Dynamic Codec Selection Algorithm for VoIP

Maja Sulovic
Faculty of Electrical Engeneering
Department of Telecommunications
Sarajevo, Bosnia and Herzegovina
maja.sulovic@etf.unsa.ba

Darijo Raca
Faculty of Electrical Engeneering
Department of Telecommunications
Sarajevo, Bosnia and Herzegovina
draca@etf.unsa.ba

Mesud Hadzialic
Faculty of Electrical Engeneering
Department of Telecommunications
Sarajevo, Bosnia and Herzegovina
mhadzialic@etf.unsa.ba

Nasuf Hadziahmetovic
Faculty of Electrical Engeneering
Department of Telecommunications
Sarajevo, Bosnia and Herzegovina
nhadziahmetovic@etf.unsa.ba

Abstract—**This paper gives a proposal on algorithm for adaptive adjustment of VoIP sources transmission rate based on voice quality estimated at the receiver. This adjustment is achieved through the appropriate use of differing voice codecs, as changes in network conditions occur, in order to maintain an efficient utilization of the available resources. We propose a simple algorithm for dynamic selection of voice codec, depending on network conditions during the on-going voice session. Algorithm is embedded in the source code of the programming environment OPNET Modeler 14.5. Simulation results show that the proposed algorithm makes better use of the available bandwidth, achieving superior performance in comparison to the situation without implementation of algorithm for adaptive codec selection.**

Keywords-dynamic codec selection; speech codec; packet delay; packet loss; MOS

I. INTRODUCTION

Nowadays, VoIP is widely accepted technology used for transmission of voice over IP networks. Packets belonging to the same VoIP session may travel independently through different paths in the network. In this paper various codecs such as G.711, G.729 and G.723 will be discussed. Each of these codecs use a different technique for sample coding and different levels of compression that directly affects the voice session quality. Selection of the right codec represents a compromise between desired performance and available resources in the network.

In this paper, as criteria for determining the quality of the voice session we use E-model, which estimates user's satisfaction through R value, on the basis of the used codec, delay and loss in the network. Since metrics values on the network level in IP network are variable, this will practically mean that during the session, same codec may not be optimal in all moments. There are different techniques used for solving this problem, e.g., AMR (Adaptive Multi Rate) codecs, implementation of additional access mechanisms at application layer, the dynamic selection of codec during the sessions and others. This paper addresses the last mentioned technique, and introduces a simple algorithm, which estimates the optimal codec during voice session, based on current and past values of average delay and current value of packet loss.

As authors of this paper, we are familiar with research work dealing with this problem. In [1], codec selection algorithm is based on average delay value which is then compared with current delay value. Algorithm verification is based on comparison between different amount of available bandwidth scenarios. It has been shown that growth of available bandwidth as a consequence, increases MOS value when using adaptive codec selection method. Similar results are also shown in [2, 3]. In [4], authors proposed adaptive multi-rate VoIP control scheme that adapts the voice encoding rate and packetization interval in relation to transmission rate in the PHY layer. In [5] authors proposed algorithm which, based on information extracted from RTCP packets and MAC layer, dynamically adapt codec for ongoing VoIP calls. In [6] authors described an end-to-end based adaptation, which adjusts application parameters to changing network conditions in order to achieve better bandwidth utilization and QoS, by employing adaptive codec switching techniques to further enhance QoS.

In this paper, the called party of a voice session is measuring the average packet delay and loss. It periodically sends reports to the calling party that is responsible for session management. Based on the received reports, algorithm that is implemented on the calling party learns about the network by memorizing the minimum and the maximum value of average packet delay and packet loss for each codec that has been used in the session. Memorized values are then compared with reported average delay and packet loss. If the algorithm determines that there are improvements of network conditions (e.g., reported average delay is less than the minimum average delay for currently used codec that algorithm has knowledge of until the moment of observing), the algorithm makes a decision about switching to a new codec with lower compression ratio. Based on the feedback in the following report algorithm decides whether the previous decision on codec change was correct or not. Incorrect decisions will occur in situations when despite new minimum average delay being detected by the algorithm, available resources in the network are not sufficient enough to use codec with lower compression ratio. All incorrect codec switching decisions will then reflect in the following report, through reported average delay that is higher than the maximum delay allowed for the voice session. The algorithm then concludes that value of the minimum delay, previously memorized, should not be used as threshold value for switching to a better codec and the session codec will switch to a codec with a higher compression ratio, thus enabling the algorithm to recognize similar situations in the future and prevent incorrect codec switch decisions from occurring.

