Ripple II: Faster Communication through Physical Vibration

Nirupam Roy, Romit Roy Choudhury

University of Illinois at Urbana Champaign

Abstract

We envision physical vibration as a new modality of data communication. In NSDI 2015, our paper reported the feasibility of modulating the vibration of a smartphone's vibra-motor. When in physical contact with another smartphone, the accelerometer of the second phone was able to decode the vibrations at around 200 bits/s. This paper builds on our first prototype, but redesigns the entire radio stack to now achieve 30 kbps. The core redesign includes (1) a new OFDM-based physical layer that uses the *microphone* as a receiver (instead of the accelerometer), and (2) a MAC layer that detects collision at the transmitter and performs proactive symbol retransmissions. We also develop two example applications on top of the vibratory radio: (1) a finger ring that transmits vibratory passwords through the finger bone to enable touch based authentication, and (2) surface communication between devices placed on the same table. The overall system entails unique challenges and opportunities, including ambient sound cancellation, OFDM over vibrations, back-EMF based carrier sensing, predictive retransmissions, bone conduction, etc. We call our system *Ripple II* to suggest the continuity from the NSDI 2015 paper. We close the paper with a video demo that streams music as OFDM packets through vibrations and plays it in real time through the receiver's speaker.

1 Introduction

Motivation: Project Ripple [34] is an attempt to enable communication through physical vibrations. The core idea is to harness the vibration motor (present in all smartphones and wearable devices) as a transmitter, and a motion sensor (like an accelerometer) as a receiver. When two smartphones come in physical contact to each other, the transmitter phone can vibrate to transfer bits of information. Transmission is even possible through other solid channels, such as between devices placed on a tabletop, or a finger ring communicating to a smartphone through bone conduction. While the exact application remains an open question (especially in the presence of NFC-like technologies), areas such as Internet of Things (IoT), intra-body networks, wearable security, and mobile payments are calling for new forms for short range communication. Qualities of a vibratory radio, including zero RF radiation, contact-only authentication, mass-scale availability, and intuitive usability, may together fill an emerging business need. This project is motivated by this "bottom up" thinking and focuses on pushing forward the vibratory capabilities.

Prior Work: Of course, the fundamental idea of utilizing vibration as a communication modality dates back to acoustics – speakers modulate bits of information into air vibrations that are picked up by microphones. Air vibrations were later extended to water, enabling under water communication [7, 6, 11] and various applications, such as SONAR [41]. In recent years, vibration through solids has been of interest, motivated primarily by the need for proximal communication. Authors in [22, 38] used Morse-style communication at 5 bit/s to exchange security keys between two mobile phones in contact. Last year, *Ripple* [34] broke away from ON/OFF communication, and developed a viable radio through techniques such as multi-carrier amplitude modulation, vibration braking, and simultaneous transmission over the 3 axes of the accelerometer. A self-sound cancellation technique also prevented acoustic eavesdroppers from decoding the sounds of vibration, offering improved security over RF based approaches. As a first attempt to vibratory radio design, Ripple achieved data rates of ≈ 200 bits/s, but left various challenges and opportunities unaddressed. This paper presents a subsequent work – Ripple II – aimed at a far more mature radio stack and two example applications.

Technical Core: Ripple II's core redesign entails the following: (1) Replacing the accelerometer with the microphone as a receiver of vibrations. The key challenge pertains to separating vibrations from ambient sounds "picked up" by the microphone. While the availability of a second microphone offers the opportunity for sound cancellation, vibrations partly pollute the second microphone as well. Moreover, techniques such as active noise cancellation are inadequate since residual phase mismatches – often tolerable in human hearing applications [37] – seriously affect demodulation. We develop variants of adaptive filtering schemes, enhanced with an understanding of the interference conditions. (2) We also discover an opportunity that allows the vibra-motor

to partially sense ambient sound interference, through a phenomenon called back-EMF in electronic circuits. The transmitter extrapolates from this partial information, using curve fitting techniques, and develops a proactive symbol retransmission scheme. The problem is new to the best of our knowledge – unlike existing wireless systems, here the transmitter is aware of the receiver's interference conditions and can adapt at the granularity of symbols. This opens both challenges and opportunities.

System and Apps: We engineer a completely functional prototype, which entails a full OFDM stack, coping with ADC saturation, synchronization, error coding, interleaving, etc. Towards real applications, we develop a (clunky) wearable finger ring and demonstrate the viability of transmitting vibratory signals through finger bones. While signals attenuate through human tissues and muscles, effective bit rates of 7.41 Kbps is still possible, adequate for applications like two-factor authentication (i.e., when the user unlocks the phone, the vibratory password decoded by the phone serves as a second channel of authentication). We also explore a second application where devices are placed on tabletops, allowing for one-to-many multicast communication (e.g., a presenter sharing slides with all members in the meeting). Lastly, we include a video demo on our project website [3] – the demo shows the transmitter streaming music through OFDM packets over vibrations and the receiver's speaker playing it in real time.

Platform and Evaluation: Our evaluation platform is composed of laptops, signal generators, vibra-motor chips, microphone chips, and home-grown circuits that interconnect them. In the basic scenario, the vibra-motor is attached to a short pencil to emulate a "stylus" like device, which then touches a microphone chip to transfer information. We generate various ambient sounds in the lab, including soft and loud music, people talking, machine hums, loud thuds and vibrations, and their combinations. Our PHY and MAC layer schemes are evaluated in these settings, against metrics such as SNR gain, bit error rate (BER), throughput, etc. At the application layer, we compute end to end data rate under modestly realistic settings, such as the human wearing the (vibra-motor embedded) ring and touching the microphone chip. We emulate wrist watches as well (2.23 Kbps), and perform an informal user study to understand if they feel the vibrations. We also explore achievable bit rates for tabletop communication, with devices placed at increasing distances on wooden surfaces.

Next Steps: There is much room for continued research and improvement. First, we have little understanding of PHY capacity and MAC layer optimality; intuitively, we believe that modeling the devices and the channel can yield reasonable performance bounds. Second, the sound cancellation techniques can perhaps benefit from deeper signal processing expertise – we have initiated collaboration towards this goal. Third, microphones and accelerometers may together present new opportunities that remain untapped in this paper. Fourth, while our prior paper mitigated attacks on vibratory sounds, visual attacks still remain a threat – a high speed camera, with line of sight to the device, may be able to decode vibrations. Finally, we need guidance on other possible use-cases and applications [5] of vibratory radios. Our ongoing work is focused on all these aspects.

In summary, the contributions of this paper are:

- *An OFDM based vibratory radio with microphones as the receiver.* The PHY layer uses variants of adaptive filtering to isolate vibrations from ambient sounds at the microphone; the MAC layer develops a transmitter side carrier sensing mechanism and uses it for proactive symbol retransmission.

- *A completely functional system borne out of significant engineering effort.* The effort includes hardware circuits on bread boards, to drivers for the vibra-motor, to bone conduction and real-time music streaming. Instantiation of the system in two applications: touch based authentication and surface communication.

The overall architecture of Ripple II is illustrated in Figure 1. The rest of the paper expands on the main modules (shaded in gray) and briefly touches upon the techniques borrowed from literature, and the engineering effort in building the prototype.

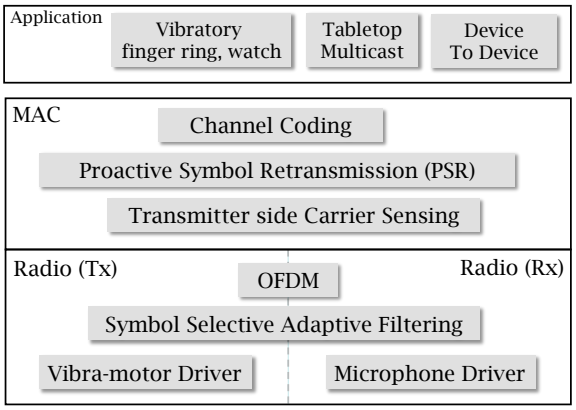

Figure 1: Ripple II's system architecture.

