The Bits of Silence : Redundant Traffic in VoIP

Mohammad A. Hoque
University of Helsinki
mohammad.a.hqoue@helsinki.fi

Petteri Nurmi
University of Helsinki
petteri.nurmi@cs.helsinki.fi

Matti Siekkinen
University of Helsinki
matti.siekkinen@helsinki.fi

Pan Hui
University of Helsinki, HKUST
pan.hui@helsinki.fi
panhui@cse.ust.hk

Sasu Tarkoma
University of Helsinki
sasu.tarkoma@helsinki.fi

ABSTRACT

Human conversation is characterized by brief pauses and so-called turn-taking behavior between the speakers. In the context of VoIP, this means that there are frequent periods where the microphone captures only background noise – or even silence whenever the microphone is muted. The bits transmitted from such silence periods introduce overhead in terms of data usage, energy consumption, and network infrastructure costs. In this paper, we contribute by shedding light on these costs for VoIP applications. We systematically measure the performance of six popular mobile VoIP applications with controlled human conversation and acoustic setup. Our analysis demonstrates that significant savings can indeed be achievable – with the best performing silence suppression technique being effective on 75% of silent pauses in the conversation in a quiet place. This results in 2-5 times data savings, and 50-90% lower energy consumption compared to the next better alternative. Even then, the effectiveness of silence suppression can be sensitive to the amount of background noise, underlying speech codec, and the device being used. The codec characteristics and performance do not depend on the network type. However, silence suppression makes VoIP traffic network friendly as much as VoLTE traffic. Our results provide new insights into VoIP performance and offer a motivation for further enhancements to a wide variety of voice assisted applications, as such intelligent home assistants and other IoT devices.

CCS CONCEPTS

• **Networks** → **Network performance evaluation**; • **Computer systems organization** → **Embedded systems**;

KEYWORDS

VoIP, speech codec, turn-taking, silence suppression, VoLTE, IoT

1 INTRODUCTION

Voice over IP (VoIP) has rapidly evolved into the de-facto method of voice communication across a wide range of devices and platforms. Examples range from smartphones to speech controlled home/personal assistants [4, 9, 22], and even environment and wildlife monitoring sensors take advantage of VoIP [24, 48]. The performance of VoIP applications is governed by *speech codecs*, which determine how audio information is encoded and transmitted. For example, WhatsApp and Skype use variants of the Opus codec [68], whereas Facebook Messenger uses internet Speech Audio Codec (iSAC) [36]. Similar codecs are also used by wireless earbuds and headsets [15], personal and home assistants, such as Alexa and Google Home [4, 9]. While the main task of a speech codec is to determine how information is encoded and decoded, it also determines how transmissions can be adjusted, e.g., by taking advantage of dynamic transmission rates or switching to a discontinuous transmission mode (DTX). Although these measures were originally designed for supporting low bandwidth communications, they have been adopted as a mechanism for reducing resource and bandwidth usage with the exponential increase in Internet traffic recently. For example, Opus supports DTX during silence.

Silence is a fundamental characteristic of speech. Indeed, human conversations are characterized by brief pauses and so-called turn-taking behavior whereby people alternate between who is currently the active speaker [35, 60]. In the context of VoIP, this means that there are frequent periods where the microphone only captures background noise – or even silence whenever the microphone is muted. Unless accounted for, this can result in significant processing and communication overhead, as the corresponding packets are effectively empty. These transmissions also prevent the network interface from taking advantage of power-saving optimizations [57]. VoIP applications should be able to mitigate those effects of the silent periods by alleviating the transmission of unnecessary information. Silence suppression is also essential for minimizing interference in the radio spectrum. For example, WiFi, wireless earbuds, cordless phones, household appliances ranging from microwave ovens to baby monitors, and diverse IoT devices increasingly all share the unlicensed radio spectrum [59].

Despite the importance of silence suppression, surprisingly little research has been carried out on examining how silence periods are handled by diverse VoIP applications and their impacts on energy and radio resource consumption. Indeed, previous research has focused on developing energy-saving mechanisms that can exploit silence in VoIP traffic [53, 57] and on analysing corresponding network flow properties, e.g., for classifying VoIP traffic [27], for speaker or dialect identification [43, 67, 72] or for adapting buffer or smoothing jitter delays [29].

In this paper, we contribute by systematically analyzing the performance characteristics of silence suppression techniques in speech codecs, through five different and popular VoIP applications on smartphones (Viber [7], Whatsapp [8], Facebook Messenger (Facebook) [5], Skype [6], Duo [3] that each use a different voice codec. As part of our analysis, we separately investigate silence suppression in Voice over LTE (VoLTE) traffic. We analyse codec performance through these leading VoIP applications on smartphones with/without wireless earbuds, as their implementations of codecs are likely to be highly optimized to provide best overall user

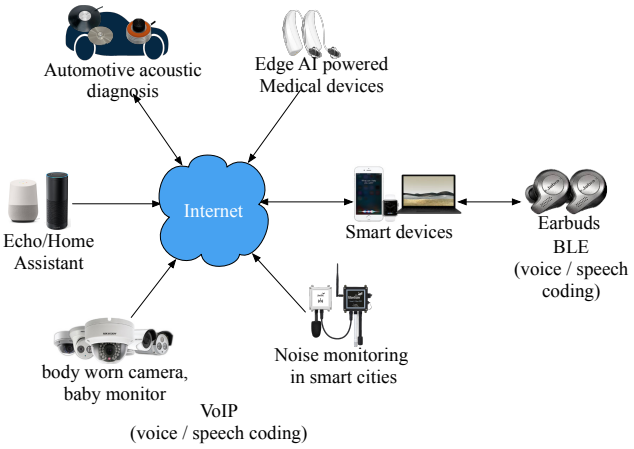


Figure 1: Common examples of smart devices and applications that utilise speech codecs.

experience. We carry out our analysis by developing an acoustic measurement setup which allows controlled evaluation of the effects of silence suppression on VoIP applications in a range of settings. We investigate how the different VoIP communication means adopt silence suppression, characterise the mechanisms that they employ, and quantify traffic, energy and LTE resource usage. We carry out our investigation considering three conversation contexts where the silence suppression mechanisms are expected to improve the resource utilization of VoIP applications. These contexts represent different points along the background noise spectrum of everyday situations (quiet environment, and two noisy environments with varying characteristics of noise).

Our results show that codecs in contemporary VoIP applications have diverse measures for silence periods, with characteristics of these measures varying significantly across applications. These characteristics have wide-ranging effects on data savings, energy consumption, networking resources, and even privacy for various VoIP applications. While we investigate mobile VoIP applications, our findings are relevant and applicable to the wide range of devices and applications that utilise voice codecs – see Figure 1 for examples of these kinds of devices.

Highlights of Findings and Analysis

Silence Suppression. We demonstrate that there is significant variability in the use of silence suppression techniques across the applications, and their effectiveness is highly variable across call contexts. Facebook and Viber offer the most savings, followed by WhatsApp. Facebook suppresses 75% of the silences in speech in a quiet room, but only 48% effective in a coffee shop or in a living room setting with ambient noise. For Viber, the corresponding savings are 39% and 30%. Consequently, Facebook transmits 2/3 times less traffic compared to Viber. Although WhatsApp employs silence suppression, it does not eliminate traffic, and fails to offer any data savings when the context changes. Skype offers significant data savings when silence is suppressed by muting the microphone.

Resource Effectiveness. We demonstrate that Facebook and Duo are the least and most energy consuming applications, respectively. Facebook is 25-190% more energy efficient for different noise contexts compared to other applications. Skype and WhatsApp perform similarly, though Skype has higher bitrates than WhatsApp. Viber also suppresses traffic for silent periods; however, it consumes more energy than Skype and WhatsApp, and the likely reason is the codec. Duo consistently consumes more energy in the presence or absence of noise. However, a group conversation may have 2-3 time energy savings, when most of the participants suppress silence by muting the microphones.

Network Flow Properties. Facebook, Viber, and WhatsApp adapt their traffic and flow properties according to conversation patterns. Facebook is the best alternative for conversations in a quiet place, whereas Viber and WhatsApp are suitable for noisy places considering data savings and energy consumption, respectively. Subsequently, the VoIP traffic from these applications is network friendly as much as VoLTE, as we demonstrate. The usage of Bluetooth low energy (BLE) earbuds and changes in network type, i.e., WiFi/LTE, do not affect the performance of the applications.

2 EXPERIMENTS

We analyze silence suppression, noise resiliency, energy consumption, and network friendliness in mobile VoIP through carefully controlled benchmarks conducted using a representative set of applications and conversation contexts. In the following, we describe the applications considered in our study, acoustic analysis of overall experimental setup, and the tools used to collect various measurement data.

2.1 VoIP Applications

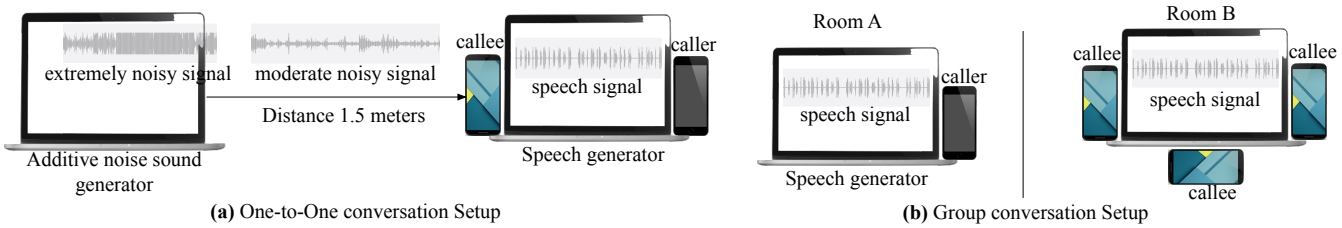
We analyse codec performance by considering five popular mobile VoIP applications:

Skype establishes P2P sessions for one-to-one calls. It relies on a server for conferencing calls with multiple participants. The network consists of two types of nodes: ordinary hosts running Skype and supernodes acting as gateways to the Skype network. Any node with the public IP address and adequate resources could operate as a supernode. At present, there are a fixed number of dedicated supernodes operated by Skype in the cloud. Traffic is transmitted directly over UDP, unless one of the devices is behind a port-restricted NAT and UDP-restricted firewall, in which case traffic is routed through the super nodes [19]. The latest Skype applications use Opus codec for encoding voice [68].

