Evaluating Quality of Encrypted VoIP Calls in a Simulation Environment

Dhérík Barison¹, Rodrigo S. Miani¹, Gean D. Breda¹, Leonardo S. Mendes¹ and Bruno B. Zarpelão²

¹ School of Electrical and Computer Engineering (FEEC), University of Campinas (UNICAMP)
Campinas, Brazil
{dherik, rsmiani, gean, lmenedes}@decom.fee.unicamp.br

²Computer Science Department, State University of Londrina (UEL)
Londrina, Brazil
brunozarpelao@uel.br

Abstract: The purpose of this work is to evaluate the quality of encrypted VoIP calls with different encryption algorithms through OpenVPN software, in order to identify differences in results between encryption algorithms and also differences between non-encrypted and encrypted calls. This evaluation will take into account the MOS (Mean Opinion Score), a method to indicate user satisfaction of voice communication quality. The encrypted VoIP calls will occur in several network scenarios, each one with different network bandwidth configurations and problems, like packet loss, out-of-order packets, and delay. These network scenarios are constructed in a simulation environment based on VirtualBox virtualization software, Linux Traffic Control tool and IXChariot performance assessment tool.

Keywords: MOS, VPN, Linux Traffic Control, IXChariot, OpenVPN.

I. Introduction

The telephony system has been coming through a profound transformation. As the world is more and more connected to Internet, the VoIP (Voice over IP) technology is being more sought-after to perform voice communication. The great advantage of this system, when compared to the conventional telephony system, lies in its flexibility and lower cost [1].

In VoIP systems, security is an important issue [2]. Since VoIP is a TCP/IP application, DoS (Denial of Service) and DDoS (Distributed Denial of Service) are serious threats to VoIP services reliability [3]. Moreover, in the same way as it occurs in the conventional telephony system, VoIP systems are also vulnerable to eavesdropping. By using the TCP/IP architecture, the VoIP technology allows this interception to be achieved in a simpler way than the way the analogical telephony system does, as it demands physical access to the telephonic network wiring. The interception of VoIP calls can be done from anywhere in the network, and if this network is the Internet, this can be done from anywhere in the world. The wire-tapping can be performed with an efficient packet sniffer, e.g. the Wireshark [4], which is able not only to get the packets but also to put them together and to reproduce the audio content of the conversation.

If the environment in question is a corporative network, and the data exchanged between the terminals are not secured, any employee can have easily access to any type of information, including internal connections carried out via VoIP. In these conditions, it is necessary to assure protection to such voice data. A commonly solution adopted by the companies is the implementation of encrypted virtual private networks as a way to protect these confidential data.

However, a problem can be detected when using cryptography. The delay generated by the encryption/decryption process can be perceived by users, causing a negative impact in the quality of voice communication. Besides that, there are the usual problems that often affect VoIP calls like packet loss, packets duplication, packet corruption, out-of-order packets, delays, and jitter. Therefore, it is prudent to analyze the impact of different encryption algorithms on the quality of VoIP calls, taking into account some scenarios where the network is showing good traffic conditions and scenarios where the network faces problems derived from congestion.

Our work is based on this context. We have built a simulation environment using three tools: OpenVPN [5], Traffic Control [6] and IXChariot [7]. OpenVPN and IXChariot are used to establish the encrypted VoIP calls and also to measure their performance. Traffic Control is used to emulate many of common problems found in a network, such as: packet loss, out-of-order packets, packet duplication, delay and jitter. Seven scenarios will be created and they will bring different configurations concerning to congestion, bandwidths, packet loss, packet duplication, delays, jitter and out-of-order packets. These scenarios are based, mainly, on Markopolou et al. [8] and Snyder [9] works. The objective is to evaluate the behavior of different encryption algorithms, in these scenarios, and to determine the extent to which the communication quality may be affected. We are also interested in investigate whether some of the encryption algorithms are more efficient than others, in some particular network situations. The quality of VoIP calls will be measured by using the Mean Opinion Score (MOS) [10], a method already established by the ITU-T (Telecommunication Standardization Sector) which enables a reliable evaluation of
quality of VoIP calls.