The proposed algorithm was implemented in simulation package OPNET Modeler 14.5. Used version of the Modeler does not have similar algorithm for dynamic codec selection that has been implemented. To the best author knowledge there is no similar solution to this problem.

II. OVERVIEW OF SPEECH CODECS

There are various codecs specified by the ITU-T. Codecs have different performance and impact on the voice quality due to different degree of compression. High degree of compression results with higher compression delay and increases loss sensitivity compared to codecs with low or no compression. Contrary to this, codecs with high degree of compression have less bandwidth requirements, and thus have better performance in network congestion situations. Therefore, it is necessary to select the appropriate codec to obtain best quality of voice with the lowest bandwidth requirements [7].

The G.711 codec does not use any compression; it has 8-kHz sampling rate, requires 64 kbit/s of audio bandwidth and provides very good quality level. The G.729 codec is computationally complex, but provides significant bandwidth savings. It has 8:1 compression and requires just 8 kbit/s of audio bandwidth. The maximum achievable MOS is about 3.9. The G.723.1 codec is mostly used in VoIP applications due to its low bandwidth requirement. There are two versions of this codec, with bit rates at 5.3 kbit/s and 6.3 kbit/s. Every codec adds additional delay to the total packet transmission delay due to signal encoding, decoding, compression and decompression. Main characteristics of the codecs mentioned are shown in Table 1 [8, 9, 10].

TABLE 1. CHARACTERISTIC OF CODECS

Codec	Bit Rate (kbit/s)	Link Utilization (kbit/s)	Delay (ms)	Loss (%)	MOS
G.711	64	87.2	0.125	7-10	4.10
G.729	8.0	31.2	15	< 2	3.92
G.723.1	5.3	20.8	37.5	< 1	3.65

III. IMPACT OF NETWORK LAYER METRICS ON QUALITY OF SPEECH

QoS concept is observed as layered model and is defined at user, application and system layer. Quality of service at observed layer of QoS layered model represents characterization of expected quality, which should be achieved during transfer of data units. Between any two layers, it is important to determine mapping between expected performance at lower layer and its impact on QoS parameters on higher layer. The main question is how given QoS guarantees, at layer N-1, impact performance metrics at layer N [11, 12].

ITU-T E-model represents an analytical model of voice quality defined in the ITU-T recommendation G.107. E-model provides a framework for real-time on-line quality estimation from network performance measurement (e.g., delay and loss characteristics) and application level factors (e.g., low bit rate codecs). The result of the E-model is the calculation of the R-factor. The R-factor can be further translated into MOS scale through these expressions:

$$MOS = \begin{cases} 1 & R < 0 \\ 1 + 0.035R + \\ 7.10^{-6}R(R-60)(100-R) & 0 < R < 100 \\ 4.5 & R > 100 \end{cases} \quad (1)$$

The R-factor is defined as:

$$R = R_0 - I_s - I_d - I_e + A \quad (2)$$

R_0 represents the basic signal-to-noise ratio. I_s reflects the impairments occuring simultaneously with the voice signal due to quantization. It is a function of several parameters, none of which are related to the underlying packet transport. I_d models the impairments caused by one-way delay. Voice quality degrades more rapidly when this delay exceeds 177.3 ms. This effect is modeled using following expression:

