Comparing Network Performance of Mobile VoIP Solutions

Rafael Dantas
Lero
University of Limerick
Limerick, Ireland
Email: rafael.dantas@lero.ie

Dr. Chris Exton
Lero
University of Limerick
Limerick, Ireland
Email: chris.exton@ul.ie

Dr. Andrew Le Gear
Horizon Globex
Nexus Innovation Centre
Limerick, Ireland
Email: andrew.legear@horizon-globex.ie

***Abstract*—Maintaining consistent VoIP quality is a challenging task, especially where it is carried using a mobile internet connection. With greater than 50% of the world's mobile user population using older GSM incarnations, this is very much a present technical research challenge. This paper presents an approach to improving mobile VoIP telephony by drastically lowering the bandwidth consumption while maintaining acceptable call quality when compared to competitor solutions. We verify this claim through a quantitative analysis of bandwidth consumption and MOS test of audio quality of 10 different VoIP solutions.**

***Index Terms*—VoIP, Mobile, Bandwidth, Consumption, Network, MOS**

I. Introduction

As the 5g era approaches and the number of connected smart devices continue to increase, we find existing network infrastructures under increasing strain [1]. This change disproportionately affects developing nations, as old hardware will be required to support the increasing demand of growing populations [1].

While old hardware can be upgraded and new equipment added, it can quickly become an expensive task, especially considering that most will expect access to the Internet with 3g or faster connections.

With the majority of the world's mobile user population continuing to use 2G networks [1], network traffic optimisations can still provide benefits for mobile VoIP applications where modern mobile internet infrastructure is not available.

In order to explore this market niche, Horizon Globex[1] created the *Smart Packet VoIP Solution (SPVS)* [2], a complete solution for ultra-low bandwidth mobile *Voice Over Internet Protocol (VoIP)* communications.

This solution includes an Android and iOS mobile application, a series of node servers responsible for broker signalling between the phones' mobile applications, VoIP servers for hosting calls between mobile applications regionally, and gateways to standard telephony protocols [2], if necessary. It also employs a series of network and codec optimisations in order to allow real-time communication that is also stable, reliable and bandwidth efficient.

This paper will provide a base comparison of bandwidth usage and perceived quality using Mean Opinion Scoring (MOS) [3] between the SPVS and its competing solutions.

II. Related Work

Performance and quality comparisons between applications are commonplace in mobile VoIP literature [4], [5], [6], [7], [8], [9], [10].

Such comparisons can be used to test the performance of a VoIP codec or VoIP network solution against its competitors or longstanding market standards in order to build a ranking of the best applications. For example, [4] used their experiment to select a new codec for the Inmarsat mini-M system. Each speech sample was processed through a number of different conditions. Then a group of subjects evaluated the quality of all samples using a 5-scale MOS test. In their experiments, the Advanced Multi-Band Excited (AMBE) codec was the best fit for their requirements. [7] tested the performance of a number of codecs, including iLBC, Skype and Speex. They used Perceptual Evaluation of Speech Quality (PESQ) to evaluate how adverse network conditions like jitter and packet loss affected the quality of the speech sample. As a result, they could infer how robust the codecs were against bad network conditions and concurrent simultaneous flows. To prove that speech codecs are not capable of providing good quality experience, [10] tested various codecs with varying sampling frequencies under many listening conditions. They used a 9-scale MOS variation to assess quality of the speech samples. They concluded that listeners prefer wider bandwidths over narrower bandwidths and stereo over mono. To evaluate the quality of the 3g network in Bangkok (Thailand), [11] tested quality variations in two mobile VoIP applications: Skype and Line. They used PESQ to measure the quality of 5 different 3g

[1]Horizon Globex is an Ireland-headquartered public company which provides bandwidth-efficient mobile VoIP solutions as both white-labelled software for mobile carriers and end-user software.