The present paper is organized as follows. Section 2 describes the related work. Section 3 presents background information about methods for VoIP calls quality evaluation. Section 4 brings details about the environment built for the tests, scenarios which emulate different problems in the networks and also how the experiments were carried out. Section 5 reveals the results of the experiments introduced in the section 4. Section 6, at last, shows the final considerations on the current paper.

II. Related Work

With respect to works on voice quality, Barbosa et al. [11] conducted experiments similar to ours, concerning to the use of different network scenarios and evaluation of quality of the calls. The authors have assessed the performance of calls realized by VoIP P2P tools like Skype and Google Talk in networks under different conditions, involving packet loss, delays and link capacity. The PESQ (Perceptual Evaluation of Speech Quality) algorithm was employed to calculate the MOS from the comparison between the audio reproduced in the destination and the audio in the origin. The authors made use of the NIST.net [12] tool to emulate the problems in the network and the bandwidth whereas we employed the Traffic Control for these tasks, instead.

Voznák [13] evaluated the impact of cryptography on the quality of VoIP communication, by using a set of tools similar to the ones we used in our work, employing the OpenVPN and the IxChariot. Yet, its focus was on the consumption alterations of the network bandwidth when the codec and the encryption algorithm were modified. Nappa et al. [14] have accomplished an analogous work, however, without putting emphasis on the bandwidth consumed in the final results. Nappa et al. also employed the OpenVPN and IxChariot, along with the iPerf program [15] to generate traffic in the network.

Markopoulou et al. [8] performed tests to check the VoIP communications quality in the Internet backbones. In these tests, he has examined the anomalies which were affecting various providers in the United States, making possible to look at the occurrence of delay, jitter, packet loss rate, and congestion. We used these real data to create some of our scenarios.

Snyder [9] proposed some scenarios that present different problems and limitations, based on measurements of the conditions of networks installed in hotels, IEEE 802.11 accesses points and other locations. Through these scenarios, he carried out performance tests on various devices that created VPNs (Virtual Private Network) and checked the behavior of each one of them in each proposed scenario. In our work, four scenarios were created based on the scenarios proposed in the Snyder’s study.

Wahab et al. [16] evaluated the impact of encryption/decryption process in Android-based VoIP clients. Authors used three different symmetric encryption algorithms (DES, RC4 and AES) and evaluated three performance parameters (delay, packet loss and throughput). According to them, the application of encryption did not have significant impact on performance parameters.

III. VoIP Quality Assessment

There are several factors that might affect the quality of VoIP calls, such as packet loss, delay, jitter, voice compression and echo cancelation algorithms. In this section, it will be presented details on methods to evaluate call quality.

MOS (Mean Opinion Score) was defined by recommendation P.800 of ITU-T and it is one of the most accepted methods to assess voice quality perception [10]. However, MOS is a subjective method. Under controlled conditions, male and female users assess the quality of pre-defined voice samples. Users score the calls with ratings from 1 to 5. The problem with MOS is that subjective tests may take a long time and may be expensive. Then, researchers developed methods to use MOS ratings in objective tests. Results obtained from an objective assessment method like E-Model, for example, may be converted to MOS ratings.

E-Model is a metric defined by ITU-T in recommendation G.107 for assessing objectively the quality of voice perceived by a user in a telephone call [17]. The E-model takes into account different problems related to telephone calls such as one way delay, coding and decoding process, noise, echo and packet loss. The model output is a quality score named R Factor. The R Factor may be converted to other quality metrics, e. g. MOS. The E-model equation includes five transmission parameters, as can be seen in (1).

\[ \text{（1）} \]

, where:

- \( R_w \): signal to noise ratio (SNR);
- \( I_v \): represents the impairments that are simultaneous to voice signal transmission;
- \( I_e \): represents the impairments that occur after voice signal transmission. It gathers the losses related to delay and echo.
- \( I_{of} \): represent the impairments related to voice coding.
- \( A \): it is named advantage factor. In satellite communications, for instance, users tolerate a lower voice quality due to the technology limitations. In PSTN calls, on the other hand, users do not tolerate quality degradation, because they expect high quality calls. The advantage factor represents these differences in users expectation.

In Table 1, it is presented the relation between R factor and MOS ratings.

