The computation of one-way delay d is not an easy task. A common approach is to send the probe packets like ICMP in [16]. Since we are focused on passive (non-intrusive) measurement, we can work with RTP/RTCP packets only. If we have RTCP, one-way delay can be computed from these packets that send periodic reports along with the RTP session. RTCP packets contain an NTP timestamp (TS) with the time at which this RTCP packet was sent, a timestamp of the last sender report received (LSR) and delay since the last sender report received (DLSR). By monitoring RTCP packets, we are able to compute the one-way delays as follows. First, we compute the round-trip delay (RTD) using two adjacent RTCP reports:

$$\begin{aligned} RTD_1 &= delay_1 + delay_2 = \\ &= TS_2 - DLSR_2 - DLSR_1 - TS_1, \end{aligned} \quad (5)$$

where TS_1 is an NTP timestamp when the first RTCP packet was sent, TS_2 is for the second RTCP packet, $DLSR_1$ and $DLSR_2$ is a delay from the last report received, see Fig. 2. In fact, Eq. 5 reflects computation of round-trip propagation delay in RFC 3550 [8] where recording time A corresponds to $(TS_2 - DLSR_2)$ and LSR to TS_1.
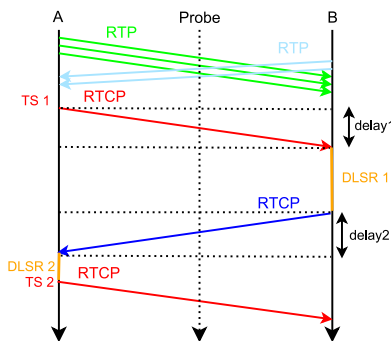

Fig. 2: Computing round-trip delay (RTD) from RTCP.

Since every packet can be routed using a different path, the average RTD delay over all RTCP packets

of the stream is considered. Thus, one-way delay d is given as follows:

$$d = \frac{\sum_{i=0}^{n} RTD_i}{2 \cdot n}. \quad (6)$$

In case of absent RTCP packets we have to use only RTP packets to determine one-way delay. Since the probe can be placed anywhere on the path between communicating parties, we are able to evaluate delay between the sender and the probe only. Precise measuring of the one-way delay of RTP packets is difficult because it requires NTP timestamps with synchronized clocks. However, RTP packets include only a sequence number and a relative timestamp that cannot be used for measurement. In our case, we use an approximated value based on assumption that the average delay relates to cumulative inter-arrival jitter. We use an algorithm implemented in VoIPmonitor, that computes delay d iteratively over subsequent packets, see Eq. 7. Similar approach can be found in [24]:

$$\begin{aligned} d_1 &= 0, \\ d_i &= \frac{(d_{i-1} \cdot i - 1) + J_i}{i}, \end{aligned} \quad (7)$$

where J_i is a cumulative interrarival jitter of packet i. Its value is calculated from the time difference D_{ij} between two adjacent packets i and j and the previous jitter as stated in RFC 3555 [8]:

$$\begin{aligned} J_1 &= 0, \\ J_i &= J_{i-1} + \frac{|D_{i-1,i}| - J_{i-1}}{16}. \end{aligned} \quad (8)$$

Delay $D_{i,j}$ between i and j is given as the difference between RTP timestamps and the times of arrival of these packets:

$$D_{i,j} = (TR_j - TR_i) - (TS_j - TS_i), \quad (9)$$

where TS_i is a RTP timestamp of packet i and TR_i is an arrival time of packet i. Since values TR_i in Eq. 9 represent actual arrival time and values TS_i represent RTP timestamp, these two values must be adjusted by dividing TS_i by sampling frequency for a given codec. Clock rates of RTP codecs are defined by IETF.