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \quad (3)$$

where d is the one-way delay (in miliseconds) and $H(x) = 0$ za $x < 0$ i $H(x) = 1$ when $x \geq 0$. I_e is the equipment impairment factor that covers the distrortion of the original voice signal due to low-rate codec and packet loss in both, network and playout buffer. I_e value is codec dependable. The advantage factor A represents the measure of the willingness of a VoIP user to trade call quality for convenience. [13]

VoIP application is very sensitive to delay. Acceptable one-way delay according to ITU-T G.114 recommendation is 150 ms. Delay between 150 ms and 400 ms makes the conversation possible, but considerably annoying. Delay over 400 ms is unacceptable according to ITU-T G.1010. Packet loss also must be managed or controlled in VoIP, since its effect on VoIP is treated as noise. Unlike delay, VoIP can tolerate packet loss to some extent. [14]

IV. Proposed Algorithm for Dynamic Speech Codec Selection

After review of most commonly used codecs and QoS metrics which evaluate performance and dependencies between QoS metrics from different layers of layered QoS model, this chapter presents details of proposed algorithm for dynamic codec selection.

A. Algorithm Design

Next steps present design of proposed algorithm in detail:

1. Define initial minimum delay values for codecs G.729A and G.723.1 5.3K
2. Define initial maximum delay values for codecs G.711 and G.729
3. Define initial maximum loss values for codecs G.711, G.729A and G.723.1 5.3K
4. Store the value of currently used codec and its average delay from report in an array that stores up to three previously used codecs and their corresponding delays from previous reports.
5. If codec changes occurred at least three times until current moment of the call session, then determine new minimum average delay if possible
    - 5.1. If current and penultimate used codecs are the same, go to step 5.1.1, else go to step 6
        - 5.1.1. If previously stored average delay is less than the current average delay, go to step 5.1.2, else go to step 6
        - 5.1.2. store average delay from the previous report into variable minimum average delay for codec used during the receiving of the previous report
6. Compare received average delay and loss with minimum average delay, maximum average delay and maximum packet loss
    - 6.1. If average delay is less than minimum average delay, change codec according to:
        - 6.1.1. If G.711, then G.729A
        - 6.1.2. If G.729, then G.723.1 5.3K
    - 6.2. If average delay is higher than maximum average delay, change codec according to:
        - 6.2.1. If G.723.1 5.3K, then G.729A
        - 6.2.2. If G.729A, then G.711
    - 6.3. Return to step 4

Step 1 defines initial minimum average delay for G.723.1 5.3K and G.729A codecs. Algorithm compares average delay from the report and minimum average delay value to assume whether network conditions have been improved. If so, algorithm assumes that there has been a release of resources in the network and switches to the codec with higher bandwidth requirements in order to improve speech quality that is expressed through the MOS value.

Steps 2 and 3 are analogous to step 1. They define maximum value of packet delay and packet loss for codecs G.711 and G.729. If average delay and loss from the currently received report are higher when compared to stored maximum values, algorithm concludes that there is network congestion occurrence, and that is necessary to use a codec that requires less bandwidth.

In step 4, algorithm learns about conditions in the network. Algorithm stores the information about three last used codecs and its resulting average delays in the array.

Step 5 is the crucial step of the algorithm. It will be explained using the following example. If voice application uses codec G.711 and if average delay from current report is higher than the maximum allowed delay for G.711, algorithm switches the session codec to one that requires less bandwidth (G.729A). Next statistical report will indicate latency reduction due to the fact that G.729A codec has lower bandwidth consumption and thus lower contribution on total network load on bottleneck. Upon receiving this report, algorithm compares last three stored delays and may assume that there was a release of resources in the network, and try to re-use the codec G.711. However, if there has been no release of resources in the network, delays in the forthcoming report for the codec G.711 will again have a high value. This way, algorithm calculates delay value for G.729A which should not trigger codec change and updates the variable that stores the minimum delay for codec G.729A. Without this step, algorithm would trigger codec change upon receiving any statistical report that is indicating latency reduction. Application would switch session codec to a codec with lower compression ratio and that would again worsen the situation in the bottleneck of the network. Described situation resulting effect would be a constant switching from codec to codec.