[2]e.g.: SIP, SS7

2573-7562/18/$31.00 ©2018 IEEE
DOI 10.1109/MobileCloud.2018.00015

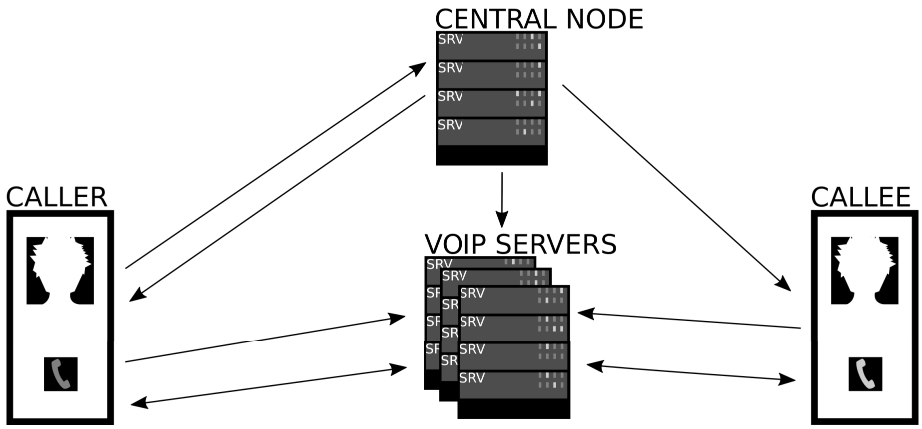

Fig. 1. Typical workflow of a SPVS call.

mobile networks within 14 universities in the city. The results were used to decide which operator was the best one, and they propose their work may be used to assess the quality of the network over an area in the future. In order to select the codec for their new ARM-based SIP phone, [5] also performed comparisons between PCM, Speex and iLBC. They used R value [12] and a 5-scale MOS to measure the performance of the aforementioned codecs. Their results show iLBC being superior to the other two under certain circumstances.

[13] criticises the MOS scale and propose alternative ideas for media quality measurement. They also analyse the suitability and limitations of MOS for different applications. Although they conclude that managing the Quality of Experience (QoE) is more important than the MOS, they also recognise the suitability of this tool for quality monitoring, metric tuning and validation.

Network performance comparisons can be also used to measure the differences between each application. To improve the usage on TCP connections via 3g/4g networks, [8] implemented a group of modifications to the TCP protocol on some routers, transparent to both ends of the link. By measuring the average usage of the network they proposed their optimisations could raise the throughput by 48 to 163%. [9] tested various algorithms in order to reduce a temporary deadlock on the TCP protocol, caused by Nagle's algorithm [14]. They describe five possible modifications to reduce the time spent in a deadlocked state. They evaluate that each modification solves the problem for a different set of cases, and they recommend that the implementer not disable the algorithm due to the protection it provides for the network. To reduce the impact on the perceived quality caused by packet loss in a VoIP system, [6] tested a number of codecs including AMR. They only measured the amount of packets dropped for each codec. Their results show that AMR has a lower chance of losing packets when compared with the other ones.

III. The Smart Packet VoIP Solution

The SPVS is a complete mobile VoIP network solution including PSTN breakout capabilities[3]. It can be up to 90% more bandwidth efficient than competing applications by employing a series of optimisations, described in section III-A.

Figure 1 shows the basic workings of a typical VoIP call using SPVS:

1) First the application contacts a central node, which will try to find the other user inside the network.
2) The central node will then assign a VoIP server for handling the call at the optimal geographic location, sending this information to both participants.
3) Both ends will try to connect with the VoIP server. This is done to avoid any NAT or proxy problems that would arise otherwise.
4) Once connected, the VoIP server becomes responsible for relaying the call between the participants.
5) If one or both the participants are not able to connect to the VoIP server, a SIP server will call the unreachable user using the normal telephony infrastructure.

A. Optimisations

Key to the efficiency gains of the SPVS over its competitors is an accumulation of important protocol optimisations.

1) Protocol header elimination: Most protocols carry some metadata alongside the payload. This header information is used by the receiver to understand and access the information contained inside, especially in cases where the configuration parameters can be changed for each individual packet.

In order to improve bandwidth efficiency, the SPVS protocol doesn't carry metadata on every single packet. Instead we use a series of metadata packets in order to coordinate the call between the devices, allowing the packet to carry only the actual encoded speech data.