2)   Computing I_{e-eff}

An effective equipment impairment factor I_{e-eff} is derived from the equipment impairment factor I_e, the packet-loss robustness factor B_{pl}, burst ratio $BurstR$ and the packet-loss probability P_{pl}. The value of I_{e-eff} can be calculated in Eq.10 according to ITU-T Rec. G.107 [7]:

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{\frac{P_{pl}}{BurstR} + B_{pl}}, \quad (10)$$

where I_e is the equipment impairment factor at zero packet loss which reflects purely codec impairment. Its values depend on subjective mean opinion score test results as well as on network experience. Normally the lower the code bit rate is, the higher the I_e value for the codec is. Recommended values for common codecs are defined in ITU-T Rec. G.113 [25, Appendix I]. B_{pl} is defined as the packet-loss robustness factor which is also codec-specific. It reflects codec's built-in packet loss concealment ability to deal with packet loss. Its value is not only codec-dependent, but also packet-size dependent. B_{pl} values are also listed in ITU-T Rec. G.113. $BurstR$ is the the burst ratio. When a packet loss is independent, $BurstR = 1$, otherwise $BurstR > 1$. Its value is given using a 2-state Markov model with transition probabilities p from "No Loss" to "Loss" states, and q vice verse. Using these probabilities, $BurstR$ can be calculated as [7]:

$$BurstR = \frac{1}{p+q} = \frac{P_{pl}/100}{p} = \frac{1 - P_{pl}/100}{q}. \quad (11)$$

There is another way how to calculate I_{e-eff} based on Pareto/D/1/K modelling of the system proposed by [26]. In this approach, jitter buffer size, codec packetization and network jitter are included into E-Model by means of substitution of packet loss P_{pl} for effective packet loss P_{plef}. This parameter is calculated using Eq. 12 and Eq. 8:

$$P_{plef} = P_{pl} + P_{jitter} - P_{pl} \cdot P_{jitter}, \quad (12)$$

where P_{jitter} is calculated using jitter buffer size x and network jitter J as follows:

$$P_{jitter} = \frac{(1 + \frac{-0,1.x}{J})^{20}}{2}, \quad (13)$$

where x is an input parameter of the system and jitter J is a cumulative inter-arrival jitter computed using Eq. 8. Thus, the calculation of I_{e-eff} as shown in Eq. 10 can be modified using effective packet loss P_{pl} into Eq. 14:

$$I_{e-eff} = I_e + (95 - I_e).\frac{P_{plef}}{P_{plef} + B_{pl}}. \quad (14)$$

In our work, we calculate R-factor using Eq. 2. Delay impairment I_d is computed using Eq. 6 when RTCP packets are found or using Eq. 7 for RTP packets only. Effective impairment factor I_{e-eff} is calculated using effective packet loss P_{plef} as shown in Eq. 14.

4.2. Packet Loss

A packet loss is a ratio between lost packets and expected packets. Number of lost packets is determined from the difference between the number of expected packets and received packets. When calculating the number of expected packets, sequence numbers are used. The number of packets expected can be computed as the difference between the highest sequence number and the first sequence number received. Since the sequence number is only 16 bits wide and will wrap around, it is necessary to extend the highest sequence number with the shifted count of sequence number wraparounds [8, Appendix A.3]. Also duplicated packets create another issue related with a packet loss computation. If we don't check duplicity, duplicated packets can be considered as correctly received and the number of received packets would be misinterpreted.

4.3. Architecture of the IPFIX Probe

The goal of our work is to present a feasible solution for on-line monitoring of VoIP calls using IPFIX. In the previous part, we showed how quality parameters can be calculated with certain approximation from RTP or RTCP packets. Here, we introduce an operational architecture of our system within the IPFIX probe.

General architecture of the probe is shown in Fig. 3. At first, incoming packets are processed in the input plugin where the Call Table is stored. If an RTP packet is detected, it is forwarded into the process plugin where RTP flow records are stored in the flow cache. After the flow expires, it is moved into the export plugin and sent via IPFIX protocol to the collector. More details about the architecture can be found in [27].
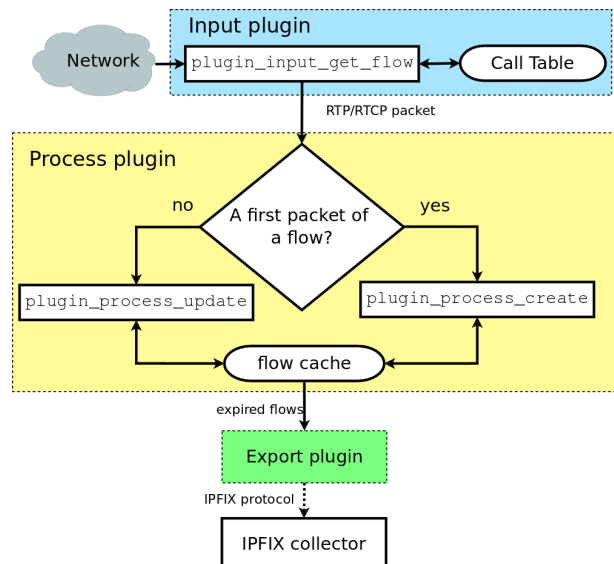

Fig. 3: Architecture of the probe.