Viber uses dedicated and dynamically allocated servers to mediate the communication between the call participants [49]. Similarly to Skype, signaling takes place over TCP. The media is encoded with IP-MR codec [1], and UDP packets are exchanged through the dynamically assigned servers.

WhatsApp uses similar client-server architecture as Viber. The call signaling takes place over TCP, and the media is transmitted over UDP. Similar to Skype, it uses Opus [42].

Facebook Messenger & Duo use WebRTC for VoIP call connections. WebRTC [36] relies on the applications for P2P call setup, and media is exchanged directly between the participants over UDP [55]. WebRTC integrates several codecs, such as Opus, iSAC, G.711, and G.722 [2]. The Facebook messenger and Duo may use any of these codecs depending on the platform and peer.



(a) One-to-One conversation Setup **(b)** Group conversation Setup

Figure 2: Acoustic experiment setup for one-to-one (a) and group conversations (b).

We chose these applications as they are the most widely installed applications on both iOS and Android devices, and as they integrate different codecs. The applications are also highly popular. Facebook and WhatsApp messenger reportedly have over one billion active users per month [64], whereas Skype and Viber report around 300 million active users per month [21, 65]. For Duo, exact statistics are not openly available, but based on downloads we would expect similar numbers also.

2.2 Acoustic Experiment Setup

We conduct all experiments with one-to-one conversations stationary in a 6 m^2 room and use different audios to emulate background noise or contexts as shown in Figure2(a). A MacBookPro manipulates the background audio to the desired level in the room, as described in the following section. A second MacBookPro produces the speech signal, and two smartphones are placed nearby it's speakers. This setting allows us to measure the effectiveness of silence suppression techniques on two different devices at the same time and offers control over noise levels and speech patterns, unlike a conventional setup using two speakers in separate rooms. We note that, while we use laptop speakers for playing out the conversations, this is unlikely to affect the results. Human speech frequency is in the range 85 to 250, depending on the human speaker's age, gender and other factors. The frequency response of microphones and speakers is optimized for this range, resulting in high fidelity output and capture of voice. Unnatural sounding artefacts generally are result of pink noise, which mostly affects higher frequencies of the audio spectrum. As these artefacts reside in different frequencies than speech, they can be filtered out by the codec and hence they do not interfere with silence suppression.

For group conversations, the devices of the caller and the callee were in separate rooms with similar background noise levels and without any additive noise, as shown in Figure 2(b). The audio volume of the devices was set to the maximum during the calls. The speaker of the smartphones was switched off to avoid echos from a double conversation effect, as the participants are very close and listening on the same conversation.

To obtain further insights into behaviour of silence suppression mechanisms with different codecs and different types of smart devices, we repeated the one-to-one conversations using two BLE earbuds, Airpod 2 and Jabra 65t Elite. These earbuds use sub-band coding (SBC) for Bluetooth. SBC also supports above mentioned codecs and DTX [15], and negotiates the codec call-by-call basis.

2.3 Conversation & Contexts

We perform our experiments considering three conversation contexts that have been designed to emulate common everyday situations and be in line with daily routines [18]. The three contexts we consider are (i) noiseless conversations, (ii) conversations with moderate noise, and (iii) conversations with extreme noise. The first two contexts represent typical indoor contexts, and the last one emulates a noisier outdoor context, in line with characteristics of daily routines.

(1) Noiseless Conversations. We first initiate VoIP calls between two devices in a noise-reduced room. Devices of both call participants were placed in the same room. The average sound level in the room was 34.3 dBA, which is in line with indoor sound levels reported in other studies [47]. The audio conversation was recorded from a high definition YouTube recording of everyday English conversation[1] and replayed on a MacBook Pro after initiating the calls. The conversation has moderate pauses and speech events, as shown in Figure 3 (b). We computed the duration of the speech and pause events using a moving maximum function in Matlab [31]. We used the amplitude 0.01 as the threshold for the moving function to separate the silent periods. This threshold value is derived from the tiny spike at 52nd second in Figure 3 (a). The output volume of the laptop was 75% of the maximum. In Figure 3 (c), we notice the sound level of the speech varies around the average sound level of the room. We conduct two sets of experiments. First, both the caller and callee remain silent. Second, both of them are exposed to the conversation speech. These experiments demonstrate whether the applications can recognize the long and brief pauses during conversations at home or office, i.e., in a noise-free or noise-reduced environment.

(2) Conversations with Noise. In urban settings, we are exposed to more sound in indoor public places or outdoor. For example, in NYC, people experience a mean sound level of 75.6 and 74.4 dBA due to pedestrian and transport traffic [50]. The community gathering in a living room can create an annoying sound of 55 dBA [70]. We repeat experiments by playing two other YouTube clips, one of a coffee shop[2] and one of a kindergarten playground[3], on top of the conversations. The corresponding sound level of the audios is presented in Figure 3 (c). We played the audios on another MacBookPro and measured the sound level on iPhone 6, which was from 1.5 meters away from the other MacBook. Similar to the earlier setup, the output volume of both laptops was 75% of the maximum. Figure 3 depicts the distributions of sound levels in these audios

[1] https://www.youtube.com/watch?v=OHK-xsvW0TQ
[2] https://www.youtube.com/watch?v=BOdLmxy06H0
[3] https://www.youtube.com/watch?v=wAjKpdokhls

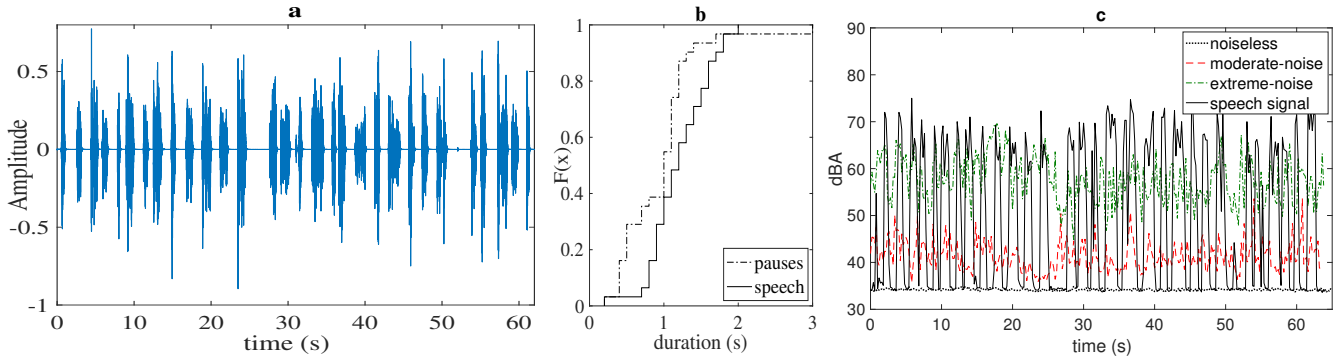


Figure 3: The acoustic properties of the conversation speech and the sound pressure level with different experiment setup. (a), The raw speech signal (b), The duration of pauses in the conversations. (c), Sound pressure levels for different call contexts measured with Decibel [11]. Decibel uses on-device microphones and measures sound pressure level at 5Hz. The measurement range is 30-130 DBA, as the mics on smartphones are usually designed for the human voice.

Conversation Type	Caller	Callee
One-to-One(VoIP)	iPhone 6	Nexus 6
One-to-One (VoLTE)	Xiaomi Mi8	Samsung S9
Group (VoIP)	iPhone 6	Nexus 6, Mi8, S9

Table 1: Conversation Experiments and the Devices used.

over time. These experiments assess whether the applications can identify pauses when conversation context becomes noisier.

(3) Muted Conversations. During the turns of an inactive speaker, the applications may transmit ambient noise. Muting the microphone can suppress such noise, and it is prevalent during group conversation over VoIP. We concatenated the same audio signal presented in Figure 3 (a) for 195s, to simulate the group conversation. We conduct two sets of such experiments with the VoIP applications. We formed application-specific groups of four participants, except for Duo, which did not support this functionality across the test devices during the time of the experiments. First, all participants actively participate in the conversation and then leave one after another in the noise-free setup emulating group dynamics. Next, we experimented by muting the microphone of the devices. Three participants mute their microphones one after another.

2.4 Measurement Tools & Configurations

We experiment with five VoIP applications with different conversation types, as shown in Table 1. The one-to-one conversations took place in the presence of WiFi and LTE networks. We also used BLE headsets during some calls. On the other hand, group conversations took place over LTE without any earbuds. Conversation for every context was repeated three times for each of the applications.

Network Layer Traffic Measurements. VoIP traffic is captured using *tcpdump* on a remote virtual interface [14], RVI, on iPhone 6 from a MacBookPro. Traffic is captured using *tcpdump* on the virtual interface. In total, we analyze around 250 VoIP call traces of 290 minutes. Unlike the VoIP traffic, VoLTE traffic does not travel through the TCP/IP stack of the OS kernel. Therefore, we could not capture VoLTE traffic.

Physical Layer Resource Measurements. We also logged LTE physical layer resource block (RB) and VoLTE information using the Network Signal Guru (NSG) [13]. NSG only works on rooted devices with LTE Qualcomm chipset (Nexus 6, Xiaomi Mi8). RB is the unit of LTE network resource [41].

Energy Consumption Measurements. We separately measured the energy consumption of Nexus 6 during VoIP conversations according to the contexts. Nexus 6 has a Coulomb counter interfaces that enable on-device current measurement at 6Hz [12]. We developed an energy profiler that uses Android APIs to sample the run time current consumption of Nexus 6 at 2Hz.