[3]While SPVS allows two users to communicate via traditional telephony infrastructure, it also supports one or both sides to be completely disconnected from regular PSTN as long as that user has access to the internet.

[4]Lowest estimate provided by Google Play Store as of 01/03/2017.

Application	Installations[4]
WeChat	2,344,124,542
QQ	2,004,169,512
Messenger	1,000,000,000
Hangouts	1,000,000,000
WhatsApp	1,000,000,000
Line	500,000,000
Skype	500,000,000
Viber	500,000,000
Telegram	100,000,000

TABLE I
INSTALLED DEVICES ESTIMATE.

2) Codec header elimination: In a similar way to protocols, codecs also carry a series of parameters that must be sent alongside the encoded data, which will be used during the decoding process. As an example, SPEEX parameters can be quite complex; as detailed by [15].

If both sides of the communication agree the configuration parameters of the payload then the header section that carries this information becomes redundant. It is then possible to reduce header size, and ultimately remove it altogether, by standardising every parameter of the transmission.

SPVS uses only one codec with a very specific set of parameters, making payload headers completely obsolete and allowing every packet to carry only the compressed speech signal.

3) Header-to-payload ratio reduction: Even if the payload does not carry a single bit of header information, every single network protocol will envelop all data produced by the layer above and append its own headers to it. Since those protocols are usually outside the scope of the application and only accessible by the OS layer or lower, the changes proposed in III-A1 cannot be applied here.

To reduce the overhead imposed by the network protocol stack we have used a frame size of 100ms instead of 20ms. This change allows for a better usage of the network infrastructure, similar to the optimisations provided by Nagle's algorithm [14] on TCP.

Using 100ms frames results in the economy of 4 frames when comparing with 20ms frames. The data economy can be represented by the following relationship:

$$S = (H_{ip} + H_{udp}) * 4 \tag{1}$$

where H_{ip} represents the size of the IP header, H_{udp} is the size of the UDP header and S is the amount of data saved by each 100ms frame sent.

According to [16], possible values for H_{ip} are $20 \leq H_{ip} \leq 60$ bytes. The header in the UDP has a fixed size [17], thus $H_{udp} = 8$ bytes. Using these values, it is possible to calculate the amount of data saved, $112 \leq S \leq 272$ bytes per frame.

Laptop 1	Lenovo Thinkpad T420
Laptop 2	Dell Latitude E7270
Operating system	Debian GNU/Linux (stretch)
Kernel version	4.6.0-1-amd64
Cell phone 1	Motorolla Moto G3 (Android 6.0)
Cell phone 2	Sony-Ericsson Xperia (Android 6.0)

TABLE II
EQUIPMENT USED ON THE EXPERIMENTS

4) Silence detection: In a normal conversation, usually only one of the participants is speaking while the other is listening to what is being said. If an application is not aware of this special case, it will always record and send speech information both ways.

Using silence detection algorithms, it is possible to avoid sending useless data through the wire[5].

The SPVS has a special silence packet, which is sent by the application whenever it detects its user has gone silent. This packet also doubles as a keep-alive message, being sent periodically by the silent side in order to maintain the open connection.

5) Codec selection: As a freely available open source voice codec, SPEEX was selected to be integrated into the SPVS solution. It also provides a good result when compared with other narrowband voice codecs like AMR and iLBC [10] [15] [7].

6) Compression level: SPEEX has many compression levels that decrease speech fidelity in order to decrease the overall size of the stream. This value ranges between 0 and 10, the former being the one with the highest compression but lowest quality while the latter is the opposite. We decided to use compression level 3. We have found that any higher setting does not provide an improvement of the perceived speech quality through the phone speaker.

7) Transport protocol selection: Although TCP [19] was built to be reliable and stable, its retransmission routines are usually more harmful to real-time applications like VoIP than the lost data they were designed to recover.

As UDP [17] only checks for consistency errors inside the packet and silently discard the corrupt ones, it was selected as the most suitable to the application. Its headers also contain only 8 bytes, reducing even more the overhead on every packet sent.

In order to reduce the chances of datagram fragmentation and avoid delays caused by reassembly, every SPVS datagram has a maximum size of 160 bytes.