Incoming packets are processed in the input plugin as shown in Fig. 4. There are two types of packets expected: signalization SIP packets and RTP/RTCP packets. If a SIP packet arrives, its header is analyzed

and important values added into the Call Table in the input plugin. If an RTP or RTCP packet is received, it is moved into the process plugin, when VoIP metrics of a current call are computed. After finishing the call, VoIP metrics are inserted into extended IPFIX records and sent to the collector.
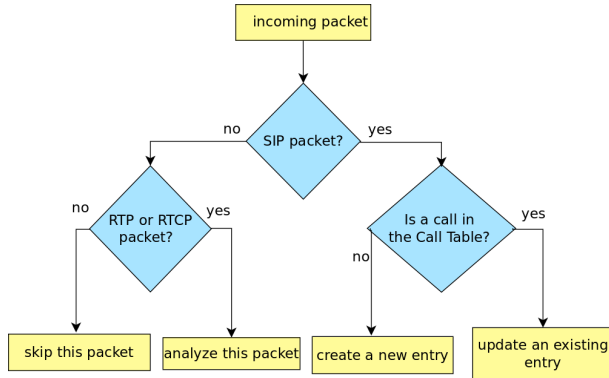

Fig. 4: VoIP packets processing.

Following VoIP metrics are added to RTP/RTCP flow records: type of the flow (RTP or RTCP), jitter, end-to-end delay, packet loss, R-factor, MOS value and the quality of the call based on MOS score, see Tab. 1. An example of IPFIX record extended by VoIP metrics is show in Fig. 5. You can see a newly defined entries with IDs 721 to 727 with entries from Tab. 1 in a hexadecimal format. In Fig. 6, there is an example of human readable output of IPFIX collector showing extended IPFIX records with voice quality metrics.

4.4. RTP Detection

In order to be able to process RTP packets even if signalization is missing we designed a new method for RTP detection [9]. It is a multi-stage filtering method that works with RTP packets first, and than with RTP streams. The method was implemented as an inde-

Fig. 5: IPFIX protocol extended by VoIP metrics.

Tab. 1: Definition of IPFIX entries for VoIP quality metrics.

Entry	ID	Data type	Example
Packet Loss	721	Float	0
Jitter	722	Float	0.201
Delay	723	Uint32	0
R-Factor	724	Float	93.2
MOS	725	Float	4.409
Quality	726	Char	Excellent
FlowType	727	Char	RTP

pendent tool and successfully tested on RTP datasets. The first-stage of processing filters incoming packets by rules based on RTP validity checks [8, Appendix A] and our observations:

- Only IPv4/6 packets with UDP payload are permitted.

- The src/dst ports of UDP must be higher than 1023.

- The length of a packet header must be at least minimal RTP header length according to CSRC Count (CC), i.e., higher than $12 + 4 \times CC$ bytes.

- RTP version must be 2.

- RTP payload type must be within the range defined by RFC 3550. Packets with PT type containing *unassigned* or *reserved* values are filtered out.

- If padding bit P is set, the last byte of the padding is checked with the total length of the packet.

If a packet successfully passes all the above written filtering rules, it is marked as an RTP packet. Then, the second stage of detection using RTP flows observation is applied. This phase helps to decrease number of false negatives for short RTP streams. More details about the method can be found in [9].