3 ONE-TO-ONE CONVERSATIONS

According to human conversation theory, an inactive speaker can be silent while waiting for his turn during conversations while listening to the active speaker. And the active speaker may have average pauses of 200-600 ms in the speech [35]. In this section, we demonstrate that only a few modern VoIP applications can detect silence in conversations and suppress traffic during those silent periods. Indeed, their efficiency depends on call contexts or surrounding noise. They are less efficient in a moderately noisy environment, such as in a coffee shop or a living room compared to a silent place. Their performance diminishes as the context becomes comparatively noisier, and, as the context changes, the applications employ different packet sizes and packet gaps.

3.1 Silent Turns and Inactive Speakers

During a conversation, the active speaker utters phrases subsequently with brief pauses, whereas the inactive speaker listens to those phrases and waits for his/her turn. This waiting time can vary from 600 ms to 2 seconds in natural conversation [44]. We first examine how many bytes an application sends to the other end when the speaker is *inactive or silent*, i.e., only background noise is captured and transported between the devices. We use bits per second to benchmark the application performance.

The results in Table 2 show that Facebook exchanges the smallest amount of traffic, followed by Viber, and WhatsApp. These three applications would exchange only a hundred bytes for an inactive speaker during the 1 second of waiting or inactive turn. The bitrates of Skype and Duo are 4-20 times higher than the other three applications and they may not suppress silence at all.

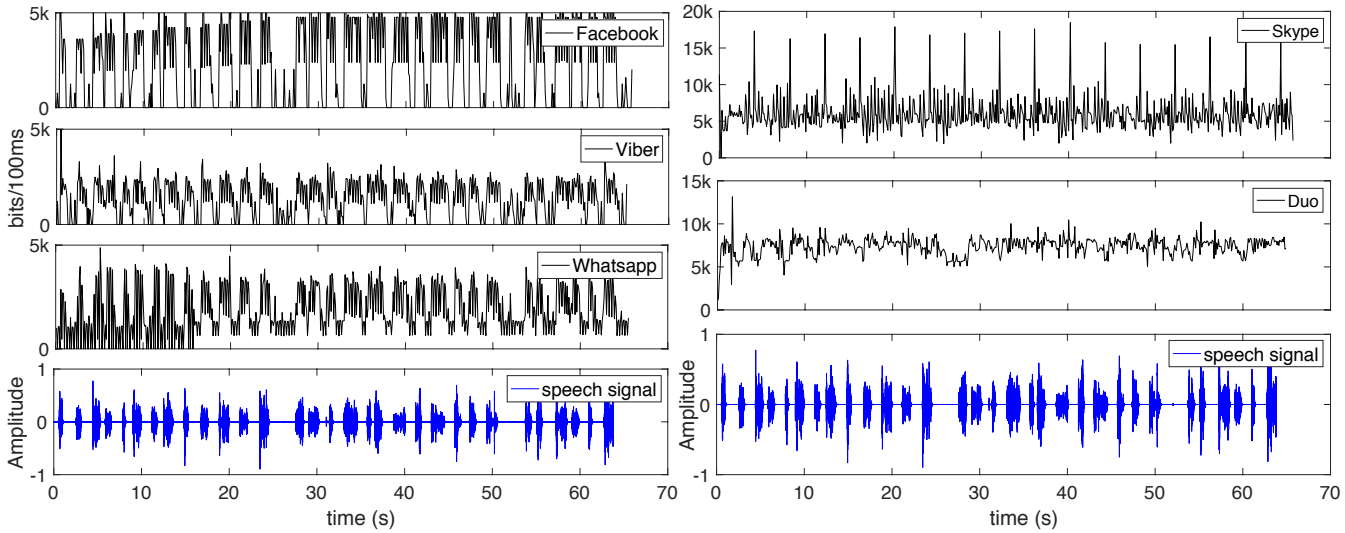


Figure 4: The synchronization of speech signal and VoIP traffic of five applications without any additional noise.

Device	WhatsApp	Skype	Duo	Viber	Facebook
Nexus 6	10 (0.5)	41 (7)	55 (2)	5.8 (4)	2.4 (1)
iPhone 6	14 (4)	42 (8)	63 (4)	5.6 (6)	3.7 (1)

Table 2: Average bitrates (kbps) of the VoIP applications during the silent turns, and standard deviation of measurements.

During the silent turns, both Facebook and Viber flows have the largest inter-packet gaps, whereas WhatsApp packets have mostly 60 ms gaps. The distributions of packet size in the silent calls also vary among the applications. Viber has smaller bitrates, and uses smaller packets (≈55 bytes) compared to the other applications. Facebook only sends 2-3 large packets (≈300 bytes) in a second, whereas Viber and WhatsApp send 16-20 smaller packets.

3.2 Contexts and Pauses in Conversations

The previous Section demonstrated that three of the tested VoIP applications use very low bitrates, and two applications apply very high bitrates during the silent turns when the speaker is continuously inactive. We next examine how the applications perform under natural conversation and how different call contexts affect their performance, demonstrating that (i) length of pauses is a significant factor in determining when silence suppression is used; and (ii) amount of noise significantly impacts silence suppression performance.

(1) Noiseless Conversation. Figure 4 illustrates bitrates of the applications (bits/100ms). WhatsApp maintains a low bitrate during the first 30 seconds of the calls, but overall its bitrate fluctuates during the conversation. Viber's bitrate also varies similarly. Similarly to WhatsApp and Viber, Facebook has fluctuating bitrates, which suggests that these applications may detect silence and suppress the packets for those brief pauses in the conversations as well.

To understand this effect better, we synchronize the bitrates with the audio signal. Figure 4 presents the actual signal and maps it according to the timestamp of the packets in the traces. In the synchronized signal, the conversation begins exactly at the 10th second after the call setup. It also demonstrates that the fluctuations in the bitrates are according to the signal. The bitrates of Facebook and Viber drop to zero during the corresponding brief pauses in the signal. After 15th second, WhatsApp maintains a minimum

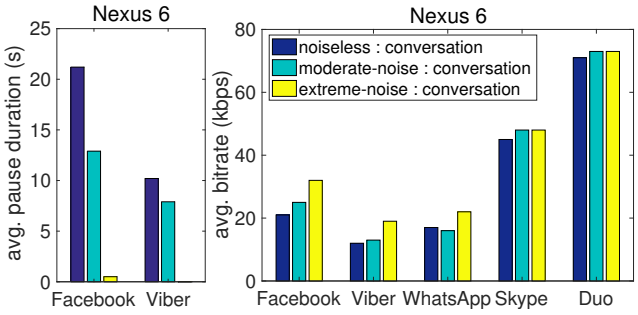


Figure 5: The duration of pauses for different call contexts on Nexus 6 (Left) and the bitrates of the applications to those context (Right).

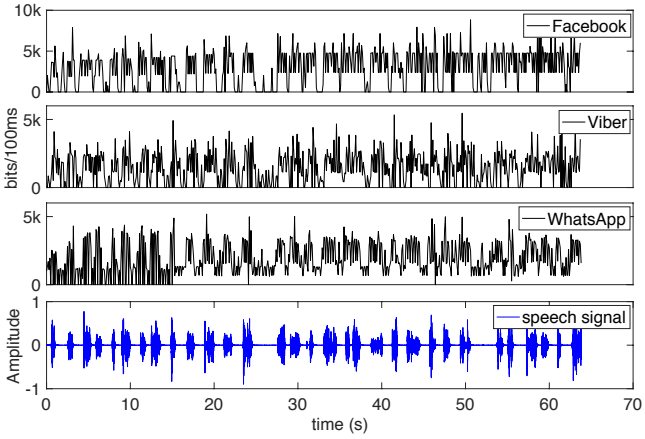


Figure 6: The synchronization of speech signal and bitrates of three applications in the presence of moderate noise.

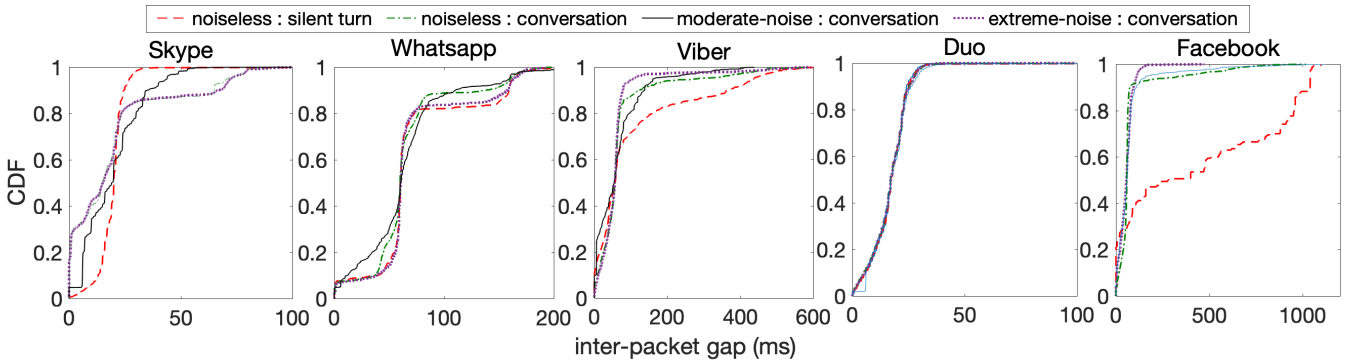


Figure 7: Inter-packet gaps of silent turns and conversations originating from Nexus 6 for different contexts.

bitrate higher than zero. Nevertheless, the pattern from silence suppression is obvious.

For Skype and Duo, Figure 4 does not demonstrate similar changes in the bitrates according to the speech signal. Rather, there are relatively small changes in bitrates compared to the silent turns, as shown in Figure 5. Duo's average bitrate increases by 15 kbps compared to the silent turns presented in Table 2. Their performance is in line with their limited suppression performance demonstrated in the earlier section. In Figure 5, the average bitrate is computed from the total amount of bytes exchanged during total pause duration from three traces for a context.