<table>
<thead>
<tr>
<th>Factor R</th>
<th>MOS</th>
<th>User satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 – 100</td>
<td>4.34 to 5.0</td>
<td>Very satisfied</td>
</tr>
<tr>
<td>80 – 90</td>
<td>4.03 to 4.34</td>
<td>Satisfied</td>
</tr>
<tr>
<td>70 – 80</td>
<td>3.60 to 4.03</td>
<td>Some dissatisfied users</td>
</tr>
<tr>
<td>60 – 70</td>
<td>3.10 to 3.60</td>
<td>Many dissatisfied users</td>
</tr>
<tr>
<td>50 – 60</td>
<td>2.58 to 3.10</td>
<td>Most users dissatisfied</td>
</tr>
<tr>
<td>0 – 50</td>
<td>1.0 to 2.58</td>
<td>Not recommended</td>
</tr>
</tbody>
</table>

Table 1. MOS, R factor and user satisfaction.
Evaluating Quality of Encrypted VoIP Calls in a Simulation Environment

IV. Methods

In this section, it will be shown the simulation environment created for the realization of VoIP calls, which employs the OpenVPN. Traffic Control and IxChariot tools, the test scenarios that include different network conditions and details on how these tests will be done.

A. Simulation environment

The objective of this work is to evaluate the impact of using cryptography on VoIP calls. To perform the experiments, it was developed the simulation environment as illustrated in Figure 1.

![Figure 1. Simulation environment.](image)

VirtualBox [18] is a virtualization software created by Sun Microsystems. A virtual machine allows an invited operating system to work inside a host operating system, sharing the computer resources. VirtualBox manages resources sharing, enabling the execution of several virtual machines and the communication between them and the host operating system. With VirtualBox, the host operating system can be different of the invited operating systems. For instance, if VirtualBox is deployed in a Windows XP environment, it is possible to create virtual machines with any version of Linux. In this work, we adopt VirtualBox to facilitate the creation of an environment with the call originator and the called party using just one computer.

Open VPN [5] is an open source software that implements virtual private network (VPN) methods in order to create secure point-to-point or site-to-site connections. Open VPN supports several encryption algorithms, since it uses the OpenSSL library. The OpenSSL is an open source implementation of the SSL (Secure Sockets Layer) protocol that support symmetric, asymmetric and hash encryption algorithms. With OpenVPN, each transmitted packet has to be encrypted in the source and decrypted in the destination. Therefore, a delay is inserted in the communication. Depending on the extension of this delay, the communication quality may get worse. In latency sensitive services like VoIP, communication quality degradation represents a significant problem. Encryption also makes packets bigger, demanding more bandwidth. In this work, we evaluate the influence of OpenVPN encryption on VoIP calls quality.

OpenVPN configuration in a VoIP environment is straightforward. One of the call terminals should be the OpenVPN server. The other call terminal should play the role of client, which has the OpenVPN server IP address in its configuration. Then, OpenVPN creates a virtual network interface that will be used to establish secure communication sessions.

In our simulation environment, OpenVPN was put together in the Linux operating system, whereas the Windows version was obtained right from the Internet Open VPN’s official page. We have used the 2.0 rc193 version on the two operating systems. For the tests done in this paper, we used the encryption algorithms AES and Blowfish with a 128-bit key and the Triple DES with one key of 168 bits. AES is defined by the United States National Institute of Standards and Technology (NIST) as an encryption standard. Blowfish and Triple DES are included in several encryption products and security protocols. For instance, the Internet Engineering Task Force (IETF) recommends, in the RFC 4344, the implementation of these encryption algorithms in SSH.

For the three encryption algorithms, we chose the CBC (Cipher-block chaining) mode of operation. This mode is recommended to be used with the AES, Blowfish and Triple DES algorithms, since it is the most appropriate for symmetric encryption algorithms [11]. The CBC is also the operation method recommended by the developers of the OpenVPN to all encryption algorithms supported by the tool.

IxChariot [7] is a commercial tool for assessing the performance of a network and its devices. It helps to identify problems by reporting performance metrics such as used bandwidth, packet loss, amount of incoming packets and so on. IxChariot presents this data in plots, making the data analysis easier. In our experiments, we use the IxChariot VoIP component, which reports delay, packet loss, jitter and the MOS rating converted from R factor of each call. IxChariot is composed by a console, which is responsible for creating and defining how the experiments will be performed, and the endpoints, which are controlled by the console during the experiment execution. The IxChariot console only runs on Windows. The endpoints run on Windows or Linux. In our simulation environment, an endpoint was deployed in the Windows machine and another endpoint was deployed in the Linux machine. The console was deployed in the Windows machine.