5. Tests

This section presents the comparison of our implementation of on-line voice quality monitoring with three other tools: open-source packet analyzer Wireshark, a commercial tool PacketScan from GL Communication Inc. and VoIP analyzer VoIPmonitor. All these tools analyzed the same reference pcap file with G.711 codec. Generally, Wireshark monitors RTP jitter, skew time, delta time, and packet loss. PacketScan calculates conversational MOS, average gap, jitter and packet loss. VoIPmonitor is able to compute MOS for the fixed or adaptive size of jitter buffer and packet loss.

```
Src IPv4:sPort        ->   Dst IPv4:dPort        Flow_Type  Avg_Jitter[ms] Delay[ms] Packet_Loss[%]  R_Factor  MOS    Quality
192.168.2.106:7078    ->   213.168.165.12:13446  RTP        7.910          7         0.000           92.999    4.405  Excellent
192.168.2.106:7079    ->   213.168.165.12:13447  RTCP       76.062         15        0.000           90.091    4.341  Excellent
213.168.165.12:13447  ->   192.168.2.106:7079    RTCP       0.344          11        0.000           92.906    4.404  Excellent
213.168.165.12:13446  ->   192.168.2.106:7078    RTP        0.604          0         0.000           93.200    4.409  Excellent
```

Fig. 6: IPFIX protocol extended by VoIP metrics.

Our tests were done on-line using `tcpreplay` program to send test pcap files through the network. Table 2 shows the results obtained by analysis of RTP or RTCP packets.

Tab. 2: VoIP metrics by Wireshark (W), PacketScan (PS), VoIPmonitor (VPM) and IPFIX plugin (IPF).

Metric	W	PS	VPM	IPF
RTP Jitter (ms)	8.10	7.00	8.00	7.91
RTP Loss (%)	0.00	0.00	0.00	0.00
RTP Delay (ms)	–	0.00	–	0.00
R-factor	–	93.0	–	92.99
MOS	–	4.20	4.50	4.41
RTCP Jitter (ms)	–	7.60	7.61	7.60
RTCP Loss (%)	–	0.00	0.00	0.00
RTCP Delay (ms)	–	9.23	–	15.00
R-factor	–	–	–	90.09
MOS	–	–	–	4.34

In this table, we can see results computed using RTP packets only (the upper part of the table) and using RTCP packets (the lower part of the table). RTP delay is calculated using Eq. 6 for RTP and Eq. 5 using RTCP packets. You can see a great difference between RTP jitter and RTCP jitter. The reason is, that RTCP jitter is calculated by an end-point and sent to the sender while RTP jitter is calculated by an inter-mediated monitoring device. If a device is closer to the sender, jitter will be lower because of lower impact of intermediate network. In this case the probe was placed more likely very close to the sender. From the same reason, there is a difference between RTP and RTCP delay where RTP delay is a delay between the sender and the probe while RTCP delay corresponds to end-to-end delay between communicating end-points. We can also see that neither Wireshark nor PacketScan nor VoIPmonitor compute R-factor using RTCP values. In case of our IPFIX plugin, we can see that R-factor values based on RTP calculation and RTCP calculation are very similar. Due to end-to-end delay approximation in RTP, RTCP R-factor represents more accurate value than RTP R-factor.

There is also a significant difference between RTCP delay calculation in PacketScan and our tool. PacketScan documentation says that round-trip time is computed as $RTD = R_2 - R_1 - DLSR$ where R_2 is an arrival time of a RTCP SR recorded by PacketScan and R_1 is an arrival time of RTCP RR. This means that it is not an end-to-end delay but end-to-PacketScan delay. Thus its value is lower than our value measured between communicating end-points, see Eq. 5.

In the second test, we simulated packet loss by removing random packets from our pcap files using `editcap`. Since RTCP packets were not changed, only RTP calculation reflects packet loss, see Tab. 3.

Tab. 3: VoIP metrics by Wireshark (W), PacketScan (PS), VoIPmonitor (VPM) and IPFIX plugin (IPF).

Metric	W	PS	VPM	IPF
RTP Jitter (ms)	0.20	0.00	1.00	0.0202
RTP Loss (%)	1.50	1.50	–	1.48
RTP Delay (ms)	–	0.00	–	0.00
R-factor	–	93.0	–	80.96
MOS	–	4.20	4.10	4.06
RTCP Jitter (ms)	–	1	3.5	3.65
RTCP Loss(%)	1.5	1.5	1.48	1.48
RTCP Delay (ms)	–	2.65	–	41.00
R-factor	–	–	–	92.11
MOS	–	–	–	4.39

We can see that all tools were able to detect packet loss. R-factor and MOS values are worse for RTP where packet loss was detected in comparison to RTCP R-factor where packet loss was not simulated.