From the presented sampled traces, in Figure 5, we compute the total duration of pause events for five applications. We consider 200 ms as the minimum gap in the traces, as there is a pause of 200 ms in the conversation. Figure 5 shows that Facebook and Viber on Nexus 6 do not send traffic for 21.2 and 10.2 seconds, respectively, during the total 26.5 seconds of brief pauses in the speech. Facebook also exchanges the smallest amount of traffic, followed by Viber and WhatsApp. WhatsApp generates mostly 100 ms gaps during the first 15 seconds, which do not reflect the pauses in the signal during that period. After that, no gaps exist, and there is always a small amount of traffic during those pauses, as shown in Figure 4. This pattern is according to lower bitrates observed during the silent turns presented in Table 2. These results suggest that Facebook messenger is the most efficient application in suppressing silence in a noise-free conversation. However, Viber has the smallest bitrate, followed by WhatsApp, Facebook, Skype, and Duo (Figure 5).

(2) Conversations with moderate noise. We next repeat the experiments by adding moderate background noise to the environment, as discussed in Section 2. The noise emulates background conversations among multiple people in a coffee shop or in a living room. We play the noisy audio and then establish the call at 5th second of noise playback, and the conversation speech begins at the 10th second of the conversation. This ensures that all the applications experience similar noise during conversations.

Figure 6 shows that the bitrates of the applications fluctuate even with moderate noise, except naturally Skype and Duo, which do not seem to incorporate silence suppression. The figure demonstrates that Facebook and Viber have moderate gaps according to the speech signal. Figure 5 shows the duration of pauses for applications with moderate noise in the room. The pause duration decreases, and bitrates increase compared to noiseless conversations. The savings from pauses in Facebook and Viber reduce to 48% and 30% of total pause duration, respectively.

(3) Conversations with extreme noise. Finally, we compare the applications under conditions emulating an outdoor context by introducing extreme noise discussed in Section 2. Figure 5 shows that only the Facebook conversations had 500 ms pauses according to the speech signal. The figure also shows that all the applications have the highest bitrates in this very noisy call context. The bitrates of WhatsApp, Viber, and Facebook increases as the context becomes noisier. As before, Skype and Duo have negligible changes in bitrates compared to the less noisy contexts.

Compared to the silent turns (Table 2), Facebook, Viber, and WhatsApp have 2-10 higher bitrates during the speech. Skype and Duo have little increase in the bitrates during actual conversation.

3.3 Contexts and Flow Properties

In Section 3.1, we have briefly discussed how the packet gaps and packet size vary among the applications during the silent turns. This section examines these properties among applications in light of the conversation context and the corresponding codec.

Figure 7 and 8 show that conversation speech from all the applications experience smaller packet gaps and have larger packets compared to the silent turns. The applications also employ different flow properties as the conversation context changes from silent to noisy and noisier as demonstrated in the figures. WhatsApp has almost a constant inter-packet gap, and the packet size increases with the ambient noise. Viber, on the other hand, increases packet size and reduces the packet gap as the intensity of the ambient noise increases. In the case of Facebook, the inter-packet gap increases, and the distribution of packet size remains almost the same. Skype and Duo do not have noticeable variations in the distributions of packet gaps and packet size.

WhatsApp uses Opus codec [34], which combines SILK and CELT codecs. Among these, only SILK is a variable bitrate codec. Figure 4 shows that the bitrate fluctuates between 10-32 kbps, which suggests that WhatsApp specifically uses Opus/SILK codec. The distributions of packet gaps in Figure 7 suggest that it encodes 60 ms conversation, which is also a codec attribute. The bitrate also fluctuates according to the speech signal, which in turn emphasizes that SILK also has built-in support for silence suppression, i.e., DTX. However, the codec does not eliminate traffic completely, and the baseline bitrate is 10 kbps.

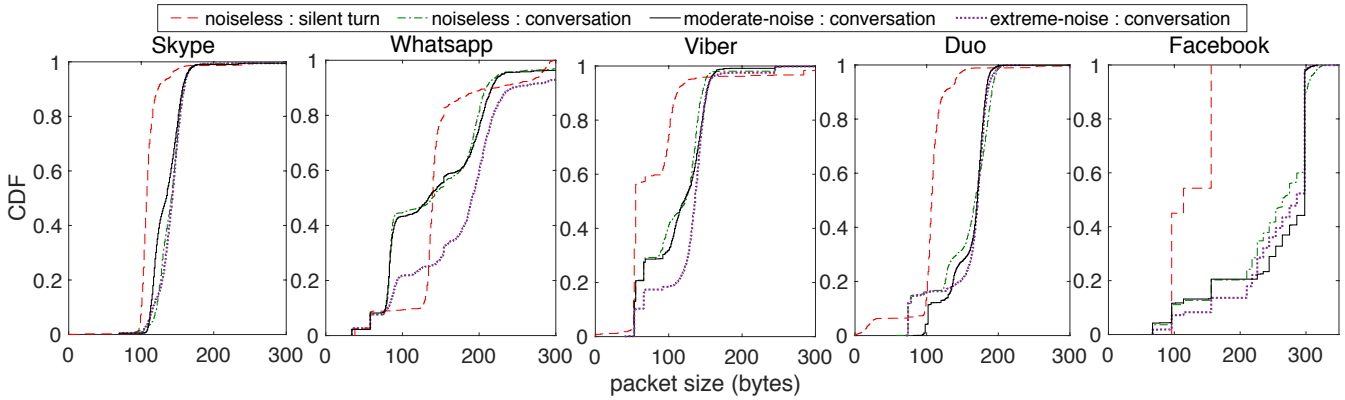


Figure 8: Comparison of packet sizes for the silent turns and different conversation contexts on Nexus 6.

VoIPApps	LTE			WiFi		
	bitrate (kbps)	pkt.gap (ms)	pkt.size (bytes)	bitrate (kbps)	pkt.gap (ms)	pkt.size (bytes)
Skype	54(12)	20(20)	145(77)	51(15)	20(7)	142(114)
WhatsApp	18(6)	69(40)	151(82)	18(7)	70(39)	195(85)
Duo	70(10)	17(10)	153(40)	66(12)	17(9)	148(22)
Viber	12(6)	71(81)	112(37)	13(6)	68(71)	125(37)
Facebook	23(13)	91(150)	243(77)	21(14)	91(135)	245(72)

Table 3: Average flow features of conversations on Nexus 6 without any additive noise. The numbers inside the parentheses represent the measure of spread (standard deviation).

Skype also uses the Opus codec. Contrary to WhatsApp's performance, we notice almost constant bitrate pattern of Skype in Figure 4 and very small packet gaps in Figure 7. Such patterns suggest that Skype specifically uses Opus/CELT, which is a low latency constant bitrate codec.

On the other hand, Viber uses IP-MR codec [1], which also supports variable bitrate. Subsequently, Figure 4 shows that Viber's bitrate alternates between 12-24 kbps. Besides, the bitrate constantly varies according to pauses in the conversation. Therefore, this codec also supports DTX. During the pauses, the average bitrate can be reduced to 5.4 kbps.

Duo uses WebRTC and thus can use any of the codecs supported by WebRTC. The flow properties, i.e., packet size (160 Bytes) and packet gaps (17 ms), suggest that Duo uses G.711 with constant 64 kbps bitrate. This codec does not have an integrated silence suppression technique [76]. Consequently, the bitrate is always above 50 kbps, as shown in Figure 5. Although Duo does not have noticeable changes in the packet gaps, it uses smaller packets during the silent turns.

Similar to Duo, Facebook Messenger relies on WebRTC framework. However, Facebook uses iSAC codec [33]. The bitrate patterns in Figure 4 and 6 suggest that iSAC is a variable bitrate codec. The bitrate pattern fluctuates with the speech signal, as it has integrated DTX [51]. In the absence of speech, the bitrate reduces to 2.5 kbps.

3.4 Device Variation & Connectivity

Note that a particular codec may not be tight to a particular application and vice versa. For example, Facebook Messenger uses the

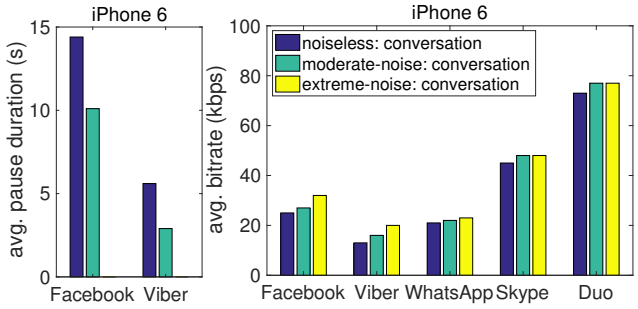


Figure 9: The duration of pauses for different contexts on iPhone 6 (Left) and the bitrates of the applications during the pauses according to those context (Right).

Opus codec with Mozilla Firefox browser [33], and the WebRTC protocol uses G.711 as the fallback codec [26].

In Table 2, we have already presented that the applications on the iPhone have higher bitrates than those on Nexus 6 during the silent turns. Therefore, we also computed the total pause duration on iPhone 6 during conversations. Figure 9 shows the average silent periods from three traces for a context. In the figure, the average bitrate is computed from the total amount of bytes exchanged during total pause duration from three traces for a context. The figure demonstrates that Facebook and Viber suppress traffic also on iPhone 6 according to the noise. However, the silent periods are smaller, and the applications have higher bitrates on iPhone 6 compared to Nexus 6. In other words, silence suppression effectiveness varies across devices and is more efficient on the Android handset than on the iPhone in our experiments. We could not find a noticeable difference with Nexus 6 flow properties other than larger packet sizes.

