Messages Behind the Sound: Real-Time Hidden Acoustic Signal Capture with Smartphones

Qian Wang[†]
qianwang@whu.edu.cn

Kui Ren[‡]
kuiren@buffalo.edu

Man Zhou[†], Tao Lei[†]
{zhouman,leitao}@whu.edu.cn

Dimitrios Koutsonikolas[‡]
dimitrio@buffalo.edu

Lu Su[‡]
lusu@buffalo.edu

[†]The State Key Lab of Software Engineering, School of Computer Science, Wuhan University, P. R. China
[‡]Dept. of Computer Science and Engineering, The State University of New York at Buffalo, USA

ABSTRACT

With the ever-increasing use of smart devices, recent research endeavors have led to unobtrusive screen-camera communication channel designs, which allow simultaneous screen viewing and hidden screen-camera communication. Such practices, albeit innovative and effective, require well-controlled alignment of camera and screen and obstacle-free access.

In this paper, we develop Dolphin, a novel form of real-time acoustics-based dual-channel communication, which uses a speaker and the microphones on off-the-shelf smartphones to achieve concurrent audible and hidden communication. By leveraging masking effects of the human auditory system and readily available audio signals in our daily lives, Dolphin ensures real-time unobtrusive speaker-microphone data communication without affecting the primary audio-hearing experience for human users, while, at the same time, it overcomes the main limitations of existing screen-camera links. Our Dolphin prototype, built using off-the-shelf smartphones, realizes real-time hidden communication, supports up to 8-meter signal capture distance and $\pm 90^\circ$ listening angle and achieves decoding rate above 80% without error correction. Further, it achieves average data rates of up to 500bps while keeping the decoding rate above 95% within a distance of 1m.

CCS Concepts

•Networks → Mobile networks; •Human-centered computing → Mobile devices;

Keywords

Speaker-microphone communication; hidden audible communication; dual-mode communication

MobiCom'16, October 03-07, 2016, New York City, NY, USA

DOI: http://dx.doi.org/10.1145/2973750.2973765

1. INTRODUCTION

With the ever-increasing popularity of smart devices in our daily lives, people more and more heavily rely on them to gather and spread a wide variety of information in the cyber-physical world. At the same time, various surrounding devices equipped with screens and speakers, e.g., stadium screens & sports displays, advertising electronic boards, TVs, desktop/tablet PCs, and laptops, have become a readily available information source for human users. As announced in Sandvine's semiannual "Global Internet Phenomena report" [1], video and audio streaming accounts for more than 70% of all broadband network traffic in North America during peak hours. Under this trend, it is highly expected that the screens and speakers convey vivid information through videos and audios to human users while further delivering other meaningful and customized content to smart devices held by human users. For example, a sports fan could be watching NBA live streams on the stadium screen, while receiving background information or statistics for each player and team on his/her smart device without resorting to the Internet. Another real-life example could be a person watching advertisements on TV while receiving instant notifications, offers, and promotions on his/her device.

In existing video-based applications, this side information is usually directly displayed on top of the video content or encoded into visual patterns and then shown on the screen. This practice inevitably causes resource contention, since the coded images on the screen (reserved for devices) interfere with the content the screen is displaying (reserved for users), leading to unpleasant and distracting viewing experience for human users. Recent research endeavors [22, 13, 20, 14] have tried to eliminate this tension between users and devices by developing techniques that allow the screen to concurrently display content to users and communicate side information to devices, finally enabling real-time unobtrusive screen-camera communication.

Such practices, albeit innovative and effective, still have practical limitations in real-world scenarios, mainly because they require a direct visible communication path between the screen content and the camera capture window. First, the well-controlled alignment of screen and camera undermines the flexibility of a dual-mode communication system. In most cases, users holding smart devices are moving around public spaces such as malls and cafes. While the user can still see the content displayed on the screen, the camera of the smart device cannot accurately capture the full screen

on target from a wide range of viewing angles, in addition to its sensitivity to device shaking. Second, screen-camera communication highly relies on the camera's line of sight (LOS). If there are obstacles or moving objects in between the screen and the camera, the device will fail to capture and decode any useful information from the screen content. Third, the communication/viewing distance is restricted by the size of the screen, which cannot be freely adjusted once deployed.

To avoid the practical limitations of unobtrusive screen-camera communication, we develop Dolphin, a novel form of real-time dual-channel communication over speaker-microphone links, which leverages sound signals instead of visible light. Dolphin generates composite audio for the speaker by multiplexing the original audio (intended for the human listener) and the embedded data signals (intended for smartphones). The composite audio can be rendered to human ears without affecting the content perception of the original audio. The user thus listens to the audio as usual without sensing the embedded data. In the meantime, the data signals carried by the composite audio can be captured and extracted by the smartphone microphones.

The inherent properties of audio signals overcome several of the limitations of unobtrusive screen-camera communication systems. First, the sound travels to all directions and thus makes the signal receiving angle broader compared to the highly directional visible light beams. Second, the sound can be transmitted by diffraction and reflection even with some small obstacles while visible light is easy to be blocked. Third, the fact that acoustic frequencies are easy to be separated on off-the-shelf smartphones (as opposed to visible light frequencies which require special hardware) motivates us to adopt OFDM to increase the throughput of speaker-microphone communication. The fixed screen size limits the flexibility of screen-camera communication. For example, the camera needs to focus on the full screen steadily during communication, while the speaker volume can be adjusted to control the speaker-microphone communication distance and a small device motion is allowed.

The design of Dolphin addresses three major challenges. First, there is an inherent tradeoff between audio quality and signal robustness. While a stronger embedded signal can resist the speaker-microphone channel interference, it may not be unobtrusive to the human ear. To seek the best tradeoff, we propose an adaptive signal embedding approach, which chooses the modulation method and the embedded signal strength adaptively based on the energy characteristics of the carrier audio. Second, the speaker-microphone links suffer from serious distortion caused by both the acoustic channel (e.g., ambient noise, multipath interference, device mobility, etc.) and the smartphone hardware limitations (e.g., the frequency selectivity of the microphone). To combat ambient noise and multi-path interference, we adopt OFDM for the embedded signal and determine the system parameters according to the characteristics of speaker-microphone links. We further adopt channel estimation based on a hybrid-type pilot arrangement to minimize the impact of frequency-time selective fading and Doppler frequency offset. Third, various practical environments result in different levels of bit error rates. To enhance the transmission reliability, we design a Bi-level orthogonal error correction (OEC) scheme according to the bit error distribution.

We built a Dolphin prototype using an HiVi M200MKIII loudspeaker as the sender and different smartphone platforms as receivers, and evaluated user perception, data communication performance and other practical considerations. Our results show that Dolphin is able to achieve throughput up to 500bps averaged over various audio contents while keeping the decoding rate above 95% within a distance of 1m. Our prototype supports a range of up to 8 meters and a listening angle of up to $\pm 90^\circ$ (given the reference point facing the speaker) and achieves a decoding rate above 80% without error correction, when the speaker volume is 80dB. Finally, Dolphin realizes real-time hidden data transmission with average symbol encoding time 1.1ms and average symbol decoding time $24.6 \sim 36.6$ms on different smartphones. The main contributions of this work are summarized as follows.

- We propose Dolphin, a novel form of real-time unobtrusive speaker-microphone hidden communication, which allows information data streams to be embedded into audio signals and transmitted to smartphones while remaining unobtrusive to the human ear.
- We propose an adaptive embedding approach based on OFDM and energy analysis of the carrier audio signal, which makes the embedded information over various types of audio unobtrusive. To enhance Dolphin's robustness and reliability, we leverage pilot-based channel estimation during signal extraction and design a novel orthogonal error correction (OEC) mechanism to correct small data decoding errors. The result is a flexible and lightweight design that supports both real-time and offline decodings.
- We build a Dolphin prototype using off-the-shelf smartphones and demonstrate that it is possible to enable flexible data transmissions in real-time unobtrusively atop arbitrary audio content. Our results show that Dolphin overcomes several of the limitations of VLC-based unobtrusive screen-camera communication systems and can be adopted as a complementary or joint dual-mode communication strategy along with such systems to enhance the data transmission rate and reliability under various practical settings.

2. BACKGROUND

In this section, we present some related basic properties of the human auditory system [32], the speaker, and the smartphone microphone, which provide us with the theoretical basis for the design of Dolphin.

2.1 Human Auditory System

Human ear is the core instrument in the human auditory system, which reacts to sounds and obtains the perception of loudness, pitch, and semantics. We mainly describe it from two aspects: the perception of loudness and pitch, and the masking effects.