Step 6 includes decision making on the optimal codec for current conditions in the network. The algorithm compares average delay from last received report with minimum and maximum delay and maximum packet loss defined for currently used codec. If current average delay exceeds maximum delay or maximum packet loss, algorithm switches the session codec to one with lower bandwidth requirements. Contrary to this, if current average delay is below minimum delay, algorithm switches the session codec to one with higher bandwidth requirements.

B. Implementation in OPNET Modeler 14.5

This section provides a brief description of the proposed algorithm implementation in the OPNET Modeler. To implement dynamic codec selection algorithm in the OPNET Modeler, we had to modify process models, which implement the calling party as well as the called party. Process in OPNET Modeler is modeled as a finite state machine (FSM) that allowed us to specify the C/C++ code that implements a process.

 ISBN: 978-1-61208-127-4

For sending received statistics report from the called party to the calling party, firstly received statistics (average delay and loss) on the called party is collected and afterwards collected data is forwarded to the calling party. Statistical data are collected in “receive” state in the called party process editor, see Figure 1. Upon receiving voice packet, “receive” state calculates current average packet delay and packet loss. Report sending interval is set to 5 seconds. Once this interval is ended, process model that represents called party switches its state and enters into state responsible for sending collected data to calling party. If average delay exceeds 300 ms before 5 s interval is completed, called party sends additional early report for fast adaptation to network conditions change. For forwarding statistical data to the calling party a new “send report” state is created, within called party process. Newly created state will be responsible for sending periodic reports that are sent via RTP.

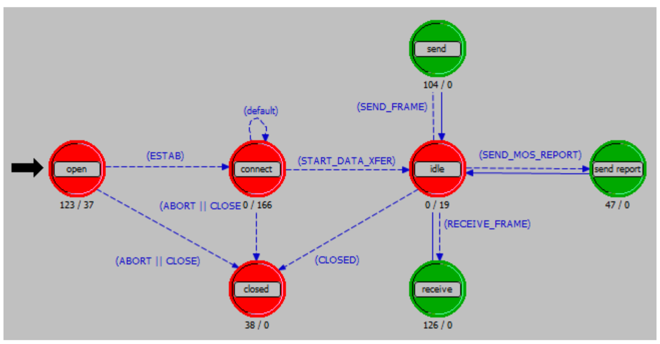

Figure 1. Called party process editor

Algorithm described in Section 4.1 is implemented in “receive” state in the calling party process editor. If algorithm determines that the codec in use needs to be changed, calling party changes currently used session codec. From that moment on, every voice packet sent to the called party includes information about new session codec, which is stored in the packet header. [15]

V. SIMULATION RESULTS

Simulation of simple network containing two end nodes is created for the purpose of verification of the proposed algorithm. Our goal was to create three cases where available bandwidth is optimized for voice traffic transmission with only one of the mentioned codecs. Nodes support voice and video applications. The voice application is set to default OPNET setting IP Telephony. Video applications settings are set to achieve mentioned goals, and can be seen in Table 2. Thus we created three profiles, VoIP Profile, Video High Load Profile and Video Low Load Profile. Video High Load Profile bitrate is 20 kbit/s; Video Low Load Profile bitrate is 10 kbit/s.

Elements presented in Table 3 are used in the simulation. Network consists of two core routers. Link between these routers has a capacity of 160 kbit/s, and represents a bottleneck of the topology. Each core router has 10 Mbit/s connection with a switch. Each switch has 10 Mbit/s link connection with workstation. Link between routers has FIFO queuing implemented, where maximum number of packets in the buffer is 100. In order to reduce simulation duration, we choose to simulate simple network topology. Therefore, although our network topology is very simple, our simulation results are valid and can, with proper equipment, be tested in more realistic case studies. Measured metrics in this simulation are average delay at the network level, sent/received number of bytes and MOS value.