IV. EVALUATION

Section IV-A will analyse the network performance of the SPVS, described in section III, against the competing solutions in order to test the assertions made in section I.

To achieve this goal, we designed two experiments. The first measures the network performance of the solutions by comparing the data transmission rates and overall data usage

Since WeChat and QQ's mainly operate on the Chinese market, their numbers were taken from Huawei's application store.

Hangouts' numbers might not reflect its actual popularity, since Android devices usually come with it installed.

[5]The Speex library, for example, ships with built in algorithms for silence detection [18].

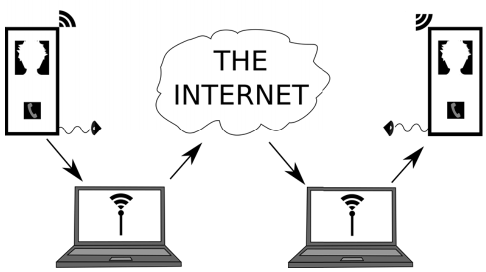

Fig. 2. Representation of the experiment.

for the same message, detailed in section IV-A. The second is a MOS adherent listening test to access the perceived speech quality of the solutions, described in section IV-B.

For these experiments, we have selected some of the most commonly used applications in the western market, namely "Facebook Messenger", Hangouts, Line, Skype, Telegram, Viber, WhatsApp. We have also selected two popular Chinese applications in order to represent that market, namely QQ and WeChat. Table I refers to the number of installed devices for all aforementioned applications.

A. Network performance test

This experiment was designed to provide quantitative data for the analysis on section IV-D. We used two Android mobile phones (see table II) where the tested application would be installed with two GNU/Linux laptops acting as access points for both mobile phones and an earphone to reproduce a half-minute-long audio file through both devices' microphones. Figure 2 is an illustration of this experiment.

In order to properly intercept all data going through the Internet, both phones were connected to it via the wireless access points provided by both laptops, which were connected to the Internet via two independent Ethernet cables.

Using a personalised command-line interface described on sub-section IV-C to schedule 40 test executions for each application described in section I. This CLI was responsible for synchronising the actions of both computers during each test.

All tests performed followed these steps:

- All tests are queued on the console, grouped by application.
- Using a "next" command, the terminal loads the next test and waits for a user input. This input signals that both phones are connected in a call and ready for the audio playback.
- Once the "enter" key is pressed, "tcpdump" is invoked as a background process, in order to record all network activity necessary for data collection.
- Immediately after the last step, the tool plays a half-minute-long audio file. This file contains phrases that alternate between the left and right channels to simulate a conversation with moments of activity and silence.
- Both sides wait until the audio reproduction is complete. Then the terminal signals "tcpdump" to stop recording the network.
- A counter is incremented to signals the completion of this test and the interface waits for a new "next" command or another user interaction.

All data collected by tcpdump was filtered using Wireshark, a network packet capture tool, which can conveniently read the format used by tcpdump and output the desired fields into the console, for further analysis.

Since all data going through the laptop was recorded into files, the most accessed IP address for that call was used to filter the data.

B. Mean opinion score (MOS) test

The second experiment is a MOS test, divided into two parts; a recording session and the listening test. The first half aims to record speech samples as they are received by the mobile device after being transmitted through all the solutions listed in IV. The second half is a listening test using the MOS scale of the aforementioned samples.

As recommended by [3], all original samples contain two short phrases separated by a brief moment of silence by the speaker. Both phrases also did not have obvious relationship between each other. All samples are shorter than three seconds.

As recommended by [20], all samples were normalised to -26 LKFS [6] with maximum true peak under -1 dBTP [7].

The recording session used two mobile phones to simulate a conversation. The first phone was placed inside a sound room and had two speakers were set at 10cm distance of the first phone's microphone and sound levels were calibrated to be under 80 dB(A) [8]. The ambient noise level of the sound room was 33.9 dB(A). This phone was used to call the second phone using one of the applications mentioned in I. The second phone was directly connected to a laptop using an auxiliary audio cable and was used to record the calls.