We repeated the experiments with two additional configurations; conversations via BLE earbuds and WiFi. We connected a Jabra 65t Elite earbud and Apple AirPods 2 with Nexus 6 and iPhone 6, respectively. These two devices use the SBC having support for the codecs being used by the applications on smartphones. Nevertheless, the applications have had similar performance in silence suppression, as stated above, and the traffic pattern described earlier. Although an early study found changes in bitrate as the connectivity changes from Ethernet to WiMax [32], in our research switching from LTE

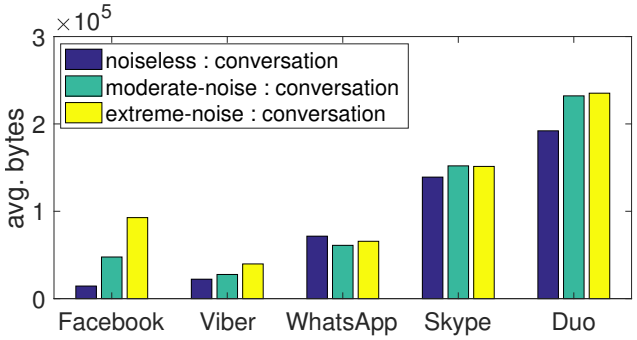


Figure 10: The amount of data sent by Nexus 6 during the conversation pauses for different contexts. The average is computed from three traffic traces for a context.

to WiFi did not affect the performance and flow properties as shown in Table 3. It is very likely that applications adapted bitrates as the earlier wireless networks had limited bandwidth.

3.5 Summary

Our results demonstrate that VoIP applications can significantly save traffic by suppressing silence during the silent turns and pauses in the conversation speech. Facebook and Viber can suppress the very minimum of 200 ms gaps. Facebook and Viber can suppress traffic for 75% and 39% of total silent periods, respectively, in a silent place. With moderate noise in the room, their achievements reduce to 48% and 30%, respectively. However, these two applications offer 2-3 times more data savings when conversations take place in a less noisy environment, followed by WhatsApp. Although WhatsApp performs silence suppression, the approach does not eliminate traffic. Skype and Duo do not suppress traffic for silence.

G.711 (Duo) and Opus/CELT (Skype) are the constant bitrate codecs. In contrast, Opus/SILK (WhatsApp), iSAC (Facebook), and IP-MR (Viber) are the variable bitrate codecs. The applications suppress silence with the help of DTX. At the same time, they react to the context noise to different degrees. Figure 10 demonstrates that Facebook exchanged 6-7 times more traffic in the noisier or noisiest context compared to the noiseless conversation. The reaction of Viber to moderate noise is negligible compared to Facebook; however, the data overhead is twice in the noisiest context. Surprisingly, WhatsApp sent almost a similar amount of data irrespective of the conversation context. WhatsApp bitrate does not increase either significantly due to additive noise. This suggests that Opus/SILK does not react to noise. Figure 10 illustrates similar performance of Skype and Duo.

The performance of the applications suggests that Facebook, Viber, and WhatsApp reduce data waste significantly by detecting silence and suppressing packets compared to Skype and Duo. However, their performance can vary according to the device, too, as we have compared between Nexus 6 and iPhone 6. Furthermore, the addition of BLE headsets and the selection of WiFi over LTE does not affect the selection of codecs or the performance of applications. In Section 5, we measure the contribution of silence suppression techniques and flow properties on energy consumption and LTE physical layer resource consumption.

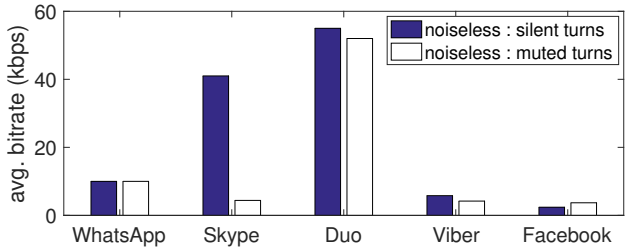


Figure 11: Bitrates of the applications during the silent and muted turns on Nexus 6.

4 GROUP CONVERSATIONS

With multiple participants in a conversation, a user may have to wait for a longer time for the turn, and such silent turns are more prevalent compared to the one-to-one conversations. Therefore, muting microphones can be more effective in suppressing surrounding noise. In this section, we first study the performance of applications or codecs for one-to-one conversations by muting the microphone during silent turns. After that, we investigate group conversations and how muting the microphone, a common practice during group calls, influences the silent turns. We use the concatenated conversation speech specified in Section 2.

4.1 Muted Turns and Flow Properties

The bar chart in Figure 11 shows the bitrates of the applications during the muted turns. Facebook has the lowest bitrate, followed by Viber, Skype, WhatsApp, and Duo. Duo's muted stream has 5-10 times higher bitrate than the other applications. Since the microphone is mute, the applications have nothing to encode, and the corresponding frames are null. The applications or the native audio interface, i.e., driver, generate these null or muted frames. Since different applications have different rates in generating muted frames on the same device, the applications or the corresponding codec dictates muted frame generation.

Muted turns of the applications have similar bitrates (Figure 11), and flow properties to the silent turns from the VoIP applications except for Skype. Skype muted packets have more than 500 ms gaps, and the maximum is approximately 1 s. Facebook also generates muted packets with similar gaps. Viber muted packets also have longer gaps; however, they are separated by less than 500 ms. WhatsApp streams, on the other hand, have 60 ms gaps. Muting microphone does not change the packet gaps of Duo streams. WhatsApp generates packets of specific 71 bytes, whereas Duo packets are mostly 115 bytes. Viber generates tiny packets, and most of the packets are less than 55 bytes.

4.2 Conversations with Muted Turns

In addition to Nexus 6 and iPhone 6, we used Samsung Galaxy S9 and Xiaomi Mi 8. The calls were initiated from the iPhone 6 (caller), and all other devices accepted the calls (callees). All the participants had outgoing traffic and combined traffic from 3 other participants.

(1) Noiseless Group Conversation. We first examined the flow properties of a quiet VoIP group conversation. All participants accepted the call within the first 10 seconds of the call. All of them used the same concatenated speech signal. The bitrates of the aggregated streams are 99, 54, 37, 100 kbps, respectively, for Skype, WhatsApp,

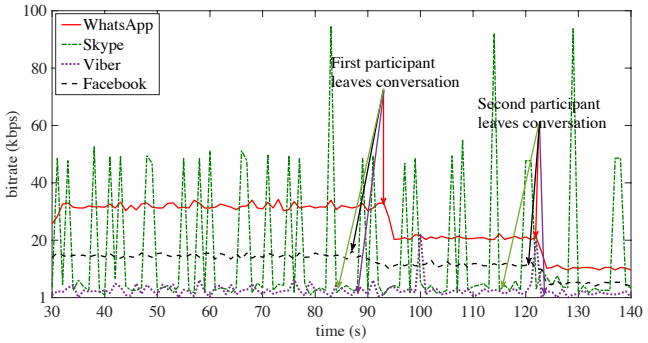


Figure 12: Muted group conversation bitrates of the applications observed on iPhone 6 during the common muted period among the participants.

Viber, and Facebook. The bitrate of an individual participant is one third of this aggregated bitrate.

(2) Muted Turns in Group Conversation. The calls were initiated from the iPhone 6 (caller), and all other devices accepted the calls (callees). After the first 30 seconds of the call, we muted all the callee devices for 60 seconds. After the mute period had ended, devices dropped the call one at a time every 30 seconds (i.e., at 90, 120, and 150 seconds). The iPhone 6 had combined incoming traffic of 3 muted participants. The other devices had a combined incoming bitrate from 2 muted and a non-muted participant.

4.3 Summary

Viber consumes the least data during group conversations, followed by WhatsApp and even when the microphone is muted. Muting microphone for Skype turns can save 11 times data. The aggregated bitrates of the muted traffic from 30th to the 140th second of conversation are illustrated in Figure 12.

The figure shows that Skype generates bursty traffic. As the number of participants decreases, the number of bursts also decreases. The average bitrate of the grouped muted traffic of Skype is 16 kbps, which is the sum of bitrates from three participants. In the case of WhatsApp, the bitrate is constant 30 kbps, which decreases as the number of participants decreases. Similar to Skype, the aggregated bitrate of WhatsApp is the sum of traffic from three muted participants, as shown in Figure 11. Facebook behaves similarly.

According to the bitrates of an individual muted turns presented in Figure 11, Viber should have had an aggregated bitrate of 15 kbps. Interestingly, Viber's group bitrate is even smaller than the muted turn of a single participant. With additional muted group calls, we have found that an individual participant always has 4.5 kbps bitrate and such packets are dropped by the Viber server.

Surprisingly, all the applications or the corresponding codecs generate traffic even when the microphone is muted. We can think of two reasons behind such muted packets. First, if there is no conversation data, the application generates muted frames and sends to avoid the re-initialization latency of the audio chipset at the other end hardware [30]. Second, the muted packets may act as the remedy for NAT port binding issue for UDP flows, as those packets do not allow to expire the NAT port binding timer [19].

5 RESOURCE CONSUMPTION

In the earlier sections, we have demonstrated how various applications or codecs deal with silent turns, pauses in conversation and noise contexts. In this section, we investigate the contribution of silence suppression and reciprocal contexts on LTE resource consumption and the energy consumption of mobile devices. It might also be interesting to compare all these applications with VoLTE, as the resource and energy consumption of VoLTE might be more conservative compared to these applications.

5.1 LTE Resource Block Allocation

NSG takes a snapshot of the LTE network status after every 500 ms and generates binary logs. We manually extracted the modulation schemes and the number of physical resource blocks for both uplink and downlink during the muted turns and extreme-noise context. Although it is not possible to characterize these physical layer parameters for every packet using NSG, our findings provide insights into the network friendliness of the VoIP applications.