2 Development Platform and Overview

2.1 Vibratory Transmitter

A vibration motor (also called "vibra-motor") is an electro-mechanical device that moves a metallic mass

rhythmically around a neutral position to generate vibrations. While there are various kinds of vibra-motors, a popular one is called *Linear Resonant Actuators (LRA)* shown in Figure 2. With LRA, vibration is generated by the linear movement of a magnetic mass suspended near a coil, called the "voice coil". Upon applying AC current to the motor, the coil also behaves like a magnet (due to the generated electromagnetic field) and causes the mass to be attracted or repelled, depending on the direction of the current. This generates vibration at the same frequency as the input AC signal, while the amplitude of vibration is dictated by the signal's peak-to-peak voltage. Thus LRAs offer control on both the magnitude and frequency of vibration. As an aside, most mobile phones today use LRA based vibra-motors.

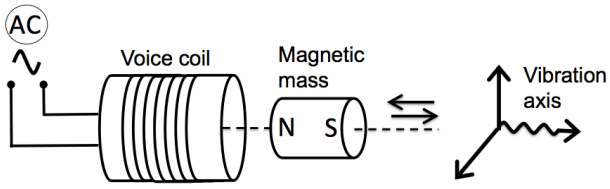

Figure 2: Basic sketch of an LRA vibra-motor.

We control the vibra-motor through an Agilent 33500B waveform generator, which is indirectly controlled by MATLAB running on a laptop. The laptop generates the desired digital samples; the waveform generator converts the samples to an analog wave and transmits to the vibra-motor. The peak-to-peak output voltage is stabilized at 5V, the maximum supported by the vibra-motor chip. We generate OFDM symbols through MATLAB and drive the motor as desired.

2.2 Microphone as a Receiver

Our prior work [34] used a vibra-motor as the transmitter and an accelerometer as the receiver[1]. The accelerometer demodulated vibratory QPSK symbols and corrected for errors using simple gray coding techniques. The low bandwidth of accelerometer chips (800Hz) proved to be the main bottleneck to link capacity, resulting in $\approx$ 200 bits/s. This paper breaks away from accelerometers and identifies the possibility of using microphones as a vibration receiver.

Like accelerometers, microphones also transduce physical motion to electrical signals using a diaphragm that responds to changes in (acoustic) air pressure. Figure 3 shows a microphone chip and the basic internal architecture – as the diaphragm vibrates inside a magnetic field, the produced electrical signals are amplified and sampled by an ADC. *Unsurprisingly, the diaphragm can also be made to vibrate by physically touching a vibra-motor to the microphone chip.* Since microphones are designed for greater sensitivity and operate over a wider frequency range, they can serve as a better receiver (an alternative to accelerometers). The tradeoff, however, is that the vibration measured at the ADC is actually an aggregate of the physical vibration and the air vibration from ambient sounds (e.g., people talking). Ripple II needs to isolate physical from acoustic vibrations to accomplish high bandwidth vibratory communication.

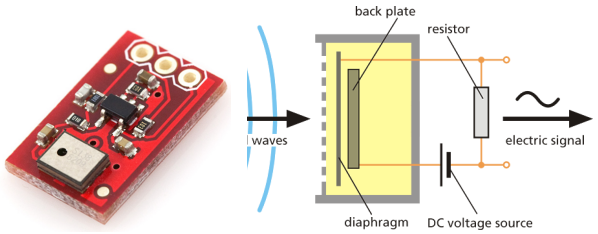

Figure 3: (a) MEMS microphone chip, the diaphragm hole near bottom left (b) Microphone circuit sketch.

Figure 4 shows our overall hardware set-up. The vibra-motor is taped to the back of a short pencil and the tip of the pencil now acts like a stylus, touching the microphone chip. Transmission bits produced by the laptop are converted to a signal waveform by the signal generator, which then drives the transmitter; the microphone decodes these bits through realtime processing on a laptop. The following subsections detail the technical modules in the PHY, MAC, and Application layers.

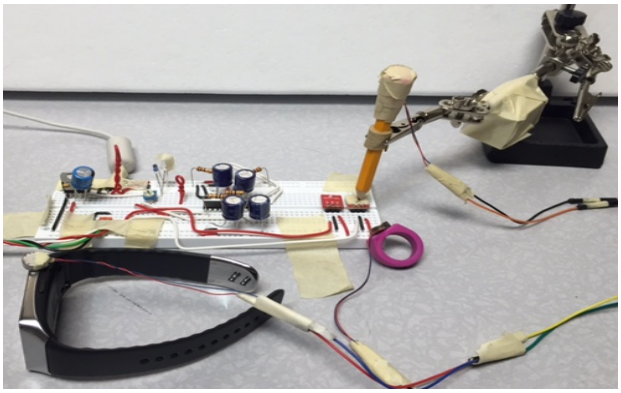

Figure 4: Ripple II's experimentation set-up (3 vibra-motors attached to a pencil, ring, and watch). The stylus touching a microphone, the second microphone nearby.

3 PHY: Vibratory Radio

We begin with the design of the microphone-receiver, followed by our implementation of OFDM.

3.1 Separating Vibration from Sound

While the microphone offers larger bandwidth compared to the accelerometer, its sensitivity to ambient sound is

[1] Accelerometers are MEMS devices that transduce physical motion into electrical signals by measuring the extent to which a tiny seismic mass moves inside fixed electrodes (see [24] for details).

a disadvantage. Unless filtered out, the vibration SINR will be low, especially in loud environments. We attempted various techniques (algorithms and hacks); we detail the ones that worked and touch upon the failures.

Covering the Sound Hole

The microphone chip has a circular opening (like a small hole) that exposes the diaphragm to air pressure. To prevent ambient sounds from polluting Ripple II's vibratory signals, we covered the hole with a stiff synthetic rubber sheet (somewhat like a stethoscope). However, when a vibrating object comes in contact with this rubber sheet, the air trapped inside the hole still oscillates, causing the diaphragm to produce the desired signals. Figure 5 compares the frequency responses of the altered and the standard microphones for vibration and sound, respectively. Figure 5(a) shows an average 18.2dB gain for vibration signals over the standard microphone; at some frequencies the difference is 43.8dB. On the other hand, Figure 5(b) shows that the average sound attenuation at the altered microphone is around 12.3dB. For both the signal (i.e., vibration) and the noise (i.e., ambient sounds), the higher frequency proves better (useful later in Section 5).

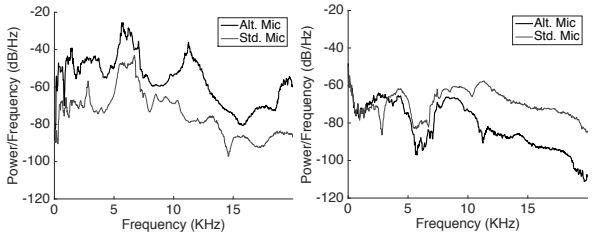

Figure 5: Covering the sound hole offers (a) improved vibration signal and (b) attenuated sound signals in comparison to the standard microphone.