Perception of loudness and pitch: Loudness indicates the strength of sounds. But the subjective feeling of loudness might differ from the physical measurement of sound strength. The sensitivity of human ear to the sounds of different frequencies is different. Human ear is most sensitive to the sounds in $2 \sim 4$KHz [27]. A human can hear a sound even if the physical sound strength is very low, but the physical sound strength needs to be much higher to be

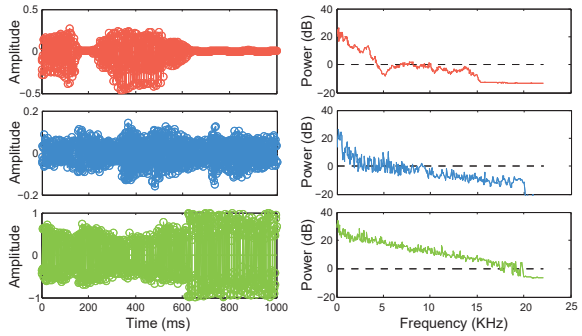


Figure 1: The time-domain plot and frequency spectrum of human voice, soft music, and rock music.

perceived by humans if the sound resides in higher frequency bands. The pitch is indicated by the frequency (Hz), and the human hearing frequency range of sounds is between 20 $\sim$ 18000Hz [27].

Masking effects: "Auditory masking effects" refers to the phenomenon that a sound in a given frequency (masking sound) hinders the perception of the human auditory system to a sound in another frequency (masked sound). The masking effect depends on the amplitude and time-frequency domain features of the two sounds, and includes frequency masking and temporal masking [19]. Frequency masking means that the stronger sound will shadow the weaker sound if the frequencies of two sounds are very close. Due to the different subjective perception to sounds in different frequencies, the lower frequency sound can effectively mask the higher frequency sound, but not vice-versa. Temporal masking means that the stronger signal will flood the weaker signal if the two sounds appear almost at the same time.

2.2 Speaker and Smartphone Microphone

The response frequency of most speakers and microphones is from 50 to 20000Hz. The speaker is a transducer that converts electrical signals into acoustic signals. But different speakers have different levels of frequency selectivity, and their performance degenerates significantly at higher frequencies. The microphone is also a transducer which converts acoustic signals into electrical signals. Limited by its size, a smartphone microphone is simple and has limited capabilities. Similar to speakers, microphones exhibit frequency selectivity. Most people almost cannot hear sounds with frequencies higher than 18KHz. However, the performance of speakers and microphones also degenerates significantly at higher frequency bands. Therefore, realizing a second acoustic channel unobtrusive to the human ear over the speaker-microphone link is not a trivial task.

3. THE ACOUSTIC SPEAKER-MICROPHONE CHANNEL

The challenges for realizing Dolphin lie in both the limitations of off-the-shelf smartphones and the characteristic of aerial acoustic communication. The design challenges due to the nature of the acoustic signal propagation and speaker-microphone characteristics include *tradeoff between audio quality and signal robustness*, *speaker-microphone frequency selectivity*, *ringing and rise time*, *phase and frequency shift*, *ambient noise*, *multipath interference*, *propagation loss* and *limited coding capacity of audio*. The successful operation of Dolphin highly depends on the characteristics of the acoustic speaker-microphone channel. Therefore, we conduct extensive experiments to understand its characteristics.

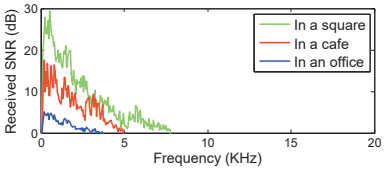


Figure 2: Spectrum of Ambient Noise.

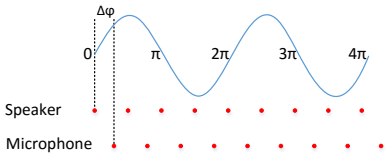


Figure 3: The red dots indicate the sampling points, and $\Delta\varphi$ indicates phase shift.

3.1 Audio Time-Frequency Characteristics

Figure 1 shows the time and frequency characteristics of three types of audio (human voice, soft music and rock music). It is obvious that different types of audio exhibit different features in both the time and the frequency domains. For example, the human voice is intermittent in the time domain due to speech pauses. The energy of soft music and human voice is focused in the 0$\sim$5KHz band. In contrast, the energy of rock music is distributed in a much wider frequency band (0$\sim$15KHz). Therefore, in order to correctly decode the embedded information without affecting the original audio, we must take the time-frequency characteristics into consideration when we design the composite audio.

3.2 Ambient Noise

Ambient noise in public spaces can cause significant interference on acoustic signals over the speaker-microphone link, resulting in low decoding rate for the embedded information. To characterize this interference, we measured the power of ambient noise in different environments. As an example, Figure 2 shows the energy distribution of ambient noise measured on a SAMSUNG GALAXY Note4 smartphone in a square and a cafe during busy hours. The ambient noise in the two locations (especially in the square) is relatively high at frequencies lower than 2KHz, but, similar to the observation in [16], it becomes negligible (i.e., close to noise levels) at frequencies higher than 8KHz. Hence, we can use a frequency higher than 8KHz to minimize the interference caused by ambient noise.

3.3 Frequency Shift

Wireless communication usually suffers from Doppler frequency shift due to mobility. The shift is more prominent for acoustic communication since the speed of the sound is relatively low. Let ν denote the speed of sound in the air, f_s denote the frequency of the signal carrier, and θ denote the angle between the moving direction of the smartphone and the speaker. When the smartphone moves from left to right with speed ν_0, the Doppler frequency shift Δf is calculated

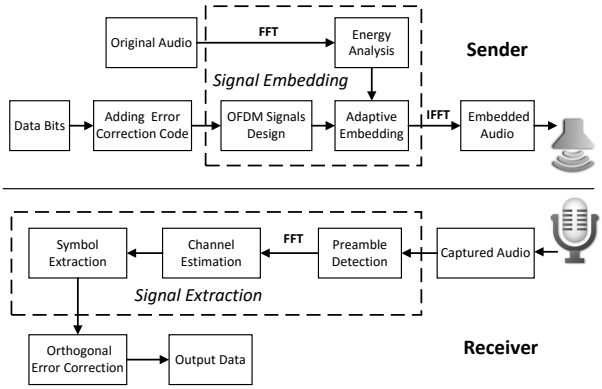


Figure 4: System architecture of Dolphin.

as

$$\Delta f = \frac{\nu_0 \cos\theta}{\nu} f_s. \quad (1)$$

From Equation 1, given that the speed of sound in the air is 340m/s, and the walking speed is about 1.5m/s, Δf cannot be ignored, especially when f_s exceeds 10KHz. Further, note that the impact of a large Doppler frequency shift is higher in OFDM systems due to the limited bandwidth of each subcarrier.

3.4 Phase Shift

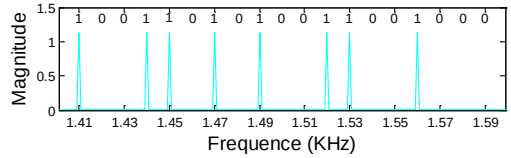
Phase shift commonly exists in wireless communications, and it is a more serious concern for off-the-shelf smartphones with low sampling rate. To our best knowledge, the maximum sampling rate of the speaker and the microphone in most off-the-shelf smartphones is 44.1KHz, which results in limited sampling points in a signal period. For example, there are only 4 sampling points in one period of a sine signal with frequency 10 KHz.

Note that the digital signals are converted into analog signals via a DAC in the speaker, and the received analog signals are converted into digital signals via an ADC in the microphone. As shown in Figure 3, one major source of the phase shift is that the sampling points at the DAC in the speaker will not be the same as those at the ADC in the microphone. In fact, the imperfect synchronization of the preamble (to be discussed in Section 4.3.1) makes phase shift more serious. For example, the phase shift of a 10 KHz sine signal is $\frac{\pi}{2}$ if the synchronization error is 1 sampling point. Typical preamble synchronization methods (e.g., [12]) result in synchronization errors within 5 sampling points. Therefore, the imperfect synchronization of the preamble makes phase shift unpredictable and the phase shift keying (PSK) technique unsuitable for Dolphin.

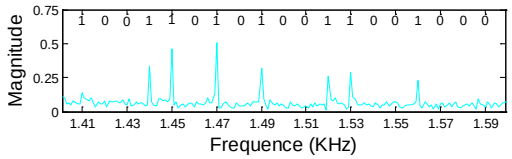
4. DOLPHIN DESIGN

4.1 System Overview

Figure 4 illustrates the system architecture of Dolphin which consists of two parts: the sender and the receiver (e.g., a TV and a smartphone, respectively). Roughly speaking, the sender embeds data (e.g., detailed description of products) into the original audio and transmits the composite audio through its speaker. The microphone on the user's smartphone captures the composite audio and decodes it to obtain the embedded data.



(a) The encoded ASK signal on the sender.