TABLE 2. ELEMENTS USED IN SIMULATION

Device Name	Device Description
CS 7609	Core routers in simulation
CS 6509	Access switch in simulation
ethernet_wkstn	This node model represents a workstation with client-server applications running over TCP/IP and UDP/IP
10BaseT	Connection between workstation and switch
ppp_adv	Data Rate is 160000; Connection between core routers

Figure 2 shows a comparison between delays in the cases when using a fixed codec and when using a dynamic codec selection. Initial codec in the last case is codec G.729A. It can be seen that during the Video High Load profile codecs G.711 and G.729A have unacceptable delays which are approximately 1.1 s and 0.7 s, respectively. After the termination of the Video High Load profile, delay values in the case when using fixed G.711, G.729A and G.723.1 5.3K codecs have acceptable delay values below 100 ms. When new video profile Video Low Load starts, delay value for G.711 codec increases its value to approximately 0.9 s, while delay values for G.723.1 3.5K and G.729A codecs increase to 200 ms at most. In fourth case, which relates to the application of the proposed algorithm for dynamic codec selection, Figure 2 clearly shows moments of codec change and adjustment to network conditions. It may be noted that in this case, delay during Video High Load profile is approximately equal to the delay when using fixed G.723.1 5.3K codec. This is due to the fact that the algorithm chose G.723.1 5.3K codec as the optimal codec. At the time of termination of the Video High Load profile, there is a gradual change of codecs, first with the change from G.723.1 5.3K to G.729A and afterwards from G.729A to G.711. After Video Low Load profile starts, slight increase in delay value can be observed, after which the algorithm concludes that there might be congestion in the network, and switches codec from G.711 to G.729A.

Figure 3 shows the MOS value in cases analogous to those with previous images. One can clearly see the advantage of using an algorithm for dynamic codec selection, where the MOS value in all the session moments

 ISBN: 978-1-61208-127-4

is approximately equal to the maximum MOS values in the other three cases.

If we observe the percentage of lost packets shown in Table IV, the worst results are obtained by using the G.711 codec, where packet loss is 7%. G.729A and G.723.1 5.3k codecs both achieve packet loss of 1%. Application of the proposed algorithm for dynamic codec selection results with no voice packet loss. Average values for delay and MOS shown in Table 4 will provide a more complete picture of the benefits of the proposed algorithm. Using proposed algorithm, we obtain best results in terms of MOS values 2.44, compared to other values obtained for G.711, G.729A and G.723.1 5.3k, respectively. The same conclusion applies to average delay, where using the algorithm proposed results with 189.9 ms delay, whereas other cases result with higher delay values.

TABLE 3. AVERAGE DELAY AND PACKET LOSS

	G.711	G.729A	G.723.1 5.3K	Dynamic Codec Selection
Delay (ms)	728.8	262.0	206.2	189.9
Packet loss (%)	7	1	1	0
MOS	1.5120	2.1488	1.9419	2.4346

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented standard voice codecs used for transmission of voice traffic over the Internet. Basic characteristics of each of the discussed codec, its advantages and disadvantages are expressed through QoS metrics and average delay and loss at the network level, which ultimately results in the uniform assessment of quality, represented by the MOS value at user's level. Main goal for a voice over Internet session is to meet QoS recommendations and to simultaneously achieve the highest possible MOS value. In network congested condition, low rate codec has better performance than high rate codec. In contrast, when there is no network congestion, high rate codec has better performance. This means that the best network performance could be achieved through trade-off between codec bandwidth requirements and desired quality. In other words, voice session will adapt sending rate to available bandwidth. Since usage of fixed codec during one voice session cannot achieve this trade-off, we proposed a simple algorithm for dynamic selection of the codec depending on the current conditions in the network. We showed that use of a proposed algorithm maintains high level of MOS value in cases of network congestion.