Canceling Ambient Sound

Let us denote the vibration signal from the stylus as $V(t)$ and the ambient sound signal as $S(t)$. Ripple II aims to subtract $S(t)$ from the aggregate signal ($A(t) = V(t) + S(t)$) received through the microphone. A second microphone present in many devices today is a natural opportunity. In an ideal case, the second microphone should only receive the ambient sound $S(t)$ and none of the physical vibration $V(t)$ since the stylus is not in direct contact with it. In reality, however, physical vibrations also leak into the second microphone. Also, both microphones are affected by a high intensity electrical noise, $E(t)$, caused by their common supply voltage. Frequencies of this noise range from 300Hz to 2500Hz and its amplitude is comparable to $V(t)$. Finally, the microphone output also includes a native hardware noise, typically assumed to be uncorrelated additive Gaussian, denoted N_1 and N_2 for the respective microphones.

Based on the above factors, the overall system can be modeled as shown in Figure 6. The signal output from the i^{th} microphone, Y_i can thus be expressed as:

$$Y_i = H_{v_i}V + H_{s_i}S + H_eE + N_i$$

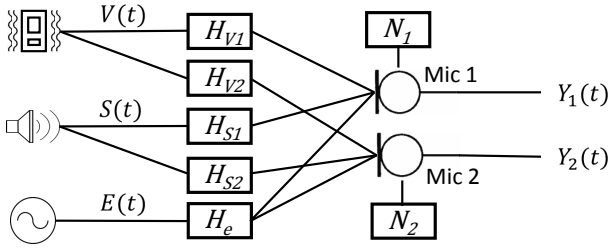

Figure 6: Modeling the signals and interferences at each of the microphones; H denotes the channel matrix and V, S, E, denotes vibration, sound, and electrical noise, respectively.

We note that extraneous physical vibrations may occur when Ripple II is transmitting information (for example, in a moving vehicle). Such vibrations are included in S since it is likely to affect both the microphones similarly. We also note that the electrical noise E is highly correlated and synchronized at both microphones, since they share a common power source. Under this model, our goal is to extract V from Y_1 and decode the content.

Failed Attempts (MIMO, NC, rPCA)

MIMO: We discovered early that electrical noise E can be removed effectively by low pass filtering Y_2 and subtracting from Y_1. Since E dominates and is phase matched across both microphones, the residue after subtraction minimally impacts V. Thus, we can rewrite $Y_i = H_{v_i}V + H_{s_i}S + N_i$. This appears to be in the form of MIMO and hence solvable without difficulty. Unfortunately, the channel matrix for ambient sound, H_{s_i}, cannot be easily measured since Ripple II has no control over the sound sources. Also, due to the time-varying nature, statistical estimates are difficult.

Classical Noise Cancellation seems applicable [25], however, the statistical nature of this algorithm does not mitigate phase mismatches. The result after subtraction does preserve the amplitude of the desired signal, which is often adequate for human perception [37]. In Ripple II, however, we need phase alignment too, or else, QAM based demodulation falters. Put differently, requirements to improve human hearing experience is less stringent than the requirements for data communication.

Robust PCA is an algorithm from 2009 used for background separation [8]. The technique builds on the result that, under certain conditions, a given matrix can be factorized into a sparse and low rank matrix. For instance, in

a talk show video, static background walls could serve as the low rank matrix (due to high similarity across video frames) and the talking people could make up the sparse matrix. In our case, we envisioned the ambient sound to be sparse and the vibration to be low rank (since the cyclic prefix of OFDM symbols can be organized to look identical across time[2].). Unfortunately, we could not design the matrices to attain adequate amount of both sparsity and low rank-ness. During the short time shifts for which the OFDM vibration symbols were identical, the sound signal changed enough that they were not sparse. When sound proved to be sparse over longer time frames, the low rank-ness disappeared. The outcomes of factorization yielded marginal gain.

Symbol Selective Adaptive Noise Filtering

Adaptive filtering (AF) is an established technique that can accept the two microphones' signals as inputs, say ($Y_1 = V_1 + S_1$) and ($Y_2 = V_2 + S_2$), and can attempt to adapt the filter coefficients for Y_2 such that the $Y_1 - Y_2$ is V_1. Conceptually, AF bolsters Y_2 in the regions where it correlates well with S_1, and then subtracts from Y_1. This works best when S_1 and Y_2 are somewhat correlated to each other, but neither is correlated to V_1. However, in our system, when ambient interference is low (i.e., V dominates S), then Y_2 correlates well with V_1 – this is why AF subtracts away the vibratory signals from Y_1, defeating the purpose. However, we observe that if we could identify OFDM symbols that are in error (i.e., S dominates V), then perhaps *only the erroneous symbols* could be subjected to AF. Since S_1 and Y_2 would correlate well in such cases, the result of $Y_1 - Y_2$ could converge to V_1. Using this intuition, we design *Symbol Selective Adaptive Noise Filtering* (SANF), sketched in Figure 7.

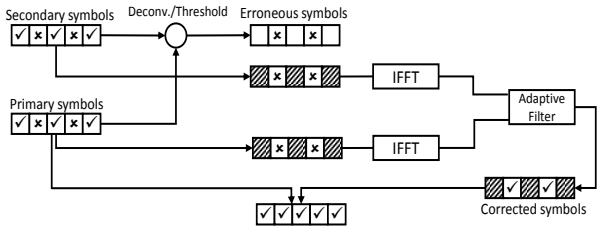

Figure 7: SANF infers erroneous symbols and only feeds these to the AF module.

Erroneous Symbol Detection. The main opportunity emerges from measurements that revealed that the vibratory channel responses at the primary and secondary microphones – H_{v_1} and H_{v_2} – maintain a constant ratio under light or no interference. This is likely due to the same solid channel between the two microphones. In the presence of sound, however, the same ratio gets polluted and thereby loses the constancy property (since sound varies over time). Thus, we first perform channel estimation for pilot subcarriers scattered across the OFDM symbol. We synchronize the secondary microphone and estimate the channel for that same pilot (the slight time offset does not affect due to the protection from the cyclic prefix). Now, deconvolution of the primary and secondary signal in the frequency domain yields the complex gain, α_p for each pilot p.

Recall, the goal is to estimate the pristine ratio of H_{v_1} and H_{v_2} in the presence of sound interference; the α_p we have is still polluted by sound interferences. Thus, we perform a least square estimation of the ratio and compute α^* for each subcarrier. Now, for any non-pilot symbol to be erroneous, the computed complex gain between the primary and secondary must be far from α^* for that subcarrier. Once the erroneous symbols are identified, we convert only those to the time domain, leaving the error-free subcarriers untouched. We obtain the time domain signals from both of the primary and secondary microphone and feed them to an adaptive filter for noise cancellation. The output of the adaptive filter is then demodulated to recover the vibratory symbols.

Amplifier Gain and Clipping

To maximize the power of the vibratory signal, we operate the receiver signal amplifier at near-maximum gain and leave just enough headroom for typical ambient sound (measured empirically). Of course, sometimes the ambient sound exceeds the headroom and drives the amplifier to *saturation* [33]. Figure 8(a) shows the output of the unsaturated amplifier; Figure 8(b) shows the saturated case – a truncated waveform. Unsurprisingly, this "clipping" effect spills energy into other frequencies, causing interference in an OFDM system. We alleviate such frequency distortion effects by replacing the flat saturation region with a cubic spline interpolation of the signal – Figure 8(c).

Our measurements also recorded consistent interference at lower frequencies (< 500Hz), caused by a combination of winds from air vents, thermal noise from electrical equipment, as well as vibrations of the human hand while holding the transmitter. The vibra-motor also exhibits resonance frequency at around 232Hz, causing the system to destabilize due to the high power gain. We deemed it suitable to sidestep these problems and moved the transmission band to begin from 500Hz.