(b) The captured ASK signal on the receiver.

Figure 5: Amplitude shift keying signal.

The sender: Raw data bits are encapsulated into packets, and bits in each packet are encoded by orthogonal error correction (OEC) codes (Section 4.4), divided into symbols, and further modulated by OFDM. We analyze the original audio stream on the fly to locate the appropriate parts to carry the embedded information packets. First, we perform energy distribution analysis to select the subcarrier modulation method for each packet. Then, we perform energy analysis on every part of the audio corresponding to a symbol. If the energy of a part is enough to mask the embedded signals, we adaptively embed the symbol into it according to its energy characteristics. Otherwise, we do not make any modifications. Finally, the sender transmits the data-embedded audio via the speaker.

The receiver: After the audio is captured by the smartphone microphone, we first detect the preamble of each packet. Then we can segment accurately each part of the audio corresponding to a symbol. Signals typically suffer serious frequency-time selective fading over the speaker-microphone link. To improve the decoding rate, we perform channel estimation before symbol extraction. Finally, we convert the corresponding audio signals into symbols, extract the data bits in each symbol, and recover the original data after OEC.

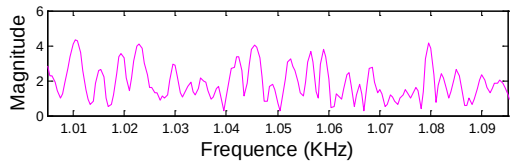
4.2 Signal Embedding

4.2.1 OFDM Signal Design

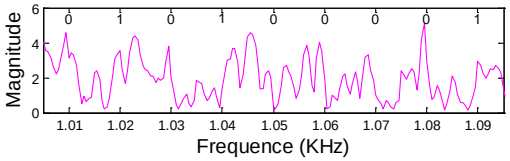
We adopt orthogonal frequency division multiplexing (OFDM) for the signal design of Dolphin for combating frequency-selective fading and multi-path interference. In this section, we describe the OFDM signal design based on the characteristics of the acoustic channel.

Choosing the operation bandwidth: Recall from Section 2.2 and Section 3.2 that the frequency response of most speakers and microphones is between 50∼20000 Hz, and the interference from the ambient noise is negligible when the frequency exceeds 8KHz. In addition, it has been shown that the bandwidth between 17 ∼ 20KHz consists of nearly inaudible frequencies [17], where a small amount of energy of the original audio can mask the embedded signals. Because this bandwidth is relative limited, we also propose to use the bandwidth below 17KHz to improve throughput. Finally, we choose 8∼20KHz as the frequency bandwidth for the embedded data.

Symbol sub-carrier modulation: As discussed in Sec-



(a) The original audio in frequency domain.



(b) The encoding EDK signals.

Figure 6: Energy difference keying signals.

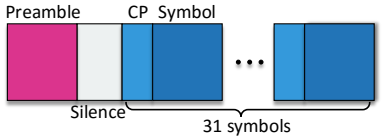


Figure 7: Dolphin packet format.

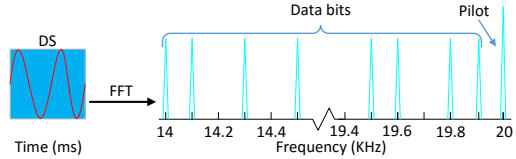


Figure 8: The data bits of an amplitude shift keying symbol.

tion 3.4, the unpredictable phase shifts due to the non-ideal synchronization of the preamble makes PSK unsuitable in Dolphin. Additionally, the limited subcarrier width in OFDM makes it hard to decode FSK-modulated signals. Hence, Dolphin uses ASK to modulate the signal on each subcarrier.

To ensure the embedded data stream is unobtrusive to the human ear, we cannot embed strong signals into a subcarrier. Thus, we use a special form of ASK, On-Off Keying (OOK). The embedded signals appear as peaks in the frequency domain, as shown in Figure 5(a). To decode the embedded data, the receiver must set a threshold to determine whether or not there are peaks on the subcarrier. However, selecting this threshold is challenging due to the speaker-microphone frequency selectivity and channel interference. As shown in Figure 5(b), peaks may be jagged or even erased. A drawback of ASK is that the energy distribution of the original audio in our embedding data bandwidth must be very low. Hence, we cut off the energy of the original audio in the embedding data bandwidth before embedding data bits. In order to make the changes unobtrusive as much as possible, we only embed data in 14∼20KHz in ASK, which means we need to cut off the energy of the original audio beyond 14KHz. If the energy distribution of the original audio is relatively high in the frequency range beyond 8KHz, we use a different modulation method called energy difference keying (EDK) instead of ASK.

EDK adjusts the energy distribution around the subcarrier central frequency to indicate 0 and 1 bits. For example, higher energy on the left of the subcarrier central frequency indicates 0, and higher energy on the right of the subcarrier central frequency indicates 1, as shown in Figure 6. Since the energy of original audio is usually low beyond 15KHz, we only embed data in 8∼14KHz in EDK. To deal with the speaker-microphone frequency selectivity and channel interference, the diversity of the energy on the left and right side of the central frequency must be sufficiently large. Thus, we adjust the energy in a frequency band B_{si} around the subcarrier central frequency rather than at some discrete frequencies. To guarantee the same level of robustness, the change of the energy distribution in the original audio with EDK is usually larger than with ASK. But in EDK, we do not need to cut off the energy of the original audio. In addition, since the frequencies in the left and right sub-carriers are very close, the energy adjustment is hard to be perceived. Hence, EDK is suitable in cases when the original audio has relatively high energy in high frequencies (e.g., rock music).

Dolphin packet format: For the convenience of data transmission and decoding, we divide the embedding data streams into packets. As shown in Figure 7, a packet consists of a preamble and 31 symbols, each preceded by a cyclic prefix (CP). The preamble is used to synchronize the packet, and the symbols contain data bits.

To synchronize the OFDM transmitter and receiver, a preamble precedes each transmitted packet. Following the approach of previous aerial acoustic communication systems (e.g., [16] and [12]), we use a chirp signal as the preamble. Its frequency ranges from f_{min} to f_{max} in the first half of the duration and then decreases back to f_{min} in the second half. In our implementation, we chose f_{max} = 19KHz and f_{min} = 17KHz, and the duration of preamble is 100ms. Due to its high energy, we pad each preamble with a silence period of 50ms to avoid interference to the data symbols.

The data bits in a symbol are embedded into a small piece of audio as a whole. As shown in Figure 8, when a symbol signal is converted from the time domain to the frequency domain, 60 subcarriers in the range 14∼19.9KHz are used to encode the data bits, and the signal in 20KHz is a pilot used for time-selective fading and Doppler frequency offset estimation. The pilot is very easy to be detected because it lies on the rightmost of the symbol spectrum. To estimate the frequency-selective fading, we set additional pilots on each subcarrier of the first symbol. A longer data symbol duration and less subcarriers increases the decoding rate but reduces throughput. In our experiments (Section 5.2), we found that a duration of 100ms and 60 subcarriers achieves a good tradeoff between robustness and throughput.

In RF OFDM radios, a cyclic prefix (CP) is designed to combat Inter Symbol Interference (ISI) and Inter-Carrier Interference (ICI). It copies a certain length from the end of the symbol signal in front of the symbol. Similarly, we adopt the cyclic prefix in acoustic OFDM to combat ISI and ICI. In our implementation, the CP duration is set to be 10ms.

4.2.2 *Energy Analysis*

We perform energy distribution analysis to select the subcarrier modulation method (ASK or EDK) for each packet. Let f (in KHz) denote the frequency, $F(f)$ denote the normalized signal magnitude at frequency f, l denote the number of sampling points in a packet, F_s denote the sampling rate, and $\Delta f_{(f_i,f_j)}$ denote the bandwidth of the frequency band $f \in [f_i, f_j]$, then the average energy spectrum density (ESD) of the audio corresponding to a packet E_{pt} is

calculated as

$$E_{pt} = \frac{l \cdot \sum_{f=0}^{20} |F(f)|^2}{2 \cdot F_s \cdot \Delta f_{(0,\ 20)}}. \quad (2)$$

The average energy spectrum density in the lower frequency band E_{pl} is calculated as

$$E_{pl} = \frac{l \cdot \sum_{f=0}^{8} |F(f)|^2}{2 \cdot F_s \cdot \Delta f_{(0,\ 8)}}. \quad (3)$$

Similarly, the average energy spectrum density in the higher frequency band E_{ph} is calculated as

$$E_{ph} = \frac{l \cdot \sum_{f=8}^{20} |F(f)|^2}{2 \cdot F_s \cdot \Delta f_{(8,\ 20)}} \quad (4)$$

The default modulation method is ASK. We choose EDK when the energy distribution satisfies the following two conditions, based on two thresholds E_{high} and R_{hl}:

$$E_{ph} > E_{high} \quad \text{and} \quad \frac{E_{ph}}{E_{pl}} > R_{hl}. \quad (5)$$

In our implementation, we empirically set E_{high}=10^{-7}J/Hz and R_{hl}=$\frac{1}{700}$. We embed a control signal at 19.6KHz into each preamble to indicate the selected modulation method to the receiver.