6. Conclusion

In this paper, we presented an improved technique for on-line monitoring of VoIP quality parameters using IPFIX frameworks. Our work includes a design of the monitoring system embedded into an IPFIX probe. The system detects RTP and RTCP packets, analyses their headers, and calculates a jitter, packet loss, delay, R-factor and MOS using the simplified E-Model abstraction on-the-fly. The simplified model uses provisional values for the equipment impairment factor I_e and packet-loss robustness factor B_{pl} as defined in ITU-T Rec. G.113 [25, Appendix I] for well-known codecs. For that reason, codecs detection was implemented as a part of RTP detection [9]. Even this approach is not as precise as PESQ methods, it can be useful for on-line monitoring.

In the future work, we will focus on improvements of end-to-end delay calculation based on [24] and comparison of our results with objective speech quality assessments.

Acknowledgment

References

[1] SUN, L., I.-H. Mkwawa, E. Jammeh and E. Ifeachor. *Guide to voice and video over IP for fixed and mobile networks*. London: Springer, 2013. ISBN 978-144-7149-057.

[2] ITU-T Recommendation P.800. *Methods for subjective determination of transmission quality*. 1996. Available at: `http://www.itu.int/rec/T-REC-P.800`.

[3] ITU-T Recommendation P.861. *Objective quality measurement of telephone-band (300–3400 Hz) speech codecs*. 1996. Available at: `http://www.itu.int/rec/T-REC-P.861/en`.

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

[5] ITU-T Recommendation P.863. *Perceptual objective listening quality assessment*. 2011. Available at: `http://www.itu.int/rec/T-REC-P.863`.

[6] ETR 250. *Transmission and Multiplexing (TM): Speech communication quality from mouth to ear for 3,1 kHz handset telephony across networks*. ETSI, 1996. Available at: `http://www.etsi.org/deliver/etsi_etr/200_299/250/01_60/etr_250e01p.pdf`.

[7] ITU-T Recommendation G.107. *The E-model: a computational model for use in transmission planning*. 2000. Available at: `http://www.itu.int/rec/T-REC-G.107`.

[8] RFC 3550. *RTP: A Transport Protocol for Real-Time Applications*. IETF, 2003. Available at: `http://www.rfc-base.org/rfc-3550.html`.

[9] KMET, M. *Analysis and Detection of Multimedia Types in RTP Traffic*. Brno, 2014. M.Sc. Thesis. Brno University of Technology. Supervisor Petr Matousek.

[10] RFC 7011. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*. IETF, 2013. Available at: `https://datatracker.ietf.org/doc/rfc7011`.

[11] RFC 7012. *Information Model for IP Flow Information Export (IPFIX)*. IETF, 2013. Available at: `https://www.ietf.org/mail-archive/web/ietf-announce/current/msg11896.html`.

[12] RFC 3954. *Cisco Systems NetFlow Services Export Version 9*. IETF, 2004. Available at: `https://www.ietf.org/mail-archive/web/ietf-announce/current/msg11896.html`.

[13] COLE, R. G. and J. H. ROSENBLUTH. Voice over IP performance monitoring. *ACM SIGCOMM Computer Communication Review*. 2001, vol. 31, iss. 2, pp. 9-24. ISSN 0146-4833. DOI: 10.1145/505666.505669.

[14] RFC 1155. *Structure and Identification of Management Information for TCP/IP-based Internets*. IETF, 2001. Available at: `https://www.ietf.org/rfc/rfc1155.txt`.

[15] VOZNAK, M. Recent Advances in Speech Quality Assessment and Their Implementation. In: *AETA 2013: Recent Advances in Electrical Engineering and Related Sciences*. Saigon: Springer, 2014, pp. 1–14. ISBN 978-3-642-41967-6. DOI: 10.1007/978-3-642-41968-3_1.

[16] JIANG, C. and P. HUANG. Research of Monitoring VoIP Voice QoS. In: *International Conference on Internet Computing & Information Services (ICICIS)*. Hong Kong: IEEE, 2011, pp. 499–502. ISBN 978-1-4577-1561-7. DOI: 10.1109/ICICIS.2011.130.