3.2 OFDM over Vibration

We implement OFDM [12] over the vibra-motor and microphone link. Although an engineering effort, we

[2]Without too much details, note that time domain signals can be shifted by several samples and yet, by OFDM design, they will map to the same frequency domain symbol – this is why we could generate low rank-ness.

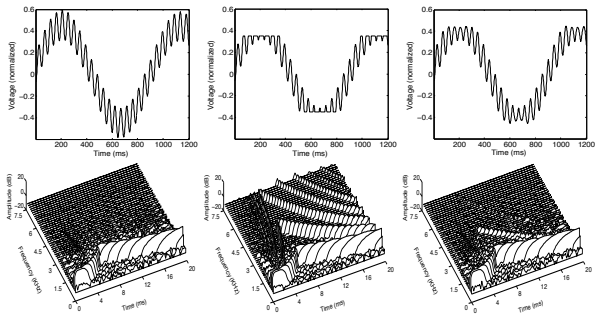

Figure 8: The waveform (1st row) and spectrogram (2nd row) of the (a) actual signal, (b) distorted signal after clipping, (3) corrected signal after spline interpolation.

briefly summarize the parameter selection process, particularly those influenced by the vibratory channel.

Channel Impulse Response

Although the vibratory channel is dominantly time-invariant and frequency selective, human factors such as hand movements and varying angle of contact inject variability. Measurements suggest similarity to a Rician fading model [40], with a strong line of sight path. The weaker multipath components are caused by the inertial movement of the motor mass – reverberation of the medium distorts the signal and multiple reflected/delayed replicas combine to create an elongated decaying response at the output. We measure the impulse response of our system using the *exponential sine-sweep* method [14] during which sinusoids of exponentially increasing frequency drives the motor. The output from the microphone is de-convolved with the weighted reverse sine-sweep to obtain the impulse response (the technique offers robustness against noise and non-linear distortions). Figure 9(a) and (b) show the measured impulse response and the corresponding *power delay profile* (PDP).

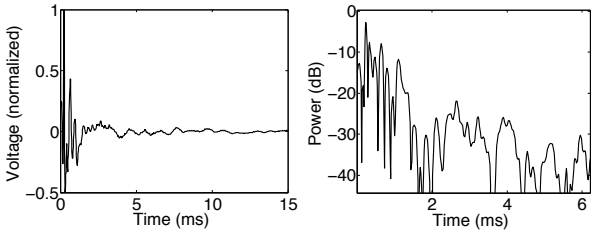

Figure 9: (a) Channel impulse response (b) Power delay profile of the vibratory channel.

Parameter Selection

Cyclic Prefix: The PDP shows 0.4ms before the multipath energy falls below 10dB of the highest peak, called "10DB maximum excess delay". This should be the separation between symbols to avoid inter symbol interference (ISI). We set the guard interval conservatively to 1ms, however, instead of leaving the channel idle during this interval, we insert 1ms of the last part of the symbol. This is called the cyclic prefix (CP) which helps cope with time synchronization errors without affecting the orthogonality of sub-carriers.

Subcarrier Bandwidth: The vibratory channel, as mentioned earlier, offers long channel coherence time, allowing for small subcarrier spacing. In practice, however, due to unpredictable phase noise, the inter carrier interference (ICI) becomes severe with small subcarrier spacing. On the other hand, the subcarriers become frequency selective for bandwidths larger than the coherence bandwidth of the channel. In such cases, the channel is no longer flat and hence equalization techniques falter [13, 35]. We measure the coherence bandwidth to be 480Hz (see Figure 10) – this is the width of the frequency-correlation function using a threshold of 0.95. We then choose the subcarrier bandwidth conservatively to 40Hz, less than the $\frac{1}{10}^{th}$ of the coherence bandwidth.

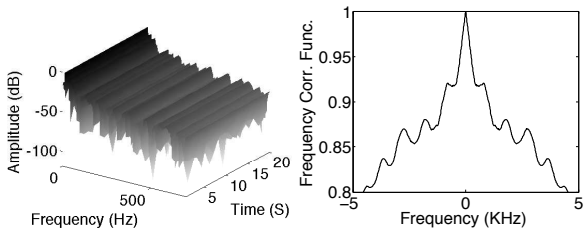

Figure 10: (a) Temporal stability of the channel, (b) The frequency-correlation function indicates the coherence bandwidth of 480Hz for the width threshold of 0.95.

Total Bandwidth: We choose the total bandwidth to be 12KHz, equal to the coherence bandwidth at correlation threshold of 0.7.

With this PHY layer in place, we now focus on a vibratory MAC layer, with the goal of reliably delivering packets to the receiver even under interference.

4 MAC Layer Design

Reliable packet delivery entails retransmitting a packet when it is received in error. In wireless systems, since the transmitter is unaware of the receiver's channel conditions, the error detection happens reactively, through an ACK from the receiver. Vibra-motors offer a new opportunity – we find that the receiver's interference conditions can be sensed at the transmitter through what is known as *back EMF*. Thus, the transmitter could potentially transmit and listen at the same time, infer symbol collisions, and retransmit symbols proactively. Efficiency can improve but some issues need mitigation.

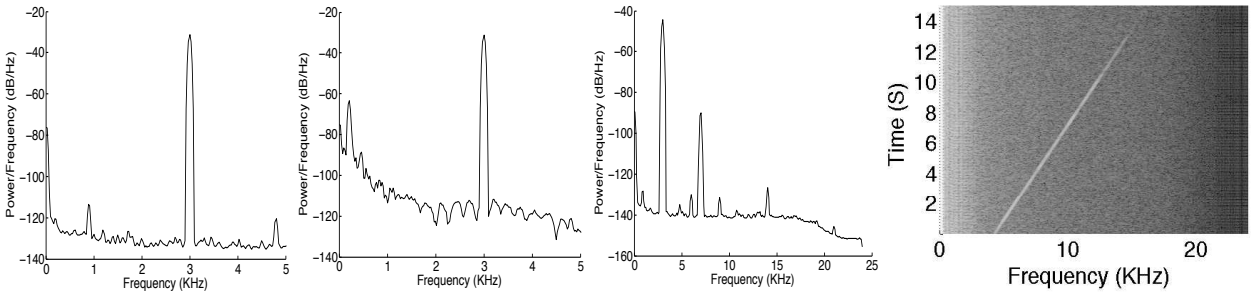

Figure 11: (a) Vibra-motor driven by a 3KHz voltage and no interference in the environment. (b) Interference introduced in the environment raises the noise floor, especially at lower frequency bands. (c) Clear detection of 7KHz interference caused by a nearby vibrator. (d) Spectrogram of acoustic chirp detected through back-EMF – the chirp was played through a speaker placed 4 feet away.

4.1 Sensing Interference from Back-EMF

Back-EMF is an electro-magnetic effect observed in magnet-based motors where relative motion occurs between the current carrying armature/coil and the magnetic field. In our vibra-motor, when the permanent magnet oscillates near the coil, the flux linkage with the coil changes due to the driving voltage and/or vibration noise. According to the Faraday's law of electromagnetic induction [16], this changing flux induces an electromotive force in the coil. Lenz's law [39] says this electromotive force acts in the reverse direction of the driving voltage, called *back-EMF of the motor*. As the rate of change of the magnetic flux is proportional to the speed of the magnetic mass, the back-EMF serves as an indicator of the extraneous vibration experienced by the mass.