The recording procedure went as follows:

- Device 1 called Device 2. Device 2 answered the call.
- Speech sample was reproduced for Device 1.
- Speech received by Device 2 was recorded by the computer.
- All steps were repeated for all applications, using the same male and female speech samples.

After the recording, all resulting samples were renormalised to -26 LKFS with maximum true peak under -1 dBTP and loaded into the survey application.

The Mean Opinion Score (MOS) [3] test was then performed with the assistance of a mobile application, followed these steps:

[6] The LKFS value represent the loudness of the sample, where each increase of 1 dB will produce a 1 LKFS unit increase on loudness readings [21].

[7] "True-peak level is the maximum (positive or negative) value of the signal waveform in the continuous time domain; this value may be higher than the largest sample value in the 48 kHz time-sampled domain." [21]

[8] The dB(A) value represent the sound level, weighted according to the response of the human ear [22].

Application	Wire throughput (kb/s)	Payload throughput (kb/s)	Packet rate (p/s)	Frame size (bytes)	Payload size (bytes)	Payload-to-header ratio
Horizon (SPVS)	8.24	5.90	7.97	130	93	2.52
WhatsApp	23.59	17.44	22.23	133	99	2.83
QQ	28.64	20.07	31.08	116	81	2.34
Facebook	29.23	24.02	16.76	219	180	4.61
Viber	40.66	28.22	45.02	113	79	2.27
WeChat	44.85	35.83	33.13	170	136	3.98
Telegram	48.22	40.40	27.71	218	183	5.17
Line	53.82	42.21	42.33	159	125	3.64
Hangouts	59.06	32.35	50.80	146	80	1.21
Skype	84.54	61.69	83.98	126	92	2.70

TABLE III
COMPARISON BETWEEN VARIOUS APPLICATIONS PERFORMANCES

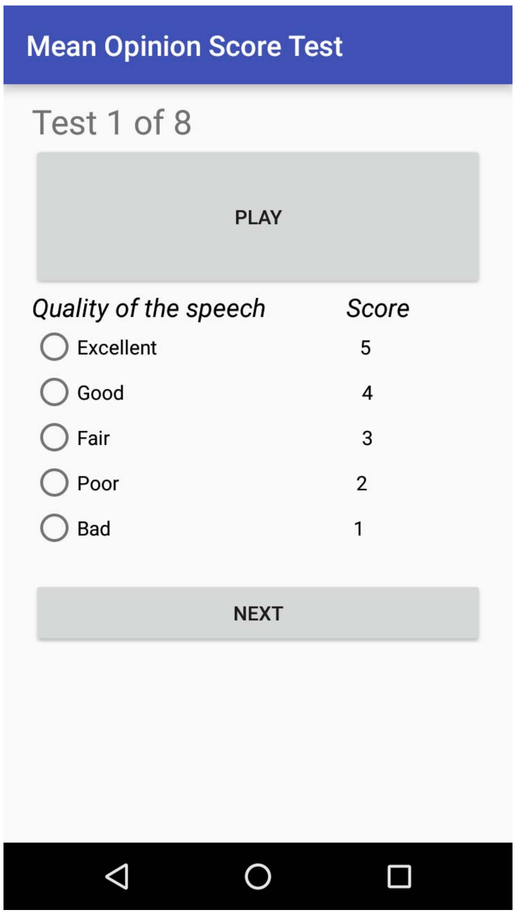

Fig. 3. Screen capture of the MOS test application.

- The subjects received a mobile device with a MOS test application running.
- They were presented to the starting screen, which contains the basic instructions for the test.
- Before the real test, users were presented with a trial run with 6 samples: Female and male speech samples using 16 kHz PCM, 12 kHz PCM, 8 kHz PCM. The score for this trial test was not used in the results.
- By the end of the trial run, the subject could ask questions about the procedure. Technical questions could only be answered after the end of the experiment.
- After all questions were satisfied, the subject would press the “start” button to the first sample of the real test. All speech samples are loaded in a random order.
- In this screen, the user had to first listen to the sample in its entirety, by pressing a “play” button.
- After the playback, the subject graded the quality of the audio sample. This scale was represented by the following options:
  - 5 - Excellent
  - 4 - Good
  - 3 - Fair
  - 2 - Poor
  - 1 - Bad
- Previous steps were repeated until all samples have been evaluated.