[17] ASSEM, H., D. MALONE, J. DUNNE and P. O'SULLIVAN. Monitoring VoIP call quality using improved simplified E-model. In: *International Conference on Computing, Networking and Communications (ICNC)*. San Diego: IEEE, 2013, pp. 927–931. ISBN 978-1-4673-5287-1. DOI: 10.1109/ICCNC.2013.6504214.

[18] CONWAY, A. E. A passive method for monitoring voice-over-IP call quality with ITU-T objective speech quality measurement methods. In: *International Conference on Communications*. New York: IEEE, 2002, pp. 2583–2586. ISBN 0-7803-7400-2. DOI: 10.1109/ICC.2002.997309.

[19] LINGFEN SUN and E.C. IFEACHOR. Voice quality prediction models and their application in VoIP networks. *IEEE Transactions on Multimedia.* 2006, vol. 8, iss. 4, pp. 809–820. ISSN 1520-9210. DOI: 10.1109/TMM.2006.876279.

[20] ITU-T Recommendation P.563. *Single-ended method for objective speech quality assessment in narrow-band telephony applications.* 2004. Available at: http://www.itu.int/rec/T-REC-P.563/en.

[21] MALFAIT, L., J. BERGER and M. KASTNER. P.563——The ITU-T Standard for Single-Ended Speech Quality Assessment. *IEEE Transactions on Audio, Speech and Language Processing.* 2006, vol. 14, iss. 6, pp. 1924–1934. ISSN 1558-7916. DOI: 10.1109/TASL.2006.883177.

[22] FALK, T. H. and W.-Y. CHAN. Single-Ended Speech Quality Measurement Using Machine Learning Methods. *IEEE Transactions on Audio, Speech and Language Processing.* 2006, vol. 14, iss. 6, pp. 1935–1947. ISSN 1558-7916. DOI: 10.1109/TASL.2006.883253.

[23] TOMALA, K., L. MACURA, M. VOZNAK and J. VYCHODIL. Monitoring the quality of speech in the communication system BESIP. In: *35th International Conference on Telecommunications and Signal Processing (TSP).* Prague: IEEE, 2012, pp. 255–258. ISBN 978-1-4673-1117-5. DOI: 10.1109/TSP.2012.6256293.

[24] NGAMWONGWATTANA, B. and R. THOMPSON. Sync: VoIP Measurement Methodology for Assessing One-Way Delay Without Clock Synchronization. *IEEE Transactions on Instrumentation and Measurement.* 2010, vol. 59, iss. 5, pp. 1318–1326. ISSN 0018-9456. DOI: 10.1109/TIM.2010.2043978.

[25] ITU-T Recommendation G.113. *Transmission impairments due to speech processing.* 2002. Available at: https://www.itu.int/rec/T-REC-G.113.

[26] HALAS, M., A. KOVAC, M. ORGON and I. BESTAK. Computationally efficient E-model improvement of MOS estimate including jitter and buffer losses. In: *35th International Conference on Telecommunications and Signal Processing (TSP).* Prague: IEEE, 2012, pp. 86–90. ISBN 978-1-4673-1117-5. DOI: 10.1109/TSP.2012.6256258.

[27] Basel, M. *Quality Analysis of VoIP Calls.* Brno, 2014. B.Sc. Thesis. Brno University of Technology. Supervisor Petr Matousek.

About Authors

Petr MATOUSEK was born in 1973 in Usti nad Labem. He received his M.Sc. in 1997 and Ph.D. in 2005 at Brno University of Technology where he is an assistant professor since 2003. His professional career includes interships at CERN, Geneva, and at LIAFA, Paris. His research interests include formal specification and verification, data analysis, network management and VoIP technologies.

Martin KMET was born in 1990 in Nove Mesto nad Vahom, Slovakia. He received his M.Sc. at Brno University of Technology, Faculty of Information Technology in specialization Computer Networks and Communication. Since 2014, he is a Ph.D. student there. His research is focused on VoIP technologies.

Martin BASEL was born in 1992 in Zlin. Currently, he studies his MSc. at Faculty of Information Technology, Brno University of Technology. In 2014, he successfully defended his bachelor project named Quality Analysis of VoIP calls.