Unsurprisingly, the interfering vibrations generate subtle movements of the vibra-motor mass, causing the voltage changes around a small resistor to be in milli-volts (below the ADC noise floor)[3]. We design a low noise amplifier, limiting the parasitic inductance/capacitance, to amplify this voltage $100x$ before feeding it to the ADC sampling circuit. Figures 11(a,b) show the difference between interference-free and interfered transmissions, as sensed through back-EMF. The noise floor increases, especially at lower frequencies where the interference is dominant. Figure 11(c) shows another case where a 7KHz interferer – a second interfering vibra-motor – is placed on the same table as our experiment; the transmitting vibra-motor detects the corresponding spike at 7KHz. We also played an acoustic chirp on a speaker 4 feet away from our devices – Figure 11(d) shows the chirp spectrogram, a reasonable reproduction of the actual. The findings extend hope that back-EMF can be useful to designing transmitter-side collision inference protocols.

[3]The measuring circuit samples the induced current as a voltage drop across a series resistor. We keep this resistor value below 0.02% of the motor's coil resistance so that the electrical property of the system remains unaffected.

4.2 Vibratory Interference

Before moving into protocol design, we characterize the nature of vibratory interference experienced by the microphone. Interferences are broadly of two kinds. (1) Ambient acoustic sounds, such as people talking, background music, machine hums, etc. and (2) physical vibrations caused by objects such as running table fans, taps and thuds on table-tops, and even natural vibration of human hands when they are holding the devices. Figure 12 shows the spectral graph of several example interferences, measured in isolation. The key observation is that interferences are heavily biased to the lower frequency bands; frequencies higher than 6KHz are rarely impacted.

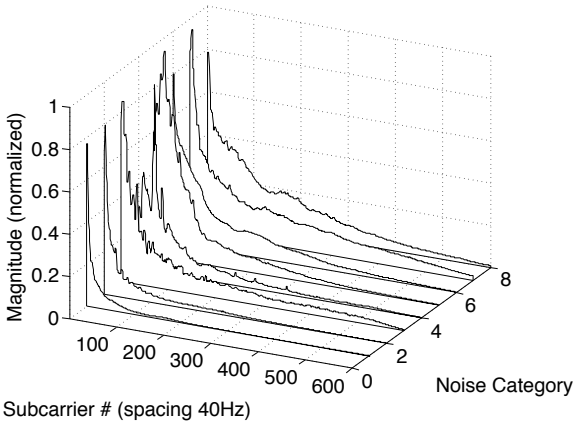

Figure 12: Spectral properties of various interferences occurring in the natural environment.

Figure 13 shows the 3D contour of acoustic interference across frequency and time – the interference stems from loud human voices. The key observation is that for any given frequency, the signal amplitude of the interference rises with time, reaches a peak, and decays again. This characteristic is highly common in a wide range of interferences, primarily because instantaneously starting or stopping strong signals is difficult. Occasionally, we find

certain machines capable of producing a sudden spike, however, their decay is still slow. We leverage back-EMF along with these properties of the interference to design a MAC protocol, called *Proactive Symbol Recovery* (PSR).

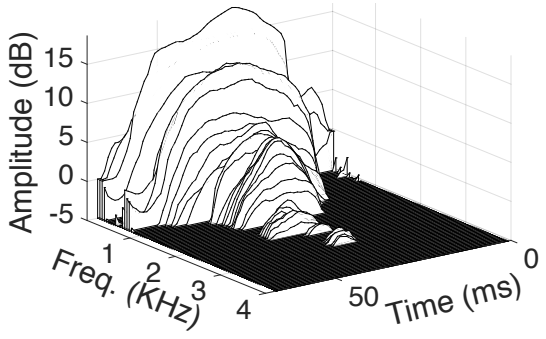

Figure 13: 3D contour of acoustic interference across frequency and time.

4.3 PSR Protocol: The Problem Definition

The protocol problem can be abstracted as follows. Consider a packet P composed of many OFDM symbols, $[S_1, S_2, S_3, ...]$, each symbol composed of n subcarriers $[f_1, f_2, f_3, ...f_n]$. Figure 14 shows the pictorial representation of such a packet, in the form of a time-frequency grid. Assume that the gray region denotes the incidence of interference, essentially the top view of Figure 13. Now, with back-EMF, the transmitter is able to sense receiver-side interference, however, the sensing is not accurate. To be able to reliably detect interference (i.e., reduce false positives), the transmitter can increase the sensing threshold – interference detected above this threshold is strongly indicative of actual interference. Assume that the interference above a given threshold is the black region in Figure 14.

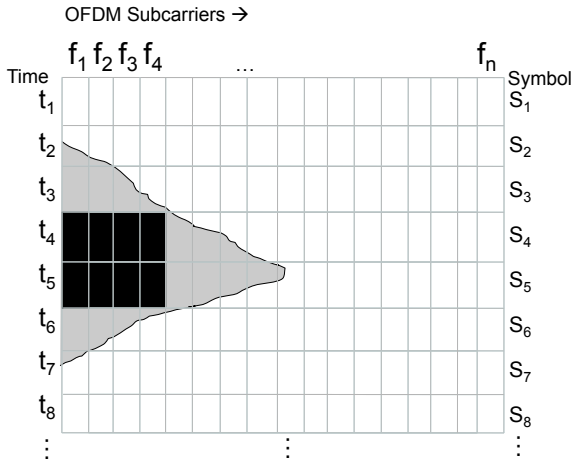

Figure 14: A packet represented in terms of OFDM symbol, each symbol to be transmitted over time.

The protocol question is: *which symbols should the transmitter retransmit, and when?* Transmitting only the symbols that are affected by the black color may still leave too many erroneous symbols – the coding scheme at the receiver may not be able to recover the packet. The transmitter essentially needs to estimate the symbols affected by the gray region too, and retransmit a subset of those symbols [23, 19]. Clearly, not all the gray-color affected symbols need to be transmitted since the coding scheme can indeed correct for some errors.

A second question pertains to interference adaptation. Once interference is detected at time t_4, the transmitter must adjust the subsequent transmissions to cope with the interference. Any adjustments – such as rate control – would need to be communicated to the receiver through some control information. However, unlike packets, symbols are not prefaced with headers; dedicating some subcarriers to a control channel will be wasteful in general. Under this constraint, the protocol needs to adapt to interference and concisely convey its adaptations to the receiver. The basic problem is new to the best of our knowledge, since existing protocols assume that the receiver has better estimates of error than the transmitter [23, 19]. In our case, the transmitter is better aware of the interference but has no control bits to convey its adaptations.

4.4 Proactive Symbol Recovery Protocol

The PSR protocol develops 2 heuristics – interference extrapolation and implicit control signaling – described next.

(1) Interference Extrapolation. Only the contour of the interference within the black region (in Figure 14) is visible to the transmitter – one could metaphorically envision it as the "part of the iceberg above water". Based on the visible shape, the transmitter may be able to extrapolate the "submerged" shape, generating an estimate of the gray region. Our measurements have consistently indicated that the interference decay is well-behaved, of course with some jitter. Hence, we model this as a curve fitting problem, and use a $3rd$ order cubic spline (the high frequency jitters are not captured). Given multiple silhouettes, one per-subcarrier, we pick the silhouette whose peak is at 80^{th} percentile among all peaks. Using this we develop an estimate of the gray region.

(2) Implicit Control Signaling. As mentioned earlier, the transmitter needs some control bits for signaling its actions to the receiver. To this end, we use a simple interleaving idea from the basics of signal processing. Specifically, when alternate subcarriers are loaded with data (and the ones in-between left empty), the time domain representation of the OFDM signal exhibits two identical copies (Figure 15). We call this the $2x$ interleaving

mode. When every $4th$ subcarrier is loaded, the time domain signal shows 4 identical copies of the same signal. The receiver recognizes these identical copies in time domain and decodes the control information. In frequency domain, it extracts the data from every $2nd$ (or $4th$) subcarrier and ignores the others. Of course, we are aware that the control bits are not free – the $2x$ and $4x$ interleaving modes reduce the bandwidth. However, we also note that energy on the loaded subcarriers increases – a $2x$ mode exhibits a 3dB gain (nearly double), lowering chances of demodulation error.