The VoIP calls will be made by the IxChariot from the Linux machine to the Windows machine. This configuration was adopted, as the Traffic Control rules are applied to the outbound traffic of the network interface created by the OpenVPN in the Linux machine. The calls to be made will employ the G.711 µ-law codec for voice codification, due to little interference exerted in the delay and in the call quality.

Traffic Control [6] allows the user to control the queues of a network interface. Traffic Control is part of the package iproute2, which includes several command line tools for handling the IP network configuration of a Linux node. There
are different reasons for controlling network traffic:

- Limiting the maximum bandwidth of an interface;
- Limiting the maximum bandwidth of a user or service;
- Enabling the distribution of unused bandwidth;
- Assuring that a given type of packet will be discarded.

For the emulation of traffic problems in the network, it will be employed a module of Traffic Control named NetEm [19]. This module interacts to the Linux core, being able to emulate the packet loss, duplication, corruption, delay and out-of-order packets.

Originally, the network interfaces of the computer used in the tests can operate at speeds up to 1 Gbps. However, for the scenarios proposed in this paper, we had to, significantly, decrease this bandwidth before inserting the VoIP traffic, in order to create specific conditions and evaluate the traffic behavior in this environment. To limit the bandwidth of the network interface, we used the Token Bucket Filter (TBF) [6]. The TBF is a discipline of queue management for the Traffic Control tool. Basically, it is responsible for delaying the packets in a way to emulate smaller bandwidths.

**B. Evaluation scenarios**

It was created different scenarios to evaluate the impact of cryptography on the VoIP calls. These scenarios were created, mainly, based on Markopoulou et al. [8] and Snyder [9] studies.

The scenarios will have different bandwidths and other characteristics such as packet loss, delays, jitters, out-of-order packets and packet duplication. Each one of these parameters will vary according to the scenario we are willing to have. Some of these scenarios will have the congestion emulation, where any given network parameters will be degraded for a short time and then will return to its prior state.

Altogether, seven scenarios will be created. Initially, we created four scenarios which were divided into “S1”, “S2”, “S3”, and “S4”, based on the Snyder’s [9] tests. The Snyder-based scenarios, to simplify the nomenclature, start with the letter “S” followed by a number.

The first scenario, named S1, will have the bandwidth limited in 0.1 Mbits, with 60 milliseconds delay, loss of 2% of the packets, 1% of out-of-order packets, 1% of duplicated packets and 20 milliseconds of jitter. There will be a congestion lasting from 3 seconds at every 20 seconds, which will bring about 30% of packet loss and 1 second delay.

In the scenario S2, the bandwidth will be limited in 0.5 Mbits, with 60 milliseconds delay, loss of 2% of the packets, 1% of out-of-order packets, 1% of duplicated packets and 20 milliseconds of jitter. Congestion will occur at every 20 seconds lasting 2 seconds, which will cause the loss of 3% of the packets and a delay of 1 second.

The scenario S3 will have the bandwidth limited in 0.5 Mbits, with 45 milliseconds of delay, 0.25% packet loss, 1% out-of-order packets, 1% duplicated packets and 10 milliseconds of jitter. This scenario will not have congestions.

In the scenario S4, at last, no kind of problem will be introduced in the network. The bandwidth will be of 100 Mbits with no delay, no packet loss, no failures and no congestions.

Some remarks should be made on the Snyder scenarios, though. In the description of his scenarios, taken up by us as references, it is not mentioned the duration of each congestion, but only the interval they occur. Thus, based on the Markopoulou et al. achievements, we have defined a value of 3 seconds of duration for each congestion. Congestions occur at every 60 to 70 seconds.

Other two scenarios were created based on Markopoulou et al. paper, and their names start with the letter “M” followed by a number.