As shown in Figure 1, voice is intermittent in the time domain due to the speech pauses. In addition, the music volume often changes with time. If we embed a data symbol into a piece of low volume audio, it will be easily perceived by the user. To avoid this situation, we perform energy analysis on every piece of audio corresponding to a symbol. The calculation of the average ESD of a symbol is similar to that of a packet. We let E_{st}, E_{sl} and E_{sh} denote the ESD of the whole frequency band, the lower frequency band, and the higher frequency band, respectively. We embed symbol bits into a piece of audio only when the average energy of this audio piece E_{st} is higher than a threshold E_{min}, which measures the minimum audio energy spectrum density the data symbol needs. For better audio quality, E_{min} should be large. But a large E_{min} also means that fewer audio pieces can be used for data embedding. By our subjective perception experiments and energy statistics of audio pieces, we set $E_{min} = 2 \times 10^{-8}$J/Hz for the tradeoff. The receiver only needs to detect the pilot in 20KHz to know whether this piece of audio is embedded with data bits or not.

4.2.3 Adaptive Embedding

Due to the temporal masking effect of the human ear, a low noise can be perceived when the energy of the original audio is low, while the noise is often unobtrusive when the energy of the original audio is very high. Based on this feature, we increase the strength of embedded signals when the audio signal is noisy and decrease it when the audio signal is quiet. In other words, the energy of embedded signals is adapted to the average energy of a piece of audio corresponding to a symbol, according to the following rule:
1) For ASK, the embedded signal energy magnitude of a symbol is calculated as

$$E_{am} = \begin{cases} N \cdot \beta^2\ E_{sl} & E_{sl} < E_{max} \\ N \cdot \beta^2\ E_{max} & E_{sl} \geq E_{max} \end{cases} \quad (6)$$

2) For EDK, the embedded signal energy magnitude of a symbol is calculated as

$$E_{en} = \begin{cases} N \cdot \beta^2\ E_{sl}\ B_{si} & E_{sl} < E_{max} \\ N \cdot \beta^2\ E_{max}\ B_{si} & E_{sl} \geq E_{max}. \end{cases} \quad (7)$$

Here, N is the number of subcarrier, β is the embedding strength coefficient and B_{si} is the adjusting bandwidth in EDK. In our implementation, B_{si} is set to be 20Hz when the subcarrier bandwidth is 100Hz. E_{max} is a threshold to measure the maximum embedding signal energy spectrum density, set to 3×10^{-7}J/Hz. When the energy of the original audio further increases, the strength of embedded signals remains unchanged since the signal is robust enough. If we further increase the strength, the noise would be too large and it is easy to be perceived. As can be seen from Equations 6 and 7, the changes in the original audio in the case of EDK are usually larger than in the case of ASK. To facilitate channel estimation (Section 4.3.2), the signal energy of pilots at the sender must be known to the receiver. Thus, we fix the energy of pilots at the sender.

4.3 Signal Extraction

Embedded signal extraction on the receiver side after the audio is captured by the smartphone microphone includes three steps: preamble detection, channel estimation, and symbol extraction.

4.3.1 Preamble Detection

A preamble is used to locate the start of a packet. In addition, we detect the control signal at 19.6KHz of the preamble to obtain the modulation method of the symbol subcarrier (Section 4.2.2). We adopt envelope detection to detect the preamble chirp signals. Theoretically, the maximum envelope corresponds to the location of the preamble. But in practice, the envelopes around the location of the preamble are very close at the receiver due to the ringing and rise time [16], resulting in synchronization errors within 5 data sampling points in our preliminary experiments. Such synchronization errors will cause unpredictable phase shift (Section 3.4). In Dolphin, however, each symbol corresponds to 4410 data sampling points and hence, errors of up to 5 data sampling points have almost no effect on the amplitude and energy distribution of the subcarrier signals. This is the reason we adopt ASK and EDK instead of PSK.

4.3.2 Channel Estimation

After the preamble is detected and located, each symbol of a packet can also be separated accurately. As mentioned above, frequency selectivity estimation (FSE), time selectivity estimation (TSE), and Doppler frequency offset elimination (DFOE) are required before symbol extraction. In Dolphin, we adopt a channel estimation technique based on pilot arrangement [5].

Choosing the type of pilot: The block-type pilot and the comb-type pilot schemes [5] are presented in Figures 9(a) and (b). Block-type pilot channel estimation is performed by sending pilots at every subcarrier; the estimation is used for a specific number of following symbols. It is effective in estimating the frequency-selective fading channel under the assumption that the channel transfer function is not changing very rapidly. Comb-type pilot channel estimation inserts pilots at a specific subcarrier of each symbol. It is

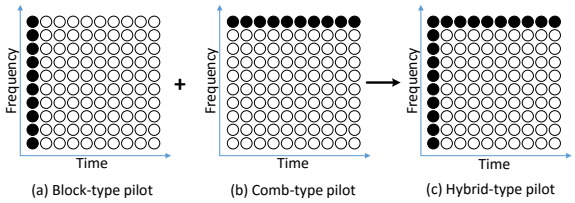


Figure 9: Hybrid-type pilot scheme. The black dots are the pilots, and the white dots are the data bits.

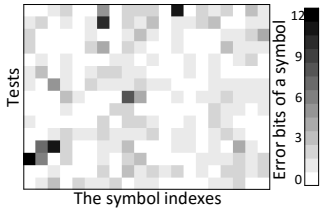


Figure 10: The error distribution of a packet under repeated tests.

effective in estimating the time-selective fading and Doppler frequency offset of each symbol and thus suitable for time-varying channels. Considering the high speaker-microphone frequency selectivity and large Doppler frequency offsets caused by mobility, we adopt a hybrid-type pilot arrangement, as shown in Figure 9(c). As mentioned in Section 4.2.1, we set pilots on each subcarrier of the first symbol in a packet to estimate the frequency-selective fading and additional pilots at 20KHz of each symbol to estimate the Doppler frequency offset and time-selective fading of each symbol.

Estimating channel transform function: We first discuss how to estimate the frequency-selective fading function (FSE) via the pilots on the first symbol of each packet. Usually, Least Square Estimation (LSE) or Minimum Mean-Square Estimation (MMSE) are used to calculate channel impulse response. MMSE performs better than LSE, but it is more complex and requires more computation resources. For real-time signal extraction, we adopt LSE in Dolphin. After removing the cyclic prefix, without taking into account ISI and ICI, the received signals in the first symbol can be expressed as

$$y(n) = x(n) \otimes h(n) + w(n) \quad n = 0, 1, \ldots, N-1, \tag{8}$$

where $w(n)$ denotes the ambient noise, $h(n)$ is the channel impulse response, and N is the number of sampling points in a symbol. We convert $y(n)$ from the time domain to the frequency domain via FFT as

$$Y(k) = X(k) * H(k) + W(k) \quad k = 0, 1, \ldots, N-1. \tag{9}$$

Let $Y_p(k)$ denote the pilot signals we extract from $Y(k)$ and $X_p(k)$ denote the known pilot signals added at the sender side. The estimated channel impulse response $H_e(k)$ can be computed as

$$H_e(k) = \frac{Y_p(k)}{X_p(k)} = H_p(k) + \frac{W_p(k)}{X_p(k)}, \tag{10}$$

where $H_p(k)$ denotes the channel impulse response of pilot signals, $W_p(k)$ is the ambient noise of pilot signals, and $\frac{W_p(k)}{X_p(k)}$ is the estimation error. Since we only encode signals at frequencies higher than 8KHz (Section 4.2.1), the ambient noise has almost no effect (Section 3.2), resulting in very small estimation error. In fact, the frequency selectivity is mainly due to the electro-mechanical components in the microphone/speaker rather than due to multipath [16]. Hence, the frequency-selective fading of the symbols following the first symbol is very similar to $H_p(k)$.

Next we discuss how to estimate the time-selective fading function (TSE) and Doppler frequency offset (DFOE) via the pilots on 20KHz subcarrier of each symbol. We use again LSE. Note that when the receiver is moving, the amplitude and phase of the channel response within one symbol will change due to the Doppler frequency offset. To compensate for the estimation error, we also need to take mobility into account. The pilot frequency f_s of transmitted signals is known (at 20KHz), and we can detect the pilots of received signals to obtain their frequencies f_r. Then we can calculate the Doppler frequency shift determinant $\nu_0 \cos\theta$ as

$$\nu_0 \cos\theta = \frac{(f_r - f_s)\nu}{f_s}. \tag{11}$$

We further calculate the frequency shift of all subcarriers in each symbol by Equation 1. After frequency offset elimination, all data signals are accurately located.