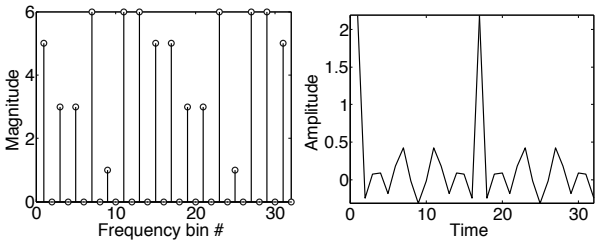

Figure 15: (a) 2x interleaving in frequency, (b) identical signal parts in time domain.

Protocol Design: We now describe the basic operation of the PSR protocol (we continue to refer to the toy example in Figure 14). When no interference is detected by the transmitter's back-EMF sensor (i.e., until time t_4), symbols are sent as usual. Upon detecting interference at t_4, the transmitter records the symbol that was affected (namely, S_4), and performs the subsequent symbol transmissions (S_5) at $2x$ interleaving mode. This continues until the interference has subsided below the transmitter's threshold. At this point, the transmitter performs the extrapolation using the interference decay data, starting from the last-observed interference peak. The interpolation suggests that the receiver may continue to experience interference until some time in the future, say till t_7. Therefore, the transmitter continues symbol transmissions in $2x$ mode, after which it falls back to no-interleaving. Observe that this interleaving mechanism is akin to halving the rate, except that it helps inform the receiver about the rate reduction.

Ideally, the interference extrapolation may help recover the symbols S_6 and S_7, however, symbols S_2 and S_3 could also be heavily interfered. To this end, the transmitter also extrapolates the front portion of the interference, and remembers the symbols that need retransmission. Once all the symbols have been transmitted, it now retransmits these symbols (S_2, S_3, and S_4 in this toy case), at the appropriate interleaving mode permissible by the then channel conditions. Importantly, the receiver must identify that these symbols are actually duplicates of prior symbols. Hence, the transmitter marks the start of these retransmissions with a $4x$ interleaved packet – the packet includes indices of all symbols that are being retransmitted. The encoding of indices is efficiently done to utilize the fewest bits possible, telling the receiver how many retransmissions to expect and which prior symbols to replace. The receiver demodulates all the symbols, performs the appropriate replacements, and feeds them through the decoder.

Coding for Error Correction: Needless to say, extrapolation will incur errors, and back-EMF sensing will experience false negatives. This will leave erroneous symbols at the receiver even after retransmissions. In fact, it would be inefficient for the transmitter to recover all symbols since the decoder at the receiver would be able to correct for some of them anyway. We implement a standard 2/3 convolutional code, with constrain length 7, to cope with inherent symbol errors in the transmission. We implement a hard decision Viterbi decoder with trace back depth of 30 to recover the bits. To cope with heavy bursts in error, we use an interleaver to spread out the bursts.

5 System Evaluation

5.1 Complete Hardware Prototype

Figure 16 shows the complete interconnection of the hardware elements in Ripple II. Very briefly, the receiver (on the left side) draws power from the USB port of a Dell laptop (or any mobile device or raspberry-pi/arduino) serving as the controller. Instead of using a separate ADC, we abuse the *Line-in audio input port* of the laptop, which comes equipped with a high speed ADC and a driver to push samples to user space. We connect signals from each microphone to one of the channels in the line-in port with the help of a three-conductor (TRS) audio jack. We run the appropriate driver to sample the signal at 48KHz, 16bit stereo mode.

The transmitter (shown on the right side) also uses a similar approach. The software controller generates digital samples that are converted to analog via the DAC of the audio port. This output signal (with appropriate amplification and shaping) feeds into the vibra-motor, which is in turn attached to the stylus or ring. We sample this line-in port at 48KHz to collect the back-emf signal along with the reference voltage. Offline processing is performed in MATLAB; realtime music streaming is performed on GNURadio.

5.2 Performance Results

We present end to end results first, followed by zoomed in results from acoustic noise cancellation (SANF) and proactive symbol retransmission (PSR). Our final results

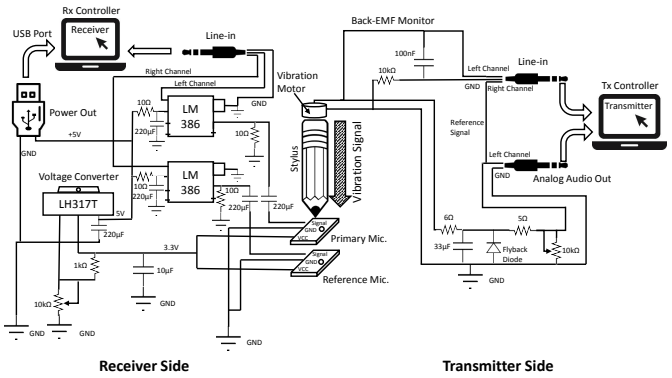

Figure 16: The complete hardware internals of Ripple II.

are drawn from 100+ sessions of experiments, each session either $1-3$ min. long, and entails vibratory transmission against diverse ambient sounds, ambient vibrations, modulations, etc. We collected 800 samples of ambient sound (e.g., supermarket ambience, in classroom noise, music nearby, etc.) and 15 ambient vibrations (e.g., walking, moving in a car, tapping on table). Half of sessions were against the natural lab sound conditions; for the other half, we played external ambience sounds through a speaker and generated vibrational noise through an external vibra-motor placed on the table. As a baseline we use the basic OFDM microphone receiver running on our hardware platform (including the covered sound hole). We compare this baseline against (1) baseline + coding, (2) baseline + coding + SANF, and (3) baseline + coding + SANF + PSR.

Ripple II Results:
Figure 17(a) shows the CDF of throughput gain computed from all the experimentation data, across all possible noise environments. The communication link operates in a high bit error rate (BER) regime and coding schemes perform worse than expected. The median gain with SANF is around 10%, with a small fraction of cases leading to negative gain. However, PSR brings appreciable benefits, mainly from retransmitting erroneous symbols and bringing the errors below the tolerable threshold. Median throughput gain with PSR is 26.6%. Figure 17(b) reports the breakup of raw throughput under various ambient sound categories. Under mechanical sound spikes alone, the performances of SANF and PSR are weak – the interpolation in PSR falters, while SANF's symbol error detection scheme is not sensitive enough. However, in other categories of noises, throughput improves – the median throughput in the "All" noise category is ≈ 27 Kbps.

SANF Results:
Figure 18(a) zooms into symbol selective aspect of SANF, and shows the fraction of symbols corrected over normal adaptive noise filtering. The correction gain improves with higher SNR, but falls beyond 15dB. This is because at $>$ 15dB SNR, SANF is unable to detect the symbol errors correctly since the interference is less pronounced – the inability to identify the erroneous symbols derails adaptive noise filtering. The sensitivity curve captures this behavior, suggesting that the symbol correction efficacy is both a function of SNR and sensitivity. Figure 18(b) shows the gain across each subcarrier – the graph is for the best SNR, 15dB.