The scenario M1 is based on the data from the provider P2 in the Markopoulou et al. paper. The bandwidth will be limited in 1.544 Mbits. Markopoulou et al. have not specified the packet loss, however, in their study; they stated that the packet loss rate did not outstrip 0.26% in any of the routes. Therefore, we adopted this value. When there are not congestions, the delay will be of 43 milliseconds and the jitter will be of 12 milliseconds. There will be two congestions and each will last 100 seconds. During the congestions, the delay will be of 78 milliseconds and the jitter will be of 82 milliseconds.

The scenario M2 is based on the data from the provider P4 in the Markopoulou et al. paper. The bandwidth will be limited in 1.544 Mbits. The packet loss will be of 0.25%. When there are not congestions, the delay will be of 40 milliseconds. Congestion will happen at every 65 seconds, lasting 3 seconds each. During the congestions, the delay will be of 275 milliseconds and the jitter will be of 50 milliseconds.

Finally, we suggest a scenario distinct from the others, named B1 scenario. This scenario does not have any problems in the network and has a bandwidth of 100 Mbits. The difference in this scenario, compared to the others, lies in the amount of VoIP calls to be made simultaneously, that is 150 calls. This scenario was built to boost the computer processing demand. With 150 simultaneous calls, the origin and destination of the calls will encrypt and decrypt a significant amount of data. The objective is to check whether, in this situation, some of the algorithms distinguish from the others and what the difference is in quality of the encrypted calls from the non-encrypted ones. Table 2 shows a summary of the proposed scenarios configuration.

**C. Evaluation conduction**

The tests will be carried out in each one of the seven scenarios proposed with the following encryption configurations: without encryption, AES, Triple DES and the BlowFish.

As for the scenarios S1, S2, S3, S4, M1 and M2, each experiment round consists of making a VoIP call. At the end of the call, the value of the MOS obtained is gathered. As for the scenario B1, each experiment round consists of making 150 simultaneous calls. At the end of these 150 calls, the value of the MOS obtained in each call is gathered.
In each scenario, twelve experiment rounds will be carried out for each encryption configuration, resulting in the gathering of twelve MOS values. The best and the worst results are discarded. The ten remaining results will be taken for standard deviation, and a value for the MOS percentage variation coefficient for each scenario combination and encryption configuration.

\[
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i
\]

(2)

\[
\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}
\]

(3)

\[
\text{Variance Coefficient} = \frac{\sigma}{\bar{x}} \times 100
\]

(4)

V. Results

The values for the arithmetic mean (\(\bar{x}\)), the standard deviation (\(\sigma\)) and the percentage variation coefficient (\(c_v\)) collected for all the scenarios are presented in Table 3.

In the scenario S4, we have the network ideal conditions, with bandwidth of 100 Mbps and without any anomaly. The MOS arithmetic mean recorded for the four different encryption algorithms were equals to 4.38. As for the scenario S4, the encryption algorithms have not affected the quality of the calls.

The scenario S3, unlike the scenario S4, includes congestions. In the scenario S3, it is already possible to notice that the network problems have influenced in the quality of VoIP calls. The best result in this scenario was the test without encryption, with a MOS of 4.095. Following that, we have the Triple DES and soon thereafter the BlowFish. Lastly, we have the AES algorithm. For this reason, we can state that no user would realize the difference between the use of AES encryption algorithm, which presented the worst result with a MOS of 4.066, and the call without encryption, which presented the best result with a 4.095 value. Figure 2 shows MOS results for each experiment round in scenario S3. MOS ratings obtained for different rounds are similar, confirming that encryption process did not influence the quality of calls in this scenario.

The scenario S2 brings the occurrence of more severe congestions than the ones found in the scenario S3. We can notice that the tests without encryption have shown the best performance. Among the encryption algorithms we have the Triple DES with the best performance, followed by the BlowFish, and, lastly, the AES. Figure 3 shows MOS results for each experiment round in scenario S2. Despite the fact that MOS results in Figure 3 did not present significant differences, there was a higher variation in \(c_v\) in scenario S2 than in scenario S3. Generally, the quality of the calls was bad for all tested configurations. Again, we do not have a significant performance difference between the various encryption configurations.