In [3], main goal is to reduce delay when congestion occurs, and algorithm is based on TCP Vegas-like congestion avoidance technique, for the rate and loss control of VoIP flows over the WLAN. The idea for our algorithm proposal is also based on delay and packet loss similar to [3], but, unlike aforementioned, it does not make difference between congestion loss and error loss of packets (due to wireless environment). Also, in [3], authors do not analyse impact of decision of algorithm on final MOS values. The algorithm in [3] is tested with more than one parallel calls, and also takes into account fairness between VoIP calls. Because of nature of our simulation, which is simple, we do not take this into account. Our algorithm is similar to one proposed in [5], but with one main difference. All of these algorithms have memoryless property that means that they do not memorize previous network conditions states. Our algorithm takes this parameter into account and based on that makes further decisions.

The next step is to examine the influence of report sending interval length on the adaptation process, which actually represents a trade-off between desired quality of conversation and adaptation rate to network changes. In future work, we intend to investigate impact of proposed algorithm in wireless environment and create real application based on proposed algorithm. Also we intend to simulate more than one call in the same time and analyse fairness impact on flows as it is done in [3].

REFERENCES

[1] A. Ossipov, “Dynamic Voice Codec Determination Mechanism for VoIP,“ MS Thesis, Wichita State University, Kansas, USA, May 2006.

[2] A. Servetti and J.C. De Martin, “Adaptive interactive speech transmission over 802.11 wireless LANs,” Proc. IEEE Int. Workshop on DSP in mobile and Vehicular Systems, Nagoja, Japan, April 2003., pp. 1-4.

[3] A. Trad, Q. Ni, and H. Afifi, “Adaptive VoIP Transmission over Heterogeneous Wired/Wireless Networks,“ 2nd International Workshop MIPS 2004, LNCS, Vol. 3311, Grenoble, France, November 2004, pp. 25-36.

[4] T. Kawata and H. Yamada, “Adaptive Multi-Rate VoIP for IEEE 802.11 Wireless Networks with Link Adaptation Function,“ Global Telecommunications Conference, GLOBECOM '06. IEEE, San Francisco, USA, December 2006, pp. 1-5.

[5] A. Sfairopoulou, C. Macián, and B. Bellalta, “VoIP codec adaptation algorithm in multirate 802.11 WLANs: distributed vs. centralized performance comparison,“ LNCS, 2007, Volume 4606/2007, Barcelona, Spain, pp. 52-61.

[6] See Leng Ng, S. Hoh, and D. Singh, “Effectiveness of adaptive codec switching VoIP application over heterogeneous networks,“ IEEE 2nd International Conference on Mobile Technology, Applications and Systems, Guangzhou, China, November 2005, pp. 1-7.

[7] H. Kazemitabar, S. Ahmed, K. Nisar, A.Said, and H. Hasbullah, “A comprehensive review on VoIP over Wireless LAN networks,“ ISSR Journal, Vol. 2, No. 2, September 2010, pp. 1-16.

[8] ITU-T Recommendation G.711, “Pulse code modulation (PCM) of voice frequencies,“ 1988.

[9] ITU-T Recommendation G.729, “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP),“ 1996.

[10] ITU-T Recommendation G.723.1, “Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s,“ 2006.

[11] Kun I. Park, “QoS in Packet Networks,“ Springer, October 2004.

[12] T. Braun, M. Diaz, J. Enriquez-Gabeiras, and T. Staub, “End-to-End Quality of Service Over Heterogeneous Networks," Springer, October 2008.

[13] ITU-T Recommendation G.107, “The E-Model, a Computational Model for Use in Transmission Planning,“ March 2005.

[14] K. Perlicki, “Simple analysis of the impact of packet loss and delay on voice transmission quality,“ Journal of Telecommunications and Information Technology, February 2002, pp. 53-56.

 ISBN: 978-1-61208-127-4

[15] "OPNET Modeler Documentation Set," Version 14.5, 2010.