4.3.3 Symbol Extraction

After DFOE, each subcarrier's embedded data is accurately located, and we use channel estimation to recover the original signals. We define a "data window" whose length is equal to the subcarrier bandwidth. The data window intercepts the data whose center frequency is the first subcarrier frequency. We demodulate the signals according to the modulation method used for the subcarrier. Then the data window moves forward at a step of one subcarrier bandwidth until the embedded bits of all subcarriers are extracted. Note that the power of the embedded signals is adaptive based on the average energy of a piece of audio corresponding to a symbol. Hence, we adjust the decision threshold of each symbol according to its average energy.

4.4 Error Correction

In this section, we first analyze the error distribution characteristics and then introduce orthogonal error correction (OEC) to enhance data reliability.

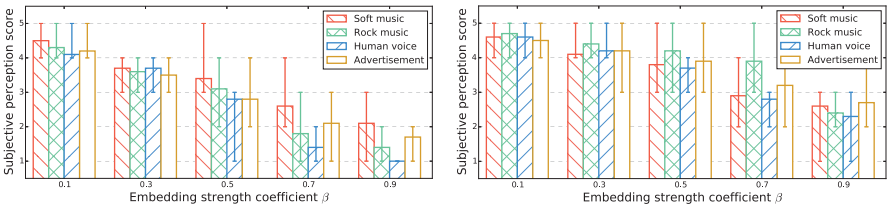
4.4.1 Analysis of Data Errors

We repeatedly test the error distribution of a packet under the same conditions (as described in our experimental settings), as shown in Figure 10. In each test, it is easy to see that most symbols have errors, but the number of error bits are typically no more than 3. The error distribution of a symbol in the frequency domain is random, and it may be caused by noise rather than the speaker-microphone frequency selectivity. Therefore, only a small error correction redundancy in the symbols can often correct all the errors. In some cases, the number of error bits in a symbol may exceed 10, probably due to high multipath interference. In those cases, we have to use excessive coding in the symbol to guarantee reliability.

4.4.2 Orthogonal Error Correction

According to the characteristics of the data errors, we design an orthogonal error correction (OEC) scheme. Our

Figure 11: Implementation of Dolphin on the smartphone.



(a) Static embedding (b) Adaptive embedding

Figure 12: Adaptive embedding improvement on subjective perception.

OEC scheme includes intra-symbol error correction and inter-symbol erasure correction in two orthogonal dimensions: time and space.

Intra-symbol error correction: Inside a symbol, we focus on errors caused by noise. In our implementation, we use the Reed-Solomon (RS) codes [25]. Based on a finite field with 15 elements (1 element represents 4 bits), an RS$(n;k)$ code has the ability to correct up to $\lfloor (n-k)/2 \rfloor$ error elements and to detect any combinations of up to $n-k$ error elements. In order to improve the error detection ability, before encoding into an RS code, the last element of the original message is set to be the XOR of all other elements in it. The receiver calculates the XOR to verify the correctness after the RS coded data has been successfully decoded.

Inter-symbol erasure correction: Inter-symbol erasure correction aims to correct the large number of errors in very few symbols, which cannot be corrected by the RS code. The symbols in a packet are denoted as $cell(i)$ $(i \in [1,30])$, and $cell(i)(j)$ denotes the bit in the jth subcarrier $(j \in [1,60])$. After running intra-symbol error correction, we know which symbols are unreliable. Now, we need to recover each of them by using other reliable symbols in the packet. Our idea is that the last m symbols in a packet are used as the parity-check symbols. We set $s = \lfloor (30+i)/m \rfloor - 1$ $(i \in [0,m))$ for each $j \in [1,60]$,

$$cell(30-i)(j) = \bigoplus_{k=1}^{s} cell(km-i)(j). \qquad (12)$$

As long as only one symbol has serious errors in s relevant symbols, the error symbol can be recovered by $s-1$ other symbols.

5. IMPLEMENTATION AND EVALUATION

We implement a prototype of Dolphin using commodity hardware. The sender is implemented on a PC equipped with a loudspeaker and the receiver is implemented as an Android app on different smartphone platforms. The app interface on GALAXY Note4 is shown in Figure 11. The sender takes an original audio stream and a data bitstream (generated with a pseudo-random data generator with a preset seed) as its input, generates the multiplexed stream, and then plays back the audio stream on the loudspeaker in real time. The receiver captures the audio stream, detects the preamble of each packet, conducts channel estimation, and extracts the embedded data in each symbol, also in real time.

Experimental Settings: We use a DELL Inspiron 3647 with 2.9 GHz CPU and 8 GB memory controlling a HiVi M200MKIII loudspeaker as the sender. The default speaker volume is 80dB (which is measured by a decibelmeter APP at 1m distance), and the default distance is 1m. At the receiver side, we use Galaxy Note4 in most of our experiments. We show the performance comparison across different smartphones in Section 5.3.5. The sampling rate on the receiver is 44.1KHz.

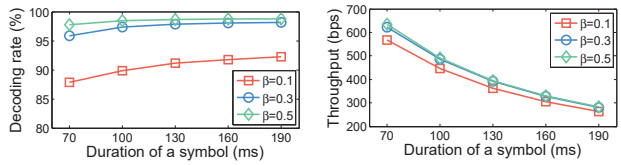
5.1 Subjective Perception Assessment

First, we conduct a user study to examine whether Dolphin has any noticeable auditory impact on the original audio content and identify a good set of design parameters for better auditory experience. Our user study is conducted with 40 participants (22 males and 18 females) in the age range from 18 to 46. We evaluate the quality of data-embedded audio with scores 5 to 1, which respectively indicate "completely unobtrusive", "almost unnoticeable", "not annoying", "slightly annoying", "annoying". We test four different types of audio sources, including soft music, rock music, human voice, and advertisements. Each type of sound source is evaluated using 10 different samples. The experiments are conducted in an office with the speaker volume set to be 80dB and a speaker-smartphone distance of 1m.

5.1.1 *Embedding Strength Coefficient β*

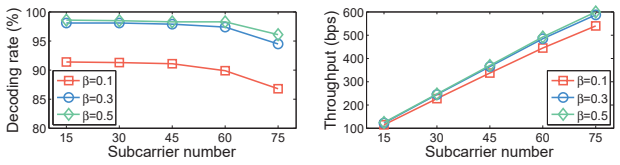
The embedding strength coefficient β is the most critical parameter that determines the embedded signal energy and affects the subjective perception. A large value of β makes communication more robust but it makes it easier for the user to perceive the change in the received audio. To isolate the impact of β and show the effectiveness of our adaptive embedding approach, we use ASK as the modulation method for all symbols and let the energy of each symbol signal not change with the energy of its carrier audio (called static embedding). In static embedding, we measure E_{sl} of 10 different samples for each type of audio source, and calculate the average value $\overline{E}_{sl}$ in advance.

Figure 12(a) presents the average subjective perception scores as β varies from 0.1 to 0.9 in static embedding. As expected, the subjective perception score decreases as β increases. However, different types of audio have different sensitivity to β. The scores of soft music and advertisements are in general higher than those of voice and rock music. In the case of human voice with no background music, the noise is easy to be perceived when the speech pauses. As for rock music, some pieces contain abundant energy in high frequencies. If we embed data symbols into such pieces and change the energy distribution, such changes are also easy to be perceived. Overall, we observe that for $\beta \geq 0.3$, almost all the subjective perception scores drop below 4 for different types of audios. On the other hand, a low β reduces the robustness of our system.



(a) Decoding rate (b) Throughput

Figure 13: The impact of T with different β on the decoding rate and throughput.



(a) Decoding rate (b) Throughput

Figure 14: The impact of N with different β on the decoding rate and throughput.

5.1.2 *Adaptive Embedding Improvement*

In adaptive embedding, we calculate the energy of each piece of carrier audio corresponding to a symbol in real-time, based on which the energy of a symbol signal is changed adaptively according to Equations 6 and 7. Figure 12(b) evaluates our adaptive embedding method (Section 4.2.3) which tries to balance the tradeoff between audio quality and signal robustness. Compared with Figure 12(a), the scores of all types of audios are obviously improved. In particularly, we observe that the use of EDK improves significantly the scores of rock music, as we explained in Section 4.2.1. The scores of voice also improve because, when the speech pauses, we do not embed data bits into it. On the other hand, the improvement of soft music is not obvious because the energy of soft music is relatively steady. When $\beta = 0.5$, of all types of audios achieve a score close to 4. Hence, β should not be larger than 0.5 to guarantee relatively good auditory experience in practice.