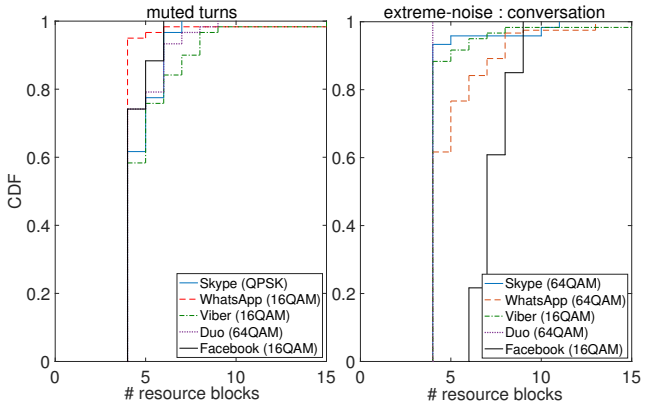


Figure 13: LTE downlink Resource Block (RB) allocation and Modulation Schemes for VoIP conversations on Nexus 6.

(1) VoIP Conversations. Figure 13 depicts that most of the downlink VoIP traffic is allocated to 4 RBs during both muted turn and the extreme noise call contexts. The size of the muted packets does not have any impact on the RB allocation. However, the bitrate plays a role in the selection of the modulation scheme. We notice that most of the Skype packets were received using the QPKS modulation scheme during the muted turns. Likewise, QPSK was applied for Facebook during the muted turns. This can be explained with the larger inter-packet gaps of Skype and Facebook. 16QAM is applied during the muted turns from WhatsApp and Viber.

In the case of extreme noise, 64QAM is applied for all the applications traffic, except Viber. We believe that Viber's smaller packet size, in Figure 8, contributes to selecting 16QAM. Facebook traffic with larger packets has higher RBs allocation, as shown in Figure 13. Group conversations required mostly 12-28 RBs with 16QAM for all the applications. When the participants were muted, Viber required 4 RBs with 16QAM.

(2) VoLTE Conversations. In this case, we used NSG on a rooted Xiaomi Mi8 device. Figure 14 shows that VoLTE conversation uses

		Downlink	Uplink
(a)	Codec	AMR-WB/12.65	AMR-WB/12.65
	Throughput	7.6 Kbps	7.6 Kbps
(b)	Codec	AMR-WB/12.65	AMR-WB/12.65
	Throughput	18.0 Kbps	18.0 Kbps

Figure 14: VoLTE codec and bitrates reported by NSG; (a) muted/silent turns, (b) conversations with noiseless/extreme-noise contexts.

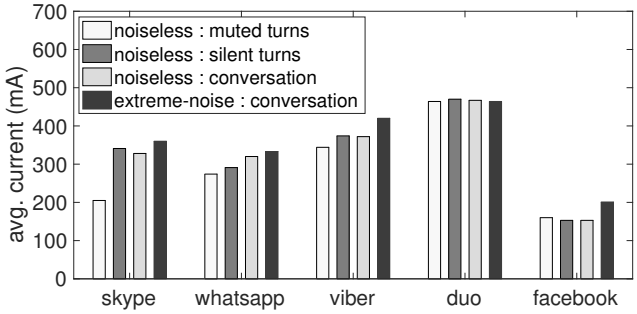


Figure 15: Current drawn by the applications on Nexus 6 for sending voice to and receiving muted traffic from iPhone 6.

AMR-WB (AMR-WB) codec [10]. When the microphone is muted, the bitrate is 7.6 kbps. Otherwise, the bitrate fluctuates within the maximum of 18.5 kbps. This suggests that this codec suppresses silence [76]. During the silent turns, QPSK was used, and mostly 3 RBs were allocated for the packets, whereas 16QAM was used, and 4 RBs were allocated for speech traffic.

All the uplink packets were encoded with 16QAM, and only 1-2 RBs were allocated irrespective of the context and application. Nevertheless, such resource consumption of VoLTE traffic is similar to the other VoIP applications, except the Facebook Messenger.

5.2 Energy Consumption

We characterize the consequence of the context of both sides of the conversation on energy consumption. We conducted three sets of follow up experiments. (i) We muted iPhone 6 to measure the energy impact of outgoing contextual conversations from Nexus 6 (Figure 15). (ii) We muted Nexus 6 to measure the energy impact of incoming contextual iPhone 6 streams (Figure 16). The display and phone speaker of Nexus 6 were off during the measurements. (iii) We measure the energy consumption of Nexus 6 for group conversations, where Nexus 6 receives aggregated conversation and muted traffic from the participants and transmits noiseless conversation. We developed an energy measurement tool for Nexus 6 to read the battery information at 2 Hz.

(1) One-to-One conversation (Transmitting Noise). Figure 15 demonstrates the Facebook and Duo are the least and most energy-consuming applications, respectively. Facebook is 25-50%, 77-130%, 76-140%, and 115-190% more energy-efficient than Skype, WhatsApp, Viber, and Duo, respectively. Skype and WhatsApp perform similarly, though Skype has a higher bitrate than WhatsApp in all the contexts except the muted stream. Interestingly, Viber consumes more energy than Skype and WhatsApp, even though it suppresses

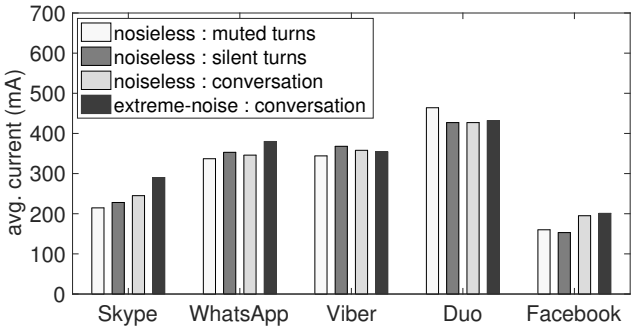


Figure 16: Current drawn by the applications on Nexus 6 for receiving voice from and sending muted packets to iPhone6.

Application	noiseless (mA)	muted turns (mA)
Skype	941	386
WhatsApp	615	361
Viber	561	331
Facebook	280	351

Table 4: Avg. current consumption of Nexus 6 during group conversations. During the muted group calls, 3 participants muted their microphones.

traffic for the silent periods, as we demonstrated in Section 3.2. This behavior was persistent across multiple measurements, and the codec is the likely reason. In general, the applications consume more energy as the context becomes noisier.

(2) One-to-One conversation (Receiving Noise). In Section 3.3, we have already demonstrated that conversation traffic from iPhone has higher bitrates than the Nexus 6. Likewise, Figure 16 demonstrates higher energy consumption compared to those presented in Figure 15. However, the energy consumption pattern is similar. Facebook consumes the least energy and outperforms other applications by similar margins. WhatsApp and Viber consume similar energy, as the silence suppression in WhatsApp does not eliminate traffic completely.

(3) Group Conversation. We also measured energy consumption during two sets of the group calls as presented in Table 4. First, all four participants have conversations without any noise. Second, only one participant speaks, and others listen to having their microphone muted. Table 4 shows that Skype is the most, and Facebook is the least energy-consuming application for noiseless group conversations. Muting microphones during group conversations can reduce the energy consumption of applications by 2-3 times.

5.3 Summary

Figure 13 emphasizes the selection of codec by the applications, and the resulting flow properties contribute to LTE RB allocation. AMR-WB and IP-MR codecs (Viber) produce small packets, which result in only 4 RBs allocation for transmitting conversation packets. VoLTE also reacts to silence, generates small packets, contributes to similar RB allocation and selection of modulation schemes. In contrast, the larger packets from iSAC contribute to a higher number of RBs allocation for Facebook. A side-benefit from silence suppression is that it can facilitate more effective network resource management

Codec-Application	Data	Noise	Energy	Network
Opus/SILK(WhatsApp)	yes	-	yes	yes
Opus/CELT (Skype)	no	–	yes	no
iSAC (Facebook)	yes	no	yes	yes
IP-MR (Viber)	yes	no	no	yes
G.711 (Duo)	no	–	no	no
AMR-WB (VoLTE)	yes	–	yes	yes

Table 5: Efficiency features of the identified applications and speech codecs.

by exploiting inter-packet gaps. For example, Skype's average inter-packet gap is 20 ms (Table 3), which means that the allocated RBs can be shared with 20 Skype calls one after another. Viber has large and varying packet gaps, and the network can share the RBs with more calls. Speech packets from iSAC require more RBs than the others; however, these RBs can be shared with more VoIP/VoLTE calls and less network friendly than IP-MR, Opus/SILK, or AMR-WB. In contrast, Duo and Skype have smaller packet gaps, and the network can share resources among limited calls compared to the other codecs. Interested readers can follow this tutorial[4] on estimating cell capacity for VoLTE calls.

The energy measurements suggest that Facebook Messenger (iSAC) is the most energy-efficient VoIP application followed by Skype (Opus/CELT), WhatsApp (Opus/SILK), Viber (IP-MR), and Duo (G.711). These results also highlight the energy efficiency of the codecs, i.e., iSAC is the most efficient among these five codecs. Unlike data savings, context does not contribute to significant energy savings for a particular application or the corresponding codec. During one-to-one conversations, a particular application does not have significant energy savings by muting microphones either. Nevertheless, the muting microphone can save significant energy across all the applications for group conversations.

The performance of the VoIP traffic in RB and energy consumption suggests that the conversation traffic from these applications also can be served similarly to VoLTE traffic. The VoIP traffic can benefit from the dedicated channel and other optimizations such as header compression [28, 52].

6 DISCUSSION

From the analysis in this work, the performance of the applications or codecs can attribute to four different high-level metrics. The attributes are data efficiency due to silence suppression, resiliency against noise, energy efficiency, and network friendliness, as shown in Table 5.

Data/Energy Consumption. Table 5 shows that silence suppressing codecs enable data and energy savings for mobile VoIP applications. Silence suppression is also vital for Bluetooth earbuds or headsets, as they are battery-powered, and the energy is being spent both by headsets and mobile devices. Beyond VoIP, silence suppression is essential for many sensor networking applications. For example, environment and wildlife monitoring sensors are battery-powered and always on.