After all subjects evaluated all samples, a simple average was applied to all applications scores in order to produce the final results.

C. Tool Support

A personalised command-line interface was created to coordinate both laptops in all the tasks they were performing during the first experiment. This interface is able to connect the computers, interface with local network configuration by invoking “tcpdump” [23] and “tc” [24], and it’s also capable of rollback a test if needed.

Two bash scripts were written for filtering of data after its collection. These filters use “tshark” [25] to filter the pcap files created by “tcpdump”.[9].

The audio sample was reproduced using “mplayer” [26], a command-line interface player which was called by the script before changing the network properties.

All data was collected using “tcpdump” [23], a tool for inspecting all traffic on a network and storing all data into disk. The output format is widely supported by many tools.

For data analysis, “tshark” [25] and “Wireshark” [27] were selected for filtering the results, removing unrelated network traffic and for selecting only the fields necessary for the analysis.

“Tc” [24] is a tool created to configure traffic control inside the Linux kernel. It is used for defining rules for incoming and outgoing packets on a device. This tool has many extensions,

[9]These scripts, along with set up advice, are available from the from the authors on request.

Application	Female Speech	Male Speech	Overall
Horizon (SPVS)	2.72 (0.89)	2.39 (0.97)	2.56 (0.94)
Facebook	2.67 (1.14)	2.61 (1.04)	2.64 (1.07)
QQ	3.06 (1.16)	3.11 (1.08)	3.08 (1.11)
WhatsApp	3.89 (0.76)	3.33 (0.91)	3.61 (0.87)
Telegram	4.17 (0.79)	3.78 (1.00)	3.97 (0.91)
WeChat	3.89 (0.96)	4.06 (0.80)	3.97 (0.88)
Line	4.39 (0.85)	4.56 (0.62)	4.47 (0.74)
Hangouts	4.78 (0.43)	4.33 (0.84)	4.56 (0.70)
Viber	4.50 (0.62)	4.67 (0.49)	4.58 (0.55)
Skype	4.83 (0.38)	4.44 (0.71)	4.64 (0.59)

TABLE IV
RESULTS OF THE MOS EXPERIMENT USING HEADPHONES

Application	Female Speech	Male Speech	Overall
Horizon (SPVS)	2.83 (1.17)	1.83 (0.75)	2.33 (1.07)
Facebook	3.33 (0.52)	2.50 (0.84)	2.92 (0.79)
QQ	3.33 (0.52)	2.67 (1.37)	3.00 (1.04)
WhatsApp	4.00 (0.63)	3.67 (1.03)	3.83 (0.84)
Telegram	4.67 (0.52)	4.00 (1.10)	4.33 (0.89)
WeChat	4.00 (0.63)	4.33 (0.82)	4.17 (0.72)
Line	4.33 (0.82)	4.33 (0.82)	4.33 (0.79)
Hangouts	4.83 (0.41)	4.50 (0.84)	4.67 (0.65)
Viber	4.50 (0.55)	4.33 (0.82)	4.42 (0.67)
Skype	5.00 (0.00)	4.33 (0.82)	4.67 (0.65)

TABLE V
RESULTS OF THE MOS EXPERIMENT USING THE PHONE'S EARPIECE

so called *queuing disciplines* or “qdiscs”, which are queues used to save packets while the network interface isn't ready to handle them.

In this experiment, every call happened after “tc” was used to create a new class. This class was configured to limit the bit rate to 100kbps, ensuring the application would use a narrowband codec and giving the previously mentioned terminal some band to run commands without interfering with the experiment.

For the second experiment, a simple mobile application was developed to reproduce the sounds and collect the scores given by the users.

D. Quantitative Analysis Results: Bandwidth Usage

Table III has a comparison between all applications analysed as part of our evaluation.