5.2 Communication Performance

We now evaluate the communication performance of Dolphin based on different metrics.

5.2.1 *Decoding Rate and Throughput*

The decoding rate and throughput are mainly affected by two factors: the symbol duration T and the number of subcarriers N. N is set to be 60 when we evaluate the impact of T; T is set to be 100ms when we evaluate the impact of N. We also evaluate the system performance with different β values. The test audio sources for different β include soft music, rock music, human voice, and advertisements. We record the results of different types of audios and calculate the average value. The experiments are conducted in an office with the speaker volume set to 80dB and a speaker-smartphone distance of 1m.

Figure 13(a) shows the impact of the symbol duration on the decoding rate. As can be seen, the decoding rate increases when T increases, since a longer duration allows for more repetitions of the same signal. When T is larger than 100ms, the average decoding rate over all audios with $\beta = 0.5$ is above 98%. However, 100% reliability is very hard to achieve in practice. In addition, we observe that the decoding rate for a given T is different for different β. When $\beta = 0.1$, the subjective perception score is ideal, but the decoding rate is obviously lower than that with $\beta = 0.3$. Therefore, there exists a tradeoff between audio quality and signal robustness. Figure 13(b) shows the effect of the symbol duration on the throughput performance. As expected, the throughput decreases when T increases. Similar to the decoding rate, a given T yields different throughputs for different β.

Figure 14(a) shows the impact of the number of subcarriers on the decoding rate. We observe that the decoding rate drops significantly when N is larger than 60. To ensure the same level of subjective perception, the total energy embedded in a symbol is constant once the piece of audio carrier is determined. If the number of subcarriers increases, then the energy per subcarrier decreases. Further, we observe that the performance with $\beta = 0.1$ is still obviously lower. Since a larger β can improve the signal robustness with ac-ceptable unobtrusiveness, we set $\beta = 0.5$ in the following experiments. Figure 14(b) shows that throughput increases when N increases, because more subcarriers can carry more information.

When T =100ms and N=60, the average throughput of different types of audios is about 500bps. We believe this throughput is sufficient for most of our targeted application scenarios because the embedded information is usually some side information (e.g., verbal descriptions of video/audio contents). Take a 1-minute advertisement as an example. Assume the ad can load about $500 \times 60/8 = 3750$ letters, and a word consists of 10 letters on average. Then, there are about 375 words which can be instant notifications, offers, and promotions, etc.

Table 1: The average real-time encoding time (ms) of a symbol on the PC.

OEC Coding (ms)	0.57
Energy Analysis (ms)	0.35
Adaptive Embedding (ms)	0.18
Total (ms)	1.1

5.2.2 *Encoding and Decoding Time*

To evaluate Dolphin's ability to support real-time communication, we measure per-symbol encoding and decoding time. We use the default setting: $T = 100ms$ and $N = 60$. At the sender, we measure the encoding time of each symbol including *OEC Coding*, *Energy Analysis*, and *Adaptive Embedding*. At the receiver, we first perform *Preamble Detection* and *Frequency Selectivity Estimation (FSE)* for each packet, then we decode each symbol. Therefore, the decoding time of each symbol only consists of *Time Selectivity Estimation (TSE)*, *Doppler Frequency Offset Elimination (DFOE)*, *Symbol Extraction* and *OEC Error Correction*.

Table 1 shows that the average encoding time of a symbol is much shorter than the symbol duration (100ms) and hence, the sender is able to support real-time operation. Table 2 plots the average time of decoding operations in two smartphones. The results show that *Preamble Detection* is the most time-consuming operation. This is because the envelopes of different piece of audio needs to be calculated to find the maximum and it involves iterative operations. However, *Preamble Detection* is only necessary for each packet

Table 2: The average time (ms) of decoding a symbol and pre-processing in real-time on two smartphones.

		Note4	S4
Prep.(ms)	Preamble Detection	369.2	542.9
	FSE	23.4	34.5
Dec.(ms)	TSE and DFOE	22.3	32.2
	Symbol Extraction	0.85	1.6
	OEC Error Correction	1.5	2.8
Dec. Sub-total (ms)		24.6	36.6

Table 3: The average goodput under different communicating distances.

Distance (m)	0∼1	1∼2	2∼3	3∼4	4∼6
Goodput (bps)	261.3	209.1	156.8	104.5	52.3

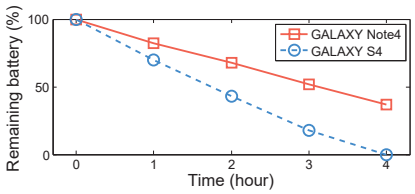


Figure 17: Energy consumption of Dolphin.

rather than each symbol, and the total decoding time of each symbol is also smaller than the symbol duration. Hence, the receiver can also support real-time decoding although with a short delay due to *Preamble Detection*.

5.2.3 Error Correction

In this experiment, we use the orthogonal error correction (OEC) scheme to correct different levels of bit error rates under different communicating distances. We vary the distance from 0 to 6m in a long corridor with a speaker volume of 80dB. We adjust the intra-symbol error correction parameter $n-k$ and the inter-symbol erasure correction parameter m to completely guarantee the correctness of decoded signals with different distances. Then, we calculate the corresponding goodput, which is defined as the ratio of the correctly decoded data bits (excluding the bits used for error-correction) to the total transmission time. From Table 3, it can be seen that the goodput decreases as the distance increases. This is because a longer communicating distance leads to more bit errors and thus we have to use more coding bits.

5.2.4 Energy Consumption

We also measure the battery consumption of our Dolphin prototype on different smartphone platforms. Figure 17 shows the remaining battery percentage of GALAXY Note4 and S4 after a 4-hour continuous acoustic signal capture and decoding. It shows that, Dolphin can support real-time embedded information delivery for more than 4 hours, and this time period is enough for most application scenarios, e.g., a basketball or football game.

5.3 Other Practical Considerations

We now evaluate the impact of other practical factors on Dolphin's performance without using OEC.

5.3.1 Distance and Angle

The impact of distance on decoding rate is significant, because the acoustic power decays with the square of the distance. We vary the distance from 2 to 10m in a long corridor with a speaker volume of 77dB and 80dB. As shown in Figure 15(a), the decoding rate decreases as the distance increases but remains above 80% for distances up to 6m with a volume of 77dB and 8m with a volume of 80dB. Obviously, Dolphin can support even longer distances by adjusting the speaker volume.

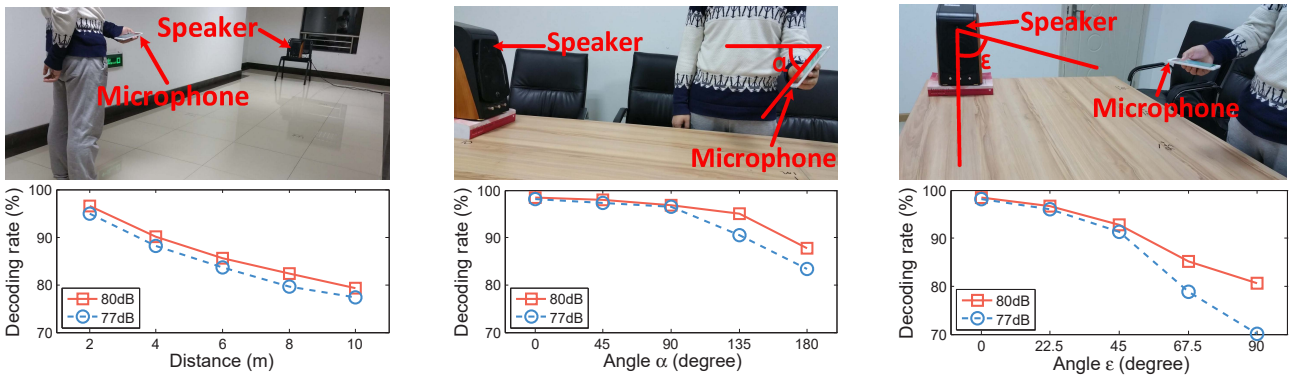
In addition, we examine Dolphin by varying the smartphone rotation and horizontal angles (Figures 15(b) and (c)) to evaluate the impact of misalignment between the sender and the receiver for two speaker volumes: 77dB and 80dB. In the first experiment, we rotate the smartphone vertically from 0° to 180°. In the second experiment, we vary the horizontal angle from 0° to 90° while keeping the microphone towards the direction of the sound source. In both experiments, the speaker-smartphone distance remains equal to 1m. As shown in Figure 15(b), Dolphin's overall performance is relatively stable when the smartphone rotates vertically from 0° to 90°. Further, when α=180°, i.e., the speaker and the smartphone face towards opposite directions, the decoding rate is still above 80%. This demonstrates the practicality of Dolphin, which does not require that the users accurately keep the microphone towards the direction of the sound source. Figure 15(c) shows that the decoding rate remains relatively stable, when the horizontal angle ε varies from 0° to 45°, but decreases sharply for larger angles . This is because the HiVi M200MKIII speaker transmits directionally. If the smartphone lies within the speaker's transmission conical beam, the microphone can capture the audio directly. Otherwise, the audio only can arrive at the receiver by reflection. Even so, the decoding rate is still above 80% with a speaker volume of 80dB when $\varepsilon = 90°$. Dolphin ensures good performance for most places around the speaker.