Network Friendliness. Periods of heavy network demand, whether due to holidays, disasters, or other events, are challenging to network capacity due to continuous peak demand of network resources. Our results showed that use of silence suppression can save 2-5 times data depending on the application and the level of background noise. This suggests that silence suppression could help to reduce network bandwidth use at peak demand (e.g., in LTE networks this results in lower consumption of resource blocks). Our findings also suggest that silence suppressing codecs would improve the performance of cross-technology devices, e.g., BLE headsets, earbuds, and other devices, operating on the shared unlicensed radio spectrum [56].

Privacy. While the gain in data waste and energy consumption are significant due to silence suppression, this conversely can harm privacy. The focus of our work is not on exploring the range of such privacy violations, indeed, several prior studies have shown that traffic pattern can be exploited to identify the speaker [43], and dialect [69, 71, 72]. Instead, our aim has been to highlight characteristics of network patterns, and demonstrate how these correlate with speech activity. The analysis in Section 3 demonstrated that Facebook, WhatsApp, and Viber's packet size distributions are affected by the contexts. In contrast, constant bitrate traffic is resilient against such attacks [67]. Traffic shaping and padding [17] or traffic morphing [73] towards constant bitrate traffic can be used to mitigate privacy vulnerabilities at the expense of higher traffic.

Performance and Context-Aware Codec Selection. At present, interoperability across devices and platforms play roles in selecting the appropriate codecs. Our findings in Table 5 suggest that voice processing applications can benefit by negotiating codec according to desired features, e.g., noise robustness, energy consumption or privacy protection. In Section 3.5, we demonstrated that changing network connectivity to WiFi does not change the application performance. Nevertheless, the applications can negotiate constant bit rate codecs when connected to public WiFi networks for privacy-preserving communication. Always on devices like Alexa or Google Home also rely on the Opus codec for speech processing [4, 9]. Since these devices stay mostly indoor, silence suppressing codecs could provide significant data savings. However, they are always connected to power and mostly connected to WiFi networks. Therefore, privacy is an essential concern, and such devices can use constant bit rate Opus/CELT codec when connected to WiFi. On the other hand, earbuds or other battery-powered devices can use iSAC. Similarly, an outdoor noise sensing device with microphone [48] can understand the noise intensity by analyzing the bitrate from iSAC, as it reacts to noise more aggressively.

7 RELATED WORK

Analysis of VoIP Applications. Baset et al. [19] studied key functionalities of Skype in terms of login, NAT traversal, and call establishment. Skype has a P2P network architecture with super nodes used to manage user logins. Ordinary hosts are the applications to place the calls and send messages. These hosts rely on various STUN protocols to find the NAT and firewalls. WhatsApp [42] application has a client-server architecture. Identifying Skype traffic has been an active research area. Bonfiglio et al. [20] proposed two methods to identify Skype voice call traffic from a collection of various packet types. They first looked into the statistical properties of message content and then matched with the Skype voice traffic sources by using Naive Bayesian techniques.

[4] http://www.techplayon.com/2286-2/

VoIP Flow Properties. Baset et al. [19] investigated Skype packet size and bitrates as well. In addition to packet size, Do and Branch used inter-packet gaps to classify VoIP traffic real time [27]. In our study, we have shown that different applications have different flow features, and such features may also vary according to the context. A recent study has shows how the traffic capturing tools may affect the measurement of these flow properties [38]. Also, the privacy implications of VoIP traffic pattern have received some attention [43, 67, 72]. Wu et al. [74] assumed that VoIP applications suppress silence. Instead of relying on the inter-packet gap or packet size, they proposed a machine learning algorithm to detect voice activity in the flow. As we have shown, in the presence of intense noise, the suppression may not be effective for some applications, and the proposed method may not work. Suh et al. [66] detected VoIP traffic from a Skype Relay node using Skype specific heuristic, as the relayed VoIP traffic from different clients form bursts.

VoIP Performance. Most previous works on VoIP performance have focused on metrics affecting the end-user. For example, Chen et al. [23] quantified the impact of bitrate, jitter, and loss, and delay on QoE. They further investigated the playout buffer management algorithms of three VoIP applications in [75]. Using this approach, the authors identified the inefficiency of the buffer management algorithm of the applications and suggested to use a modified algorithm to maintain optimal user satisfaction. Andersson et al. [16] studied the impact of various VoIP codecs on QoE in the LTE networks through simulation. In this article, we have investigated the flow features for different applications and found that a combination of smaller packet sizes and the higher inter-packet gap may influence the QoE negatively. VoIP applications also may set DSCP IP flags for different QoS guarantees from the network [37]. Recently, Skype performance was studied in a high-speed train context [45]. Dasari *et al.* measured the energy consumption of Skype video calls with different mobile devices [25]. Finally, Rämo and Toukomaa characterized the voice quality of various speech/voice codecs with subjective tests [58].

Overhead in Wireless Communication Traffic overhead with the encrypted traffic comes from the TLS handshake, and there is additional energy cost due to encryption. Naylor *et al.* [54] investigated the performance of HTTPS from traffic traces. The potential of data waste with multimedia streaming applications is quite high due to unnecessary content download [40, 46]. Similarly to the silence packets, TCP-based multimedia applications also exchange only protocol messages in the case of active flow control triggered by the streaming applications [39]. Sieber *et al.* [62, 63] quantified YouTube's traffic pattern with static and dynamic network conditions and found that YouTube may download 33% redundant traffic under dynamic network conditions. Bartendr [61] and eSchedule [40] also optimize energy consumption by pre-fetching content. However, these energy-saving approaches do not apply to the VoIP applications, as the traffic is real time and bi-directional.

8 CONCLUSIONS

Motivated by the increased and diverse usage of voice over IP applications, this measurement study reveals the data and energy cost offered by the silent suppression techniques of the underlying codecs. An efficient codec, with silence suppression, can be 50-200% more energy efficient compared to the other alternatives. However, such effectiveness is sensitive to the background noise. The performance also vary across handsets or operating systems. Interestingly, silence suppression makes VoIP traffic network friendly as much as VoLTE traffic. Conversely, our results also highlighted that silence suppression might reveal conversational patterns through network flow properties. Therefore, VoIP applications or devices can benefit significantly by selecting codecs based on the conversation context, data savings, energy consumption, and privacy requirements. Finally, our acoustic measurement setup provides a reproducible environment for testing, debugging, and evaluating different speech codecs. The use of silence and muted conversations provides a bound for the true performance of the codec, whereas recorded speech with added background noise allows investigating different conversation contexts.

ACKNOWLEDGEMENTS

We thank our shepherd Dr. Emir Halepovic and the anonymous reviewers for the valuable feedback, which helped to improve the paper. The research is supported by the Academy of Finland projects grant no 1319017, 319017, 5GEAR, FIT project, and project 16214817 from the Research Grants Council of Hong Kong.

REFERENCES

[1] Viber Turns to SPIRIT for High Quality HD Mobile VoIP Calling. https://www.spiritdsp.com/news/140-viber-turns-to-spirit-for-high-quality-hd-mobile-voip-calling/, 2011. Last Accessed: 11.12.2019.
[2] WebRTC - Frequent Questions. https://webrtc.org/faq/, 2011. Last Accessed: 12.07.2019.
[3] Duo. https://play.google.com/store/apps/details?id=com.google.android.apps.tachyon, 2018.
[4] Google Nest and Google Home device specifications. https://support.google.com/googlenest/answer/7072284?hl=en, 2018. Last Accessed: 28.08.2019.
[5] Messenger. https://play.google.com/store/apps/details?id=com.facebook.orca, 2018.
[6] Skype. https://play.google.com/store/apps/details?id=com.skype.raider, 2018.
[7] Viber. https://play.google.com/store/apps/details?id=com.viber.voip, 2018.
[8] WhatsApp. https://play.google.com/store/apps/details?id=com.whatsapp, 2018.
[9] Alexa Is Listening All The Time: Here is How To Stop It. https://www.forbes.com/sites/tjmccue/2019/04/19/alexa-is-listening-all-the-time-heres-how-to-stop-it, 2019. Last Accessed: 3.08.2019.
[10] AMR-WB/G.722.2. http://www.voiceage.com/AMR-WB.G.722.2.html, 2019. Last Accessed: 13.10.2019.
[11] Decibel X: dB, dBA Noise Meter. https://apps.apple.com/us/app/decibel-x-db-dba-noise-meter/id448155923, 2019.
[12] Measuring Device Power. https://source.android.com/devices/tech/power/device, 2019.
[13] Network Signal Guru. https://play.google.com/store/apps/details?id=com.qtrun.QuickTest, 2019.
[14] Recording a Packet Trace. https://developer.apple.com/documentation/network/recording_a_packet_trace, 2019. Last Accessed: 23.12.2019.
[15] SBC: Audio Codecs Supported (Transcoding and Pass-Through). https://support.sonus.net/display/SBXDOC61/Audio+Codecs, 2019. Last Accessed: 31.10.2019.
[16] K. Andersson, S. A. M. Mostafa, and R. Ui-Islam. Mobile VoIP user experience in LTE, year=2011. In *2011 IEEE 36th Conference on Local Computer Networks*, pages 785–788, Oct.
[17] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *ArXiv*, abs/1708.05044, 2017.
[18] N. Banovic, T. Buzali, F. Chevalier, J. Mankoff, and A. K. Dey. Modeling and Understanding Human Routine Behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 248–260, New York, NY, USA, 2016. ACM.
[19] S. Baset and H. Schulzrinne. An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *INFOCOM*. IEEE, 2006.
[20] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. Revealing Skype Traffic: When Randomness Plays with You. *SIGCOMM Comput. Commun. Rev.*,