- “Wire throughput” is an average data rate going through the network, including the headers for the physical, datalink, network and transport layers.
- “Payload throughput” is also an average data rate, including only the data carried by the transport layer, without its headers.
- “Packet rate” is the average number of packets sent per second.
- “Frame size” is an average of each individual packet size, including the entire network overhead while payload size removes the same overhead sizes.
- “Payload-to-header ratios” are simple ratios between the payload size and the average network overhead size, which is the frame size minus the payload size.

When compared against the second closest competing solution in the first three categories, the SPVS uses, in average, 65%, 66% and 52% respectively less resources.

The results are striking - using less than half bandwidth than the second closest solution, the SPVS has a significant network performance lead in this experiment.

E. Qualitative Analysis Results: Mean Opinion Score

Table IV and table V have comparisons of the opinion scores given to all applications tested in this experiment. Higher scores indicate better quality perceived by the users. The numbers inside brackets are standard deviations of all scores, smaller numbers represent higher agreement between participants. The first table is organised in ascending order of overall scores while the second table keeps the same order to facilitate comparison.

The data contained in table IV was generated by averaging answers given by 18 participants using headphones. Table V uses the answers of 6 participants using the phone's earpiece.

Female and male speech samples' scores are listed individually because different solutions may act in different ways between different voice ranges. Overall scores are the result of a simple average between all scores given to both male and female speech samples of the same application.

V. DISCUSSION

Application	Overall MOS	Wire throughput (kb/s)	Cost (cents/s)
Horizon (SPVS)	2.56	8.24	0.08
WhatsApp	3.61	23.59	0.23
QQ	3.08	28.64	0.28
Facebook	2.64	29.23	0.29
Viber	4.58	40.66	0.40
WeChat	3.97	44.85	0.44
Telegram	3.97	48.22	0.47
Line	4.47	53.82	0.53
Hangouts	4.56	59.06	0.58
Skype	4.64	84.54	0.83

TABLE VI
AVERAGE APPROXIMATE COST OF A CALL (ASSUMING 10 CENTS/MB COST)

The SPVS shows the best results with respect to the amount of data sent through the network, using at worst half as much data as the second closest competitor but on average 80% less throughput than other competitors.

Table VI shows the average cost per second of a call using each solution tested. In order to calculate the correct cost, we have assumed an approximate 10 USD cents/MB [10], since a precise figure would vary between contract agreements between the solution's developer and it's internet service provider.

Skype had the best payload-to-header ratio, but that was because the payload carried by each packet was also larger in comparison to the size of the header.

It is also surprising that most of the tested solutions were using TCP as transport protocol instead of UDP, which would be the more sensible choice for real time applications like mobile VoIP, given the reasons discussed in section III-A7.

[10]This is standard rate for inter-telecom company GSM data

As expected, there is a clear correlation between the solutions that used more bandwidth to transmit the speech samples and the final scores given by the participants, as seen in table IV.

Table V suggests that the participants might give better scores when listening the same samples when using the phone's earpiece instead of headphones.

Although male speech samples show higher deviations in table V when compared to the values in table IV, possibly due to its lower tones, both the female speech samples and the overall scores had lower deviations. This implies a higher degree of agreement between the participants when using the device's earpieces instead of headphones.

Another important aspect of mobile VoIP solutions is their ability to tolerate faults caused by a number of network problems. Many applications bet on sending smaller amounts of data more frequently in order to reduce the relevance of the packets lost, the SPVS tries to send larger amounts of data less frequently. While this behaviour may cause lost packet to be more perceptible to the listener, the developers hope the larger buffer and less intense packet rate to be ideal for bad network conditions and further studies are required to prove this hypothesis.

VI. Ecological Validity

To improve the ecological validity of this study and avoid any interruptions or sudden loud noises that might distract the participant of the task, it was conducted in a sound room.

Both male and female speech samples were used in order to account for possible variations that could be introduced by the codecs tested due to differences in tone and pitch.

The subjects were first introduced to a set of samples in a preliminary test, designed to familiarise them with the score collection application and the task of grading the samples.

All samples were presented in random order for each listener, with no preference for any application or speaker gender, in order to reduce priming effects that any particular sample might cause to subsequent scores.

VII. Threats to Validity

Network fluctuations could affect the results of this experiment, making one solution appear better or worse than it would have been otherwise. Traffic shaping rules could also affect the numbers in a similar fashion.