5.3.2 Ambient Noise

Figure 16(a) shows the impact of the ambient noise on the decoding rate. We performed experiments at three different locations: an office, a restaurant, and a square, and varied the volume from 74dB to 82dB. We observe that Dolphin is resilient to ambient noise, maintaining a decoding rate above 90% at all three locations. This is because we select the appropriate frequency bandwidth for the embedded signal to reduce the influence of ambient noise. In the office, the ambient noise is very small (Figure 2). In the cafe, the ambient noise is mainly due to the conversations among customers. However, the frequency range of human voice is relative low and does not interfere significantly with the sound signals above 8KHz. In a square, there are different sound sources, some of which generate higher frequency sounds, and Dolphin performs slightly worse compared to the other two locations.

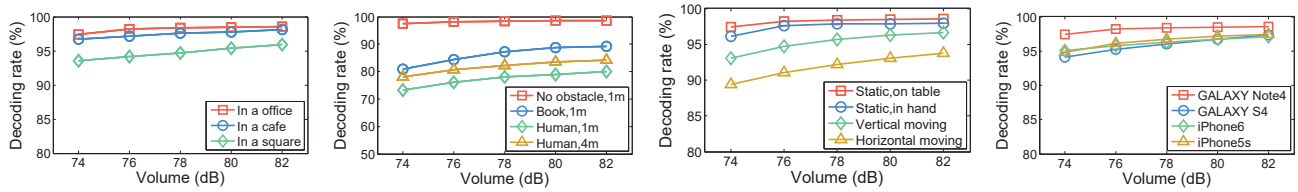
5.3.3 Obstacles

In this section, we discuss the impact of obstacles between the sound source and the receiver microphone on the decoding rate. The obstacles include a $28\times21\times5cm$ book or a human between the HiVi M200MKIII (sender) and the Galaxy Note4 phone (receiver). The LOS between the speaker and the microphone is completely blocked by the obstacles. From Figure 16(b), we can observe that the presence of an obstacle obviously decreases the decoding rate while the sound signals can still reach the receiver via diffraction. On the



(a) Distance. (b) Rotation angle α. (c) Horizontal angle ε.

Figure 15: The impact of distance and angle on decoding rate.



(a) Ambient noise. (b) Obstacles. (c) Handhold and motion. (d) Different Smartphones.

Figure 16: The impact of various practical settings on decoding rate.

one hand, the size of obstacles will affect the performance. When the volume of speaker is above 80dB, the decoding rate with the book blocking at the distance of 1m is about 90% but the decoding rate with a human blocking at the distance of 1m is about 80%. On the other hand, the distance also affects the performance. The decoding rate with the human blocking at the distance of 4m is even higher than that at the distance of 1m. As can be seen from Figure 15(a), the decoding rate decreases as the communicating distance increases. However, the HiVi M200MKIII speaker transmits directionally. When the human stands very close to the speaker, the sound conical beam will be completely blocked. When the human gradually moves away from the speaker, the unblocked signals can still reach the receiver via diffraction. Obviously, Dolphin will perform better with obstacles by using a speaker with wider transmission angle. This is a great advantage of Dolphin compared to unobtrusive screen-to-camera communication systems which are very sensitive to any obstacles.

5.3.4 *Device Motion*

We now study the impact of device motion on Dolphin's performance. We evaluate three types of motion: (i) a static user holds the Galaxy Note4 in the air facing the HiVi M200MKIIIl; in this case, the motion is due to the slight hand shaking; (ii) the user moves the smartphone slowly towards and away from the speaker (horizontal moving); and (iii) the user moves the smartphone slowly in parallel to the speaker (vertical moving). Figure 16(c) shows the results when the volume from 74dB to 82dB. First, in the case of a static user holding the phone, the performance is very close to the case of the phone placed on a table, i.e., the impact of slight hand shaking is negligible. On the other hand, the impact of the actual device motion is more prominent, especially in the case of horizontal moving, due to Doppler frequency shift (in Doppler frequency shift determinant $\nu_0 \cos\theta$, $\cos\theta$ is 1 for horizontal moving but takes its minimum value for vertical moving). However, the decoding still remains higher than 90% with both types of motion when the the volume is above 76dB. The use of pilots in each symbol helps Dolphin successfully estimate the Doppler frequency offset and reduce its effect.

5.3.5 *Different Smartphone Models*

Finally, we examine the impact of different smartphone platforms on Dolphin's performance. We use four smartphone models (GALAXY Note4, GALAXY S4, iPhone 6, and iPhone 5s). Our current implementation of the Dolphin receiver is based on the Android framework. To test Dolphin on iPhone 6 and iPhone 5s, we use the smartphones to capture the audio signals and decode them on the PC. We vary the volume from 74dB to 82dB at a distance of 1 m. As shown in Figure 16(d), the performance of GALAXY Note4 is the best and that of GALAXY S4 is the worst; such performance differences are mainly caused by the frequency selectivity of microphones. Nonetheless, all four models maintain a decoding rate higher than 95% when the volume is above 76dB. This is due to the use of pilots in the first symbol of each packet that allow the receiver to estimate the frequency selective fading function and largely eliminate the impact of frequency selectivity.

Discussion. Note that, Dolphin focuses on signal broadcasting application scenarios, and thus we implemented data encoding on the PC (connected to a high-power loudspeaker) as the transmitter. That is, Dolphin does not target smart device to smart device communication. However, to test the performance of Dolphin using a speaker of poor quality, we use GALAXY Note4 or GALAXY S4 as the

sender and GALAXY Note4 as the receiver in our experiments. Since the Dolphin sender is currently implemented on a PC, we use a PC to encode the data-embedded audio and playback them on the smartphone sender. Compared with HiVi M200MKIII loudspeaker, the smartphone speakers have lower volume and higher frequency selectivity. In addition, the smartphone speakers have higher noise which influences the auditory perception. In our test, the volume of smartphone speakers is set to 100%, which is around 65dB. We focus on the performance under several practical considerations (e.g., distance, angle and obstacles). We found that Dolphin supports up to 5-meter signal capture distance and 360° listening angle with the decoding rate above 80%. In addition, the decoding rate with the human blocking at the distance of 1m is above 85%. The results show that the signal capture distance is also limited by the volume, but better performance in listening angle and obstacles benefits from the wider transmission angle of smartphone speakers. Not surprisingly, the auditory perception is worse due to the poor quality of smartphone speakers.

6. RELATED WORK

Unobtrusive screen-camera communication: In recent years, extensive research efforts have led to specially-designed color barcodes for barcode-based VLC [18, 8, 9, 21, 30, 23, 10]. To eliminate the resource contention in the above designs, several recent studies seek to achieve unobtrusive screen-to-camera communication. Along this direction, Yuan et al. leverages watermarking to embed messages into an image [28]. In [4], the authors proposed to embed data hidden in brightness changes upon two consecutive frames. In [26, 22, 20], the key idea is to switch barcodes with complementary hues. PiCode and ViCode [11] integrate barcodes with existing images to enhance viewing experience. The most recent effort is Hilight [13, 14], which leverages the alpha channel to encode bits into the pixel translucency change. Compared to Dolphin, unobtrusive screen-to-camera communication requires well-controlled alignment of the camera and the screen and obstacle-free access.

Aerial acoustic communication: Aerial acoustic communication over speaker-microphone links has been studied in [7, 16, 31, 12, 17, 15, 29]. In [7], the authors used multiple tones to transmit data in an audible mode or a single tone in an inaudible mode. Dhwani [16] and PriWhisper [31] aim to realize secure acoustic short-range communication by leveraging the microphone-speaker links on mobile phones. In [12], chirp signals were used to realize an aerial acoustic communication system. In [15] and [29], the authors proposed to hide information in audios and use the loudspeaker and the microphone with flat frequency response to display and record data-embedded audio. However, [7, 16, 31] only focus on reliable speaker-microphone data communication, while [15, 29] were not designed for off-the-shelf smartphones without considering the characteristics of acoustic channel, and [12, 17] used the inaudible audio signals to achieve very low-rate communications. In contrast, Dolphin aims at establishing dual-mode unobtrusive communication using off-the-shelf smartphones.