The scenario S1 is almost the same as the scenario S2, except for the network bandwidth. Whereas the scenario S2 has a bandwidth of 500 Kbps, the scenario S1 has a bandwidth of only 100 Kbps. In the scenario S1, we have observed that the configuration without encryption ranked in the third position, with the Triple DES in first and the BlowFish in second. The AES, as it occurred in the scenario S3 and in the scenario S2, remained as the smallest MOS encryption algorithm, in the last position. In this scenario we had a variation of MOS results of about 3% for each encryption algorithm, above the scenarios S3 and S2, showing that this scenario is more instable than the ones we presented earlier. As for the variation, we can highlight the Triple DES, which presented a bigger variability in their results, and the AES, the encryption algorithm that had less variation in their MOS values. Like the scenario S2, the quality of the calls was bad for all the encryption configurations tested. Figure 4 presents MOS results for each experiment round of scenario S1.

After finalizing the tests, it is possible to notice that the scenarios S1 and S2 had close MOS final results, varying from 2.2 to 2.4, and the scenario S3 displayed higher values and much better levels of user satisfaction. In these three scenarios we saw that the AES encryption algorithm presented the lowest performance among the algorithms. The anomalies inserted in the network have strongly impacted on the quality of the call, whereas the use of cryptography has not brought differences of performance perceived by the end user.

In the scenario M1, unlikely it happened in the prior presented scenarios, the best performance was achieved by the AES, with a MOS of 3.807. Following this, with very close results, we have the Triple DES and the configuration without encryption with same MOS value: 3.804. Lastly, we have the BlowFish, with a MOS of 3.785.
The variation of results has got around 1%, significantly smaller than the ones in the scenarios S1, S2 and S3. Markopoulou et al. have rated good for the quality of the VoIP communication of this scenario, pointing that we have used the research data from Provider P2. In our tests, the calls were also rated as having good quality.

In the scenario M2, the best performance was obtained by the AES, with a MOS of 4.002, followed by Triple DES, the configuration without encryption (3.998) and the Blowfish (3.994). As for the variation, we can state that the variation coefficient of MOS results for each encryption algorithm has not reached 1%. Despite the congestions inserted in this scenario, we can notice that the quality of the calls, at average,
was good.

Figure 5 and Figure 6 present MOS results for each experiment round of scenarios M1 and M2, respectively. In both scenarios, MOS ratings for different encryption algorithms and different experiment rounds are very similar.

Finally, we have the results for the scenario B1. In this scenario, the goal is to make 150 simultaneous calls without any anomaly in the network to check the behavior of the encryption algorithms when there is a higher workload on the calls origin and destination processors. The values of 150 calls were defined in accordance to the set of hardware used in our tests. We have noticed in experimental tests that, when getting the amount of 150 simultaneous calls, the MOS results began to change in a remarkable way for all the encryption algorithms. It is important to note that this value, naturally, varies according to the computing power of the hardware in use.

The difference between the results of configuration without encryption and the configurations with encryption algorithms is greater than that in the preceding scenarios. Due to the delay generated by the encryption/decryption process, the MOS results of this scenario were ranked behind the scenario where no cryptography was employed. Yet, all the arithmetic mean values at the end of the tests lie between MOS 3.6 and 4.0. In theory, the user would not still be able to distinguish the difference of quality of a call encrypted with Triple DES (the worst result) from another without encryption (the best result). Figure 7 shows the MOS results for each experiment round of
Table 4 brings the arithmetic mean for each encryption algorithm in each scenario tested, and, in parenthesis, the performance rank of the encryption algorithm in its respective scenario. In the far right column, we have the arithmetic mean ($\bar{x}$) of the algorithm MOS throughout the scenarios tested. With this value, we can classify the algorithms according to their MOS quality presented in the encrypted calls by themselves. We have not included the results of the scenario S4, as they are of little relevance to demonstrate the differences among the algorithms, once all of them had the same final result of MOS.

Through this arithmetic mean, it is possible to remark that the Triple DES displayed the best performance generally speaking, whereas the AES showed the worst one. However, it not possible to say, from these results, that any given algorithm has effectively a superior performance, as the results had little variation from one to another. The variance of quality in VoIP calls when using the algorithm Triple DES (3.311) and the other using the AES (3.283) is imperceptible to the user.