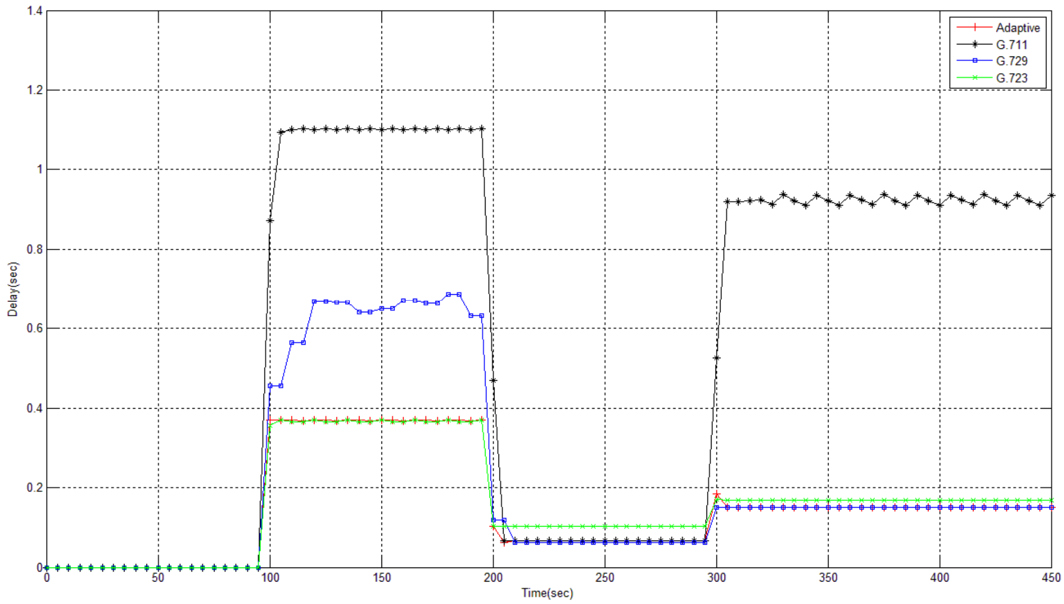

Figure 2. Delay comparison between using fixed codec and adaptive selection of speech codec

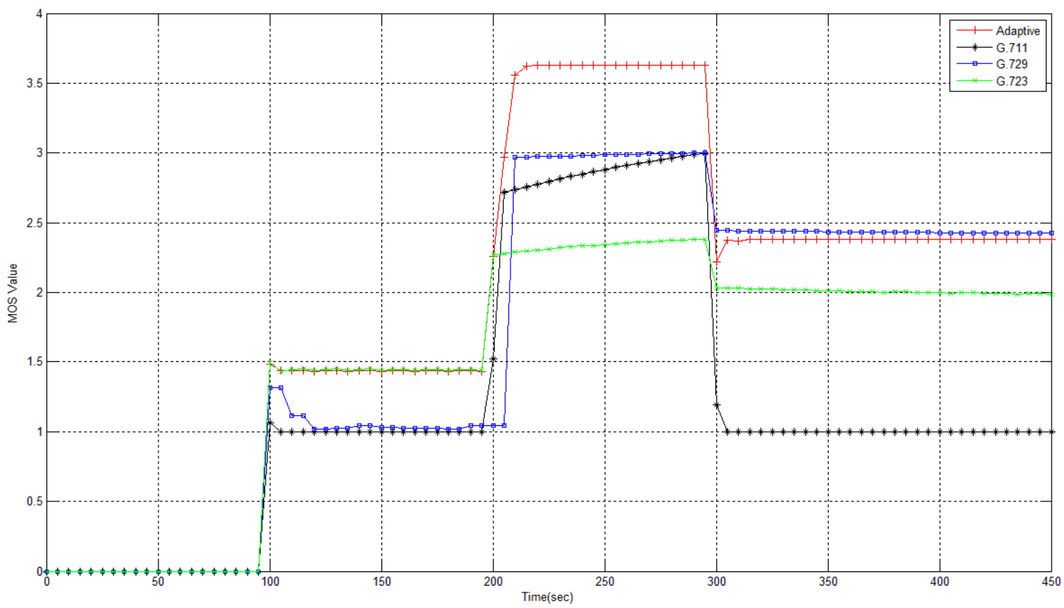

Figure 3. Obtained MOS value