Audio watermarking: With the development of network and digital technologies, digital audio is easy to be reproduced and retransmitted. Audio watermarking [6, 3, 19, 2, 24], as a means to identify the owner, encodes hidden copyright information into the digital audio. The common encoding schemes used in audio watermarking include least significant bit (LSB), spread spectrum [6], echo hiding, DCT and DWT [24] etc. In order to prevent the watermark from being readily removed by pirates, it must be robust to common audio processing (e.g., MP3 compression, cropping and resampling) and be statistically undetectable to users. To this end, for example, LSB manipulates the least significant bit of the sample points, and DWT selectively manipulates some coefficient of wavelet domain. The position to be modified usually is controlled by a key which is only known to the owner. Unlike audio watermarking which directly provides embedded copyright information audio files to users and aim to ensure copyright information cannot be removed, Dolphin seeks to enable unobtrusive data communication and provide relevant side information which users can obtain through their smartphones when the speaker plays the audio, by addressing several challenges unique to the nature of the acoustic signal propagation and speaker-microphone characteristics. Therefore, Dolphin must address real-world signal degradations over the speaker-microphone channel while watermarking does not. To achieve our goal, modifying the original audio and decoding the signals in Dolphin must take into account ambient noise, the characteristics of commercial speakers and microphones, and channel estimation.

7. CONCLUSIONS

We presented and implemented Dolphin, a new form of real-time unobtrusive dual-mode speaker-microphone communication atop any audio content generated on the fly. We implemented Dolphin on off-the-shelf smartphones and evaluated it extensively under various environments and practical considerations. Dolphin has its own superiorities and can be adopted as a complementary or joint dual-mode communication strategy with existing unobtrusive screen-to-camera communication systems to enhance the system performance under various practical settings.

8. ACKNOWLEDGMENTS

We would like to thank the shepherd and the anonymous reviewers for their useful comments and suggestions. Qian's research is supported in part by National Natural Science Foundation of China under Grant No. 61373167, National Basic Research Program of China under Grant No. 2014CB340600. Kui's research is supported in part by US National Science Foundation under grant CNS-1421903. Lu's research is supported in part by National Science Foundation under grant CNS-1566374. Qian Wang is the corresponding author.

9. REFERENCES

[1] https://www.sandvine.com/trends/global-internet-phenomena.

[2] Arnold, M. Audio watermarking: Features, applications, and algorithms. In *Proc. of ICME* (2000), Citeseer, pp. 1013–1016.

[3] Bassia, P., Pitas, I., and Nikolaidis, N. Robust audio watermarking in the time domain. *IEEE Transactions on multimedia 3*, 2 (2001), 232–241.

[4] Carvalho, R., Chu, C.-H., and Chen, L.-J. Ivc: Imperceptible video communication. In *Proc. of HotMobile (poster)* (2014), Citeseer.

[5] Coleri, S., Ergen, M., Puri, A., and Bahai, A. Channel estimation techniques based on pilot arrangement in ofdm systems. *IEEE Trans. Broadcasting 48*, 3 (2002), 223–229.

[6] Cox, I. J., Kilian, J., Leighton, F. T., and Shamoon, T. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Processing 6*, 12 (1997), 1673–1687.

[7] Gerasimov, V., and Bender, W. Things that talk: using sound for device-to-device and device-to-human communication. *IBM Systems Journal 39*, 3.4 (2000), 530–546.

[8] Hao, T., Zhou, R., and Xing, G. COBRA: color barcode streaming for smartphone systems. In *Proc. of MobiSys* (2012), ACM, pp. 85–98.

[9] Hu, W., Gu, H., and Pu, Q. LightSync: unsynchronized visual communication over screen-camera links. In *Proc. of MobiCom* (2013), ACM, pp. 15–26.

[10] Hu, W., Mao, J., Huang, Z., Xue, Y., She, J., Bian, K., and Shen, G. Strata: layered coding for scalable visual communication. In *Proc. of MobiCom* (2014), ACM, pp. 79–90.

[11] Huang, W., and Mow, W. H. Picode: 2d barcode with embedded picture and vicode: 3d barcode with embedded video. In *Proc. of MobiCom* (2013), ACM, pp. 139–142.

[12] Lee, H., Kim, T. H., Choi, J. W., and Choi, S. Chirp signal-based aerial acoustic communication for smart devices. In *Proc. of INFOCOM* (2015), IEEE, pp. 2407–2415.

[13] Li, T., An, C., Campbell, A. T., and Zhou, X. Hilight: Hiding bits in pixel translucency changes. *ACM SIGMOBILE Mobile Computing and Communications Review 18*, 3 (2015), 62–70.

[14] Li, T., An, C., Xiao, X., Campbell, A. T., and Zhou, X. Real-time screen-camera communication behind any scene. In *Proc. of MobiSys* (2015), ACM, pp. 197–211.

[15] Matsuoka, H., Nakashima, Y., and Yoshimura, T. Acoustic communication system using mobile terminal microphones. *NTT DoCoMo Tech. J 8*, 2 (2006), 2–12.

[16] Nandakumar, R., Chintalapudi, K. K., Padmanabhan, V., and Venkatesan, R. Dhwani: secure peer-to-peer acoustic nfc. *ACM SIGCOMM Computer Communication Review 43*, 4 (2013), 63–74.

[17] Nittala, A. S., Yang, X.-D., Bateman, S., Sharlin, E., and Greenberg, S. Phoneear: interactions for mobile devices that hear high-frequency sound-encoded data. In *Proc. of SIGCHI Symposium on Engineering Interactive Computing Systems* (2015), ACM, pp. 174–179.

[18] Perli, S. D., Ahmed, N., and Katabi, D. Pixnet: interference-free wireless links using lcd-camera pairs. In *Proc. of MobiCom* (2010), ACM, pp. 137–148.

[19] Swanson, M. D., Zhu, B., Tewfik, A. H., and Boney, L. Robust audio watermarking using perceptual masking. *Signal processing 66*, 3 (1998), 337–355.

[20] Wang, A., Li, Z., Peng, C., Shen, G., Fang, G., and Zeng, B. Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication. In *Proc. of MobiSys* (2015), ACM, pp. 181–195.

[21] Wang, A., Ma, S., Hu, C., Huai, J., Peng, C., and Shen, G. Enhancing reliability to boost the throughput over screen-camera links. In *Proc. of MobiCom* (2014), ACM, pp. 41–52.

[22] Wang, A., Peng, C., Zhang, O., Shen, G., and Zeng, B. Inframe: Multiflexing full-frame visible communication channel for humans and devices. In *Proc. of HotNets* (2014), ACM, p. 23.

[23] Wang, Q., Zhou, M., Ren, K., Lei, T., Li, J., and Wang, Z. Rainbar: Robust application-driven visual communication using color barcodes. In *Proc. of ICDCS* (2015), IEEE, pp. 537–546.

[24] Wang, X.-Y., and Zhao, H. A novel synchronization invariant audio watermarking scheme based on dwt and dct. *IEEE Transactions on signal processing 54*, 12 (2006), 4835–4840.

[25] Wicker, S. B. *Reed-Solomon Codes and Their Applications*. IEEE Press, Piscataway, NJ, USA, 1994.

[26] Woo, G., Lippman, A., and Raskar, R. Vrcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *Proc. of ISMAR* (2012), IEEE, pp. 59–64.

[27] Yost, W. A., and Schlauch, R. S. Fundamentals of hearing: An introduction. *The Journal of the Acoustical Society of America 110*, 4 (2001), 1713–1714.

[28] Yuan, W., Dana, K., Ashok, A., Gruteser, M., and Mandayam, N. Dynamic and invisible messaging for visual mimo. In *Proc. of WACV* (2012), IEEE, pp. 345–352.

[29] Yun, H. S., Cho, K., and Kim, N. S. Acoustic data transmission based on modulated complex lapped transform. *IEEE Signal Processing Letters 17*, 1 (2010), 67–70.

[30] Zhang, B., Ren, K., Xing, G., Fu, X., and Wang, C. SBVLC: Secure barcode-based visible light communication for smartphones. In *Proc. of INFOCOM* (2014), IEEE, pp. 2661–2669.

[31] Zhang, B., Zhan, Q., Chen, S., Li, M., Ren, K., Wang, C., and Ma, D. : Enabling keyless secure acoustic communication for smartphones. *IEEE Internet of Things Journal 1*, 1 (2014), 33–45.

[32] Zwicker, E., and Zwicker, U. T. Audio engineering and psychoacoustics: Matching signals to the final receiver, the human auditory system. *Journal of the Audio Engineering Society 39*, 3 (1991), 115–126.