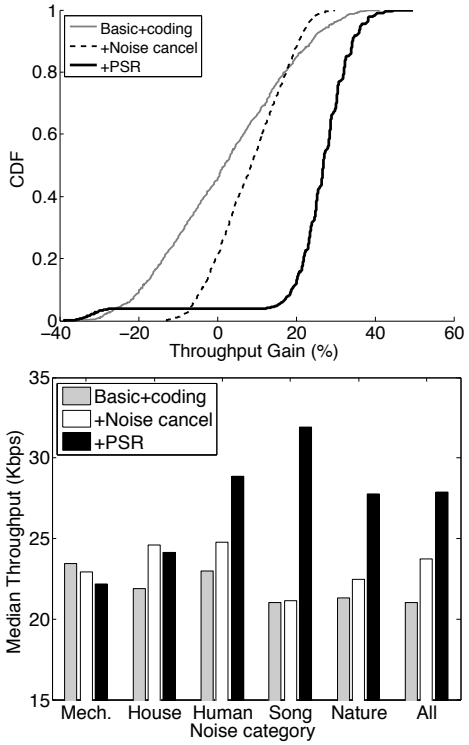

Figure 17: (a) Throughput gain across all experiments. (b) Median throughput across different ambient sound categories.

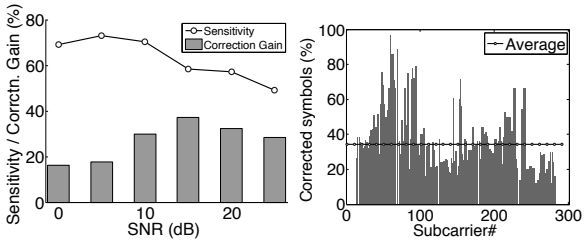

Figure 18: (a) Variation of SANF's cancellation gain and sensitivity against increasing SNR; sensitivity is the fraction of erroneous symbols detected by SANF. (b) The noise cancellation gain as the percentage of erroneous symbol per subcarrier.

PSR Results:
The core design elements in PSR pertains to (1) back-EMF based sensing and extrapolation of the interference,

and (2) reducing symbol errors via $2x$/$4x$ interleaving (expected to increase energy). To evaluate extrapolation, we first identify the set of truly erroneous symbols that should have been retransmitted by the transmitter. We know the set of symbols that PSR actually retransmitted. From these two sets, we compute the precision and recall of PSR, reflecting the combined efficacy of back-EMF sensing and interpolation. Figure 19(a) shows the results – the precision is strong but the recall is weak, indicating that PSR is conservative. This is expected/desirable since we intend to not retransmit excessively, which reduces inflation of the packet and also allows the decoder to correct for the residual errors. Of course, there is room to tune the interpolation scheme and the back-EMF sensitivity – we leave this to future work.

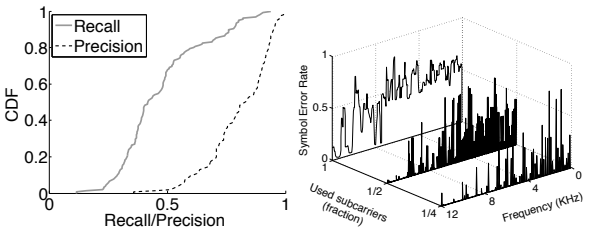

Figure 19: (a) Precision and recall to evaluate the back-EMF sensing and interference extrapolation scheme. (b) The per subcarrier symbol error rates using all, 1/2, and 1/4 of the subcarriers, while the noise power is constant.

Figure 19(b) shows the reduction in symbol error rate when half and one-forth subcarriers are loaded with data (recall we denoted this as $2x$ and $4x$ modes of transmission). Under heavy channel interference, $2x$ mode substantially reduces symbol errors, offering effects similar to rate control. However, the $2x$ mode also implicitly includes a control bit that the receiver can recognize. Measurements show that the control signaling was near perfect, meaning the receiver almost always extracted the correct data from $2x$ and $4x$ transmissions.

5.3 Applications and Capabilities

We explore potential applications of Ripple II, namely a vibratory ring and watch; tabletop communication; and device to device transfers.

(1) Finger Ring for Authentication

We envision touch based two-factor authentication – a user wearing a Ripple II ring or watch could touch the smartphone screen and the vibratory password can be conducted through the bones. The core notion generalizes to other scenarios, including unlocking car doors, door knobs, etc. While a usable system would need maturity in interfaces, energy, etc., this section only discusses the communication aspects of through-bone transmission. Figure 20(a) shows the crude finger ring prototype, placed on the index finger of the user. For our prototype, the ring is powered by a battery located outside the ring and connected via long wires. The cylindrical vibra-motor is placed horizontally on the finger to maximize area of contact, however, placement influences communication.

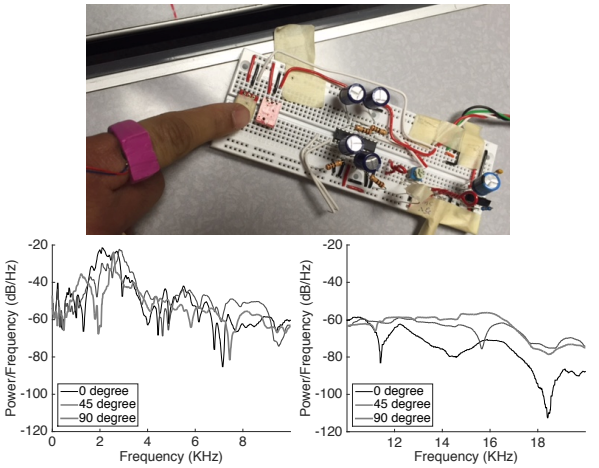

Figure 20: (a) Finger ring operated at 8KHz. (b) Incidence angles affect lower frequencies less. (c) Higher frequencies in a piston oscillator become directional and hence delivers less energy in unaligned directions.

Figure 20(b,c) shows the variation of signal power for 3 different incidence angles between the vibra-motor and the finger – incidence angle defined as the angle between the finger bone and the direction along which the vibrator mass oscillates (which is perpendicular to the base of the cylinder). Evidently, at lower frequencies, the incidence angle does not impact the signal, however, at higher frequencies the higher incidence angles reduce SNR. Moreover, higher frequencies are also less effective for signal propagation through the human body. Thus, we decide to operate the ring at 90° incidence but focus the power budget to within 8KHz.

We also performed similar experiments with a watch – pasting the vibra-motor on the wrist-bone below the watch. Performance degrades as expected, due to a longer conduction path from the wrist to the microphone. The table below summarizes results. 5 student volunteers experimented with our prototype and none of them were able to feel or hear the vibrations at all.

	Bandwidth	Modu.	Code	Tput:Kbps
Ring	8 KHz	QPSK	1/2	7.41
Watch	3 KHz	QPSK	1/2	2.23

5.4 Tabletop Communication

Multicast communication is often useful – a group picture at a restaurant needs to be shared with everyone

in the group; presentation slides need to be shared in a meeting. We envision placing all phones on the table, near each other, and performing one vibratory multicast. Figure 21 shows the outcome of such an experiment – we used the stylus to touch different locations on a table, while 2 microphone receivers were at fixed locations on this otherwise empty table. Even at nearly 2 feet away, the throughput is around 4Kbps (the X-axis has duplicate values since there were multiple distinct locations at the same distance from the microphone).

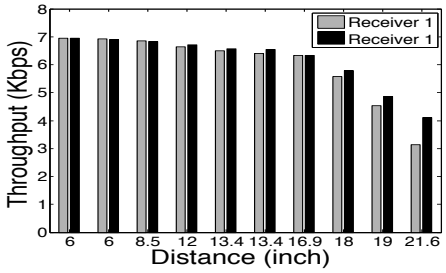

Figure 21: Throughput against varying tabletop range.