VIII. Future Work

We intend to analyse ways to improve the current protocol. Ideas for future refinements include:

- A detailed study about the effects of some bad network conditions in the scores given to all solutions.
- A dictionary that will store common patterns produced by the codec to further compress the information being transmitted.
- A permanent personal audio profile to improve codec compression when talking with the same person more than once.

IX. Conclusion

We conclude that the SPVS shows very promising results in regards to bandwidth consumption while still maintaining acceptable perceived quality when compared to other competing solutions in the mobile VoIP market.

X. Acknowledgements

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie). (SOW2016-034)

This work was possible thanks to Brian Collins, CEO of Horizon Globex, for permitting technical analysis of the product.

References

[1] C. Visual Networking Index, "Global mobile data traffic forecast update, 2016–2021 white paper," *link: http://goo.gl/ylTuVx*, 2017.

[2] B. Collins and C. Dziedzic, "Method and devices for routing in a satellite-based communication system," Jun. 21 2016, uS Patent 9,374,152.

[3] ITU-T, "Recommendation p.800, methods for subjective determination of transmission quality," *International Telecommunication Union*, 1996.

[4] S. Dimolitsas, F. Corcoran, C. Ravishankar, A. Wong, S. de Campos Neto, and R. Skaland, "Evaluation of voice codec performance for the inmarsat mini-m system," 1995.

[5] J. Zhou, T. Wu, and J. Leng, "Research on voice codec algorithms of sip phone based on embedded system," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 183–187.

[6] J. W. Seo, S. J. Woo, and K. S. Bae, "Study on the application of an amr speech codec to voip," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 1373–1376.

[7] K. Kim and Y.-J. Choi, "Performance comparison of various voip codecs in wireless environments," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*. ACM, 2011, p. 89.

[8] K. Liu and J. Y. Lee, "On improving tcp performance over mobile data networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2522–2536, 2016.

[9] J. C. Mogul and G. Minshall, "Rethinking the tcp nagle algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 1, pp. 6–20, 2001.

[10] A. Rämö, "Voice quality evaluation of various codecs," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4662–4665.

[11] P. Wuttidittachotti and T. Daengsi, "Quality evaluation of mobile networks using voip applications: a case study with skype and line based-on stationary tests in bangkok," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 7, no. 12, p. 28, 2015.

[12] J. A. Bergstra and C. Middelburg, "Itu-t recommendation g. 107: The e-model, a computational model for use in transmission planning," 2003.

[13] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[14] J. Nagle, "Congestion control in ip/tcp internetworks," 1984.

[15] E. Touloupis, A. Meliones, and S. Apostolacos, "Implementation and evaluation of a voice codec for zigbee," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*. IEEE, 2011, pp. 341–347.

[16] J. Postel *et al.*, "Rfc 791: Internet protocol," 1981.

[17] J. Postel, "Rfc 768: User datagram protocol," Tech. Rep., 1980.

[18] J.-M. Valin, "Speex: A free codec for free speech," *arXiv preprint arXiv:1602.08668*, 2016.

[19] J. Postel, "Rfc 793: Transmission control protocol," 1981.

[20] E. T. Committee *et al.*, "Loudness recommendation ebu r128," 2011.

[21] ITU-R, "Recommendation bs.1770-4, algorithms to measure audio programme loudness and true-peak audio level," 2011.
[22] ANSI, "Ansi s1.4: Specifications for sound level meters," 1983.
[23] V. Jacobson, C. Leres, and S. McCanne, "Tcpdump public repository," *Web page at http://www. tcpdump. org*, 2003.
[24] B. Hubert and A. Kuznetsov, "Tc (8)," *Web page at http://lartc. org/manpages/tc.txt*, 12 2001.
[25] G. Combs, "Tshark – the wireshark network analyser," *Web page at http://www. wireshark. org*, 2006.
[26] M. Team, "Mplayer – the movie player," *Web page at http:/www. mplayer. hq. hu*, 2005.
[27] G. Combs, "Wireshark – the wireshark network analyser," *Web page at http://www. wireshark. org*, 2006.