37(4):37–48, Aug. 2007.
[21] R. Browne. Microsoft's Skype gets a redesign, ditching Snapchat-like feature 'Highlights'. https://www.cnbc.com/2018/09/03/microsoft-owned-skype-redesign-ditches-snapchat-like-highlights.html, 2018. Last Accessed: 26.03.2020.
[22] L. Caviglione. A first look at traffic patterns of Siri. *Transactions on Emerging Telecommunications Technologies*, 26:n/a–n/a, 08 2013.
[23] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. Quantifying Skype User Satisfaction. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 399–410, New York, NY, USA, 2006. ACM.
[24] K. Darras, P. Pütz, Fahrurrozi, K. Rembold, and T. Tscharntke. Measuring sound detection spaces for acoustic animal sampling and monitoring. *Biological Conservation*, 201:29 – 37, 2016.
[25] M. Dasari, S. Vargas, A. Bhattacharya, A. Balasubramanian, S. R. Das, and M. Ferdman. Impact of Device Performance on Mobile Internet QoE. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 1–7, New York, NY, USA, 2018. ACM.
[26] M. Developers. Codecs used by webrtc. https://developer.mozilla.org/en-US/docs/Web/Media/Formats/WebRTC_codecs, 2019. Last Accessed: 27.08.2019.
[27] L. H. Do and P. Branch. Real Time VoIP Traffic Classification. Technical report, Swinburne University of Technology, Melbourne, Australia, 2009.
[28] A. Elnashar, M. A. El-Saidny, and M. Yehia. Performance evaluation of volte based on field measurement data. *ArXiv*, abs/1810.02968, 2018.
[29] D. Florencio and L. He. Enhanced adaptive playout scheduling and loss concealment techniques for voice over ip networks. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pages 129–132, May 2011.
[30] G. Gokul, Y. Yan, K. Dantu, S. Y. Ko, and L. Ziarek. Real Time Sound Processing on Android. In *Proceedings of the 14th International Workshop on Java Technologies for Real-Time and Embedded Systems*, JTRES '16, pages 3:1–3:10, New York, NY, USA, 2016. ACM.
[31] A. Grinsted. Moving averages / Moving median etc. https://www.mathworks.com/matlabcentral/fileexchange/8251-moving-averages-moving-median-etc. Retrieved January 1, 2020.
[32] E. Halepovic, M. Ghaderi, and C. Williamson. Multimedia application performance on a wimax network. *Proceedings of SPIE - The International Society for Optical Engineering*, 7253, 01 2009.
[33] P. Hancke. Messenger exposed: Investigative report. *https://webrtchacks.com/wp-content/uploads/2015/05/messenger-report.pdf*, pages 01–15, 2015. Last Accessed: 12.11.2019.
[34] P. Hancke. Whatsapp exposed: Investigative report. *https://webrtchacks.com/wp-content/uploads/2015/04/WhatsappReport.pdf*, pages 01–16, 2015. Last Accessed: 13.10.2019.
[35] M. Heldner and J. Edlund. Pauses, gaps and overlaps in conversations. *J. Phonetics*, 38:555–568, 2010.
[36] C. Holmberg, S. Hakansson, and G. Eriksson. Web real-time communication use cases and requirements. *Request for Comments (RFC)*, 7478, 2015.
[37] M. A. Hoque, H. Abbas, T. Li, Y. Li, P. Hui, and S. Tarkoma. Barriers in seamless qos for mobile applications. 2018. arXiv:1809.00659.
[38] M. A. Hoque, A. Rao, and S. Tarkoma. In situ network and application performance measurement on android devices and the imperfections, 2020. arXiv:2003.05208.
[39] M. A. Hoque, M. Siekkinen, and J. K. Nurminen. TCP Receive Buffer Aware Wireless Multimedia Streaming: An Energy Efficient Approach. In *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '13, pages 13–18, New York, NY, USA, 2013. ACM.
[40] M. A. Hoque, M. Siekkinen, and J. K. Nurminen. Using crowd-sourced viewing statistics to save energy in wireless video streaming. In *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*, MobiCom '13, pages 377–388, New York, NY, USA, 2013. ACM.
[41] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, MobiSys '12, pages 225–238, New York, NY, USA, 2012. ACM.
[42] F. Karpisek, I. Baggili, and F. Breitinger. WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages. *Digital Investigation*, 15:110 – 118, 2015. Special Issue: Big Data and Intelligent Data Analysis.
[43] L. Khan, M. Baig, and A. M. Youssef. Speaker recognition from encrypted voip communications. *Digital Investigation*, 7(1):65 – 73, 2010.
[44] S. C. Levinson and F. Torreira. Timing in turn-taking and its implications for processing models of language. In *Front. Psychol.*, 2015.
[45] L. Li, K. Xu, D. Wang, C. Peng, K. Zheng, H. Wang, R. Mijumbi, and Xiangxiang Wang. A measurement study on Skype voice and video calls in LTE networks on high speed rails. In *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, pages 1–10, June 2017.
[46] X. Li, M. Dong, Z. Ma, and F. Fernandes. GreenTube: Power Optimization for Mobile Video Streaming via Dynamic Cache Management. In *Proceedings of the ACM Multimedia*, acmmm'12, New York, NY, USA, 2012. ACM.
[47] B. Locher, A. Piquerez, M. Habermacher, M. S. Ragettli, M. Röösli, M. Brink, C. Cajochen, D. Vienneau, M. A. Foraster, U. Müller, and J. M. Wunderli. Differences between outdoor and indoor sound levels for open, tilted, and closed windows. In *International journal of environmental research and public health*, 2018.
[48] P. Maijala, Z. Shuyang, T. Heittola, and T. Virtanen. Environmental noise monitoring using source classification in sensors. *Applied Acoustics*, 129:258 – 267, 2018.
[49] R. Marik, P. Bezpalec, J. Kucerak, and L. Kencl. Revealing viber communication patterns to assess protocol vulnerability. In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pages 496–504, Dec 2015.
[50] T. P. McAlexander, R. R. M. Gershon, and R. L. Neitzel. Street-level noise in an urban setting: assessment and contribution to personal exposure. In *Environmental Health*, 2015.
[51] M. Menth, A. Binzenhöfer, and S. Mühleck. Source models for speech traffic revisited. *IEEE/ACM Trans. Netw.*, 17(4):1042–1051, Aug. 2009.
[52] M. Mohseni, S. Banani, A. Eckford, and R. Adve. Scheduling for volte: Resource allocation optimization and low-complexity algorithms. *IEEE Transactions on Wireless Communications*, 18:1534 – 1547, 01 2019.
[53] V. Namboodiri and L. Gao. Towards energy efficient voip over wireless lans. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 169–178. ACM, 2008.
[54] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste. The cost of the "s" in https. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 133–140, New York, NY, USA, 2014. ACM.
[55] N. Pathania, R. Singh, isha, and A. Malik. Comparative Study of Audio and Video Chat Application Over the Internet. In *2018 International Conference on Intelligent Circuits and Systems (ICICS)*, pages 251–257, April 2018.
[56] T. Pulkkinen, J. Nurminen, and P. Nurmi. Understanding WiFi Cross-Technology Interference Detection in the Real World. In *Proceedings of the 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020.
[57] A. J. Pyles, Z. Ren, G. Zhou, and X. Liu. Sifi: exploiting VoIP silence for WiFi energy savings in smart phones. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 325–334. ACM, 2011.
[58] A. Rämö and H. Toukomaa. Voice quality characterization of ietf opus codec. In *INTERSPEECH*, 2011.
[59] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: Detecting non-wifi rf devices using commodity wifi hardware. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, page 137–154, New York, NY, USA, 2011. Association for Computing Machinery.
[60] H. Sacks, E. A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier, 1978.
[61] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 85–96, New York, NY, USA, 2010. ACM.
[62] C. Sieber, A. Blenk, M. Hinteregger, and W. Kellerer. The cost of aggressive http adaptive streaming: Quantifying youtube's redundant traffic. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1261–1267, May 2015.
[63] C. Sieber, P. Heegaard, T. Hoßfeld, and W. Kellerer. Sacrificing efficiency for quality of experience: YouTube's redundant traffic behavior. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 503–511, May 2016.
[64] Statistia. Most popular global mobile messenger apps as of October 2019, based on number of monthly active users. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, 2019. Last Accessed: 25.03.2020.
[65] F. Statistics. Mind-Blowing Viber Statistics. https://99firms.com/blog/viber-statistics/, 2020. Last Accessed: 26.03.2020.
[66] K. Suh, D. R. Figueiredo, J. Kurose, and D. Towsley. Characterizing and Detecting Skype-Relayed Traffic. pages 1–12, April 2006.
[67] T. Vaidya, T. Walsh, and M. Sherr. Whisper: A unilateral defense against voip traffic re-identification attacks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, pages 286–296, New York, NY, USA, 2019. ACM.
[68] J. Valin, K. Vos, T. Terriberry, and A. Moizard. RFC 6716: Definition of the Opus audio codec. *Internet engineering task force (IETF) standard*, 2012.
[69] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose. Phonotactic reconstruction of encrypted voip conversations: Hookt on fon-iks. In *2011 IEEE Symposium on Security and Privacy*, pages 3–18, May 2011.
[70] WHO. Common noise guideline values. *https://www.who.int/docstore/peh/noise/Comnoise-4.pdf*, pages 55–65, 2013. Last Accessed: 10.10.2019.
[71] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 35–49, May 2008.
[72] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In *Proceedings of*

16th USENIX Security Symposium on USENIX Security Symposium, SS'07, pages 4:1–4:12, Berkeley, CA, USA, 2007. USENIX Association.

[73] C. V. Wright, S. E. Coull, and F. Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *In Proceedings of the 16th Network and Distributed Security Symposium*, pages 237–250. IEEE, 2009.

[74] C.-C. Wu, K.-T. Chen, Y.-C. Chang, and C.-L. Lei. Detecting VoIP Traffic Based on Human Conversation Patterns. In H. Schulzrinne, R. State, and S. Niccolini, editors, *Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks*, pages 280–295, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[75] C.-C. Wu, K.-T. Chen, C.-Y. Huang, and C.-L. Lei. An Empirical Evaluation of VoIP Playout Buffer Dimensioning in Skype, Google Talk, and MSN Messenger. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '09, pages 97–102, New York, NY, USA, 2009. ACM.

[76] R. Zopf. RFC 3389: Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN). *Internet engineering task force (IETF) standard*, 2002.