With the results obtained in our study, it is possible to make a point on how each scenario of the network we tested has influenced the results in the quality of a VoIP call. We can, this way, check which scenario brought about greater variation in the MOS results. Initially, it is necessary to estimate the arithmetic mean ($\bar{x}_{\text{scenario}}$) by applying the MOS arithmetic means of every encryption algorithm at the
end of each scenario, such calculation illustrated by (5). Further that, it is calculated the standard deviation for, afterwards, getting the percentage variation coefficient.

Table 5 shows these values and the user satisfaction levels according to the results obtained. The scenario M2, if not taking in consideration the scenario S4, was the steadiest of all, for the encryption algorithms got always very close and little variation MOS values. Then we have the scenarios S3 and M1 with a variation of 0.27% and 0.30% respectively. Finally, we have the scenarios S2 and S1 as the ones which presented the highest instability in the MOS values of the algorithms, showing values of 1.36% and 1.3% respectively. These results suggest that the biggest variances of MOS values gathered are related to the anomalies inserted in the network and are not related to the utilization of different encryption algorithms.

VI. Conclusion

The main goal of this paper is to check the impact of different encryption algorithms on VoIP calls established in networks under different conditions. Several conditions of operation presented in each scenario were created by using resources from Traffic Control software. The encryption was carried out by the OpenVPN. VoIP calls were created and evaluated by the IxChariot program. The MOS was used to indicate the quality of calls being made.

In scenarios with just a single call at a time, where various network problems were introduced, e.g., delay, jitter, and packet loss, there were not remarkable differences between an encrypted VoIP call and a non-encrypted one. The main problem in these cases was caused by the traffic anomalies inserted in the network. In these conditions, we should take advantage of other criteria to choose the encryption algorithm, for instance, its vulnerability background, compatibility with other resources or key sizes.

In the scenario where 150 simultaneous calls were emulated, the difference between quality of the calls for different encryption algorithms and for the configuration without encryption was bigger than the difference found in the tests carried out with a single call. However, the differences are not big enough to be perceived by the end users.

Future research should be conducted to evaluate the relationship between the different network problems and the MOS. Additional research may be also carried out to create more evaluation scenarios to distinguish the impact in the MOS caused by network problems and encryption algorithms. It would be also useful to repeat the analysis using public-key encryption algorithms, due to its performance issues.

References


Author Biographies

Dherik Barison received his BS degree in computer science from the State University of Londrina, Brazil, and a MSc degree in electrical engineering from the University of Campinas, Brazil. His research interests include information security and VoIP (Voice over IP).

Rodrigo Sanches Miani received his BS degree in mathematics from the Federal University of São Carlos, Brazil, and the MSc degree in electrical engineering from the University of Campinas, Brazil. He is currently a PhD candidate at the Laboratory of Communication Networks, School of Electrical and Computer Engineering, University of Campinas. His research interests include information security, security quantification models, network security, Open Access MAN and communication network management.

Gean Davis Breda received his BS degree in electrical engineering from the Federal University of Santa Maria, Brazil, in 1993. He received his MSc degree in electrical engineering from the School of Electrical and Computer Engineering of the University of Campinas, Brazil in 1996, and in 2008, at the same university, received his PhD in electrical engineering. During 1996–2002, he worked as a system engineer in CDMA cellular systems, where he was responsible for system support and operation. Also, since 2009 he has been working with Metropolitan Area Networks. His research interests include QoS in telecommunication systems, and billing.

Leonardo de Souza Mendes received his BS degree in 1985 from the Gama Filho University, Rio de Janeiro, his MSc degree in 1987 from the Catholic University of Rio de Janeiro, and his PhD degree in 1991 from Syracuse University, all in electrical engineering. In 1992 he joined the School of Electrical Engineering of the University of Campinas, Brazil. Professor Mendes’s recent R&D focus is in the studies and development of communications engineering applications for metropolitan IP networks. Professor Mendes created, at UNICAMP, the Laboratory of Communications Network (LaRCom), of which he is now the Director and also the main coordinator.

Bruno Bogaz Zarpelão received his BS degree in computer science from the State University of Londrina, Brazil, and a PhD degree in electrical engineering from the University of Campinas, Brazil. He is currently a professor at the Computer Science Department, State University of Londrina, Brazil. His research interests include smart cities, Open Access MAN, communication network management and information security.