5.5 P2P Money Transfer

In developing regions, mobile payments may be viable with basic phones with vibra-motor and microphones. Perhaps a USB stick can transfer data to phones/tablets on physical contact. Such apps obviously need higher data rates and some may require real time operation. The table below shows possibilities when vibra-motor on stylus' and smartphones are touched to microphones. We have also built a demonstration of a real time music streaming system over vibrations – please see video demo here: `http://synrg.csl.illinois.edu/ripple/`

	Bandwidth	Modu.	Code	Tput:Kbps
Stylus	12 KHz	16 QAM	2/3	29.19
Phone	12 KHz	16 QAM	2/3	26.13

6 Related Work

Vibratory communication: Authors in [38] and [22] were the first to conceive the idea of communicating through physical vibrations. They both encode vibrations through (ON/OFF) Morse code, with pulse durations of around one second (i.e., 1 bits/s). This is adequate for applications like secure pairing between two smart phones, or sending a tiny URL over tens of seconds. Our prior work in NSDI 2015 [34] developed a fuller vibratory radio through multi-frequency modulation, self-jamming based security, and resonance braking, ultimately translating to 200 bits/s. Ripple II is a push-forward of the Ripple project, but with microphone as the receiver, and augmented with a new PHY/MAC layer offering 150x throughput gain. Ripple II still preserves Ripple's security properties via self-sound cancellation.

Dhwani [31] and Chirp [2] address conceptually similar problems, although on the acoustic platform; vibra-motors bring about new set of challenges and opportunities. Technologies like Bump [1, 28, 36, 9, 20, 18, 26] use accelerometer/vibrator-motor responses to facilitate secure pairing between devices. TagTile [4] uses high frequency sound to achieve association between phones and point-of-sale devices. However, these techniques are primarily designed for few bits of exchange; Ripple II aims high bitrate transmission with the same ease as Bump and Tagtile. Further, as indicated by researchers [38, 17], the lack of the dynamic secret message in Bump-like techniques makes them less secure in the wild. These modes also require Internet connectivity and trusted third party servers to function, none of which is needed in Ripple II.

Vibration generation and sensing: Creative research in the domain of haptic feedback has investigated the state-of-the-art in electro-mechanical vibrations [32, 10]. Applications in assisted learning, touch-augmented environments, and haptic learning have used vibrations for communication to humans [30, 15, 21, 32, 10]. However, the push for high communication data rates between vibrators and microphones/accelerometers is unexplored to the best of our knowledge. Off late, personal/environment sensing on mobile devices has gained research attention. Applications like (sp)iPhone [27] and TapPrints[29] demonstrate the ability to infer keystrokes through background motion sensing. While many more efforts are around activity recognition from vibration signatures, this paper aims to modulate vibration for communication.

7 Conclusion

Ripple II is an attempt to enable touch-based vibratory communication between a vibra-motor and a microphone. We develop a vibratory radio at the PHY and MAC layer, and explore a few possible applications in authentication, device to device streaming, and tabletop communication. While additional work is needed to attain maturity, we believe this paper is a concrete step towards demonstrating an alternative communication mode, that has remained relatively unexplored in the past.

Acknowledgement

We sincerely thank our shepherd Dr. Jonathan Smith and the anonymous reviewers for their valuable feedback. We are also thankful to Prof. Pramod Viswanath and Shaileshh Venkatakrishnan for the various discussions on this topic. We are grateful to Qualcomm, Huawei, and NSF (grant CNS-1430033) for partially funding this research.

References

[1] Bump Technologies. `http://blog.bu.mp`. Accessed: 7th February, 2016.

[2] Chirp. `http://www.chirp.io`. Accessed: 7th February, 2016.

[3] Ripple webpage. `http://synrg.csl.illinois.edu/ripple/`. Accessed: 7th February, 2016.

[4] Tagtile report. `http://www.mybanktracker.com/news/2011/10/07/tagtile-easier-reward/`. Accessed: 24th September, 2015.

[5] ADKINS, J., FLASPOHLER, G., AND DUTTA, P. Ving: Bootstrapping the Desktop Area Network with a Vibratory Ping. *Ann Arbor 1001* (2015), 48109.

[6] AKYILDIZ, I. F., POMPILI, D., AND MELODIA, T. Underwater acoustic sensor networks: research challenges. *Ad hoc networks 3*, 3 (2005), 257–279.

[7] BURDIC, W. S. *Underwater acoustic system analysis*. Prentice Hall, 1991.

[8] CANDÈS, E. J., LI, X., MA, Y., AND WRIGHT, J. Robust principal component analysis? *Journal of the ACM (JACM) 58*, 3 (2011), 11.

[9] CASTELLUCCIA, C., AND MUTAF, P. Shake them up!: a movement-based pairing protocol for cpu-constrained devices. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services* (2005), ACM, pp. 51–64.

[10] CHO, Y.-J., YAND, T., AND KWON, D.-S. A New Miniature Smart Actuator based on Piezoelectric material and Solenoid for Mobile Devices. In *The 5th International Conference on the Advanced Mechatronics, ICAM* (2010), pp. 615–620.

[11] COATES, R. F. *Underwater acoustic systems*. Halsted Press, 1989.

[12] DEBBAH, M. Short introduction to OFDM. *White Paper, Mobile Communications Group, Institut Eurecom* (2004).

[13] ENGELS, M., AND PETRÉ, F. *Broadband fixed wireless access: a system perspective*. Springer Science & Business Media, 2006.

[14] FARINA, A. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *Audio Engineering Society Convention 108* (2000), Audio Engineering Society.

[15] FEYGIN, D., KEEHNER, M., AND TENDICK, F. Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on* (2002), IEEE, pp. 40–47.

[16] GALILI, I., KAPLAN, D., AND LEHAVI, Y. Teaching Faradays law of electromagnetic induction in an introductory physics course. *American journal of physics 74*, 4 (2006), 337–343.

[17] HALEVI, T., AND SAXENA, N. On pairing constrained wireless devices based on secrecy of auxiliary channels: The case of acoustic eavesdropping. In *Proceedings of the 17th ACM conference on Computer and communications security* (2010), ACM, pp. 97–108.

[18] HALPERIN, D., HEYDT-BENJAMIN, T. S., RANSFORD, B., CLARK, S. S., DEFEND, B., MORGAN, W., FU, K., KOHNO, T., AND MAISEL, W. H. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on* (2008), IEEE, pp. 129–142.

[19] HAN, B., SCHULMAN, A., GRINGOLI, F., SPRING, N., BHATTACHARJEE, B., NAVA, L., JI, L., LEE, S., AND MILLER, R. R. Maranello: Practical Partial Packet Recovery for 802.11. In *NSDI* (2010), pp. 205–218.

[20] HOLMQUIST, L. E., MATTERN, F., SCHIELE, B., ALAHUHTA, P., BEIGL, M., AND GELLERSEN, H.-W. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Ubicomp 2001: Ubiquitous Computing* (2001), Springer, pp. 116–122.

[21] HUANG, K., DO, E.-L., AND STARNER, T. PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on* (2008), IEEE, pp. 41–44.

[22] HWANG, I., CHO, J., AND OH, S. Privacy-aware communication for smartphones using vibration. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on* (2012), IEEE, pp. 447–452.

[23] JAMIESON, K., AND BALAKRISHNAN, H. PPR: Partial packet recovery for wireless networks. *ACM SIGCOMM Computer Communication Review 37*, 4 (2007), 409–420.

[24] KAAJAKARI, V., ET AL. Practical MEMS: Design of microsystems, accelerometers, gyroscopes, RF MEMS, optical MEMS, and microfluidic systems. *Las Vegas, NV: Small Gear Publishing* (2009).

[25] KRAWCZYK, M., AND GERKMANN, T. STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22*, 12 (2014), 1931–1940.

[26] LESTER, J., HANNAFORD, B., AND BORRIELLO, G. Are You with Me?–Using Accelerometers to Determine If Two Devices Are Carried by the Same Person. In *Pervasive computing*. Springer, 2004, pp. 33–50.

[27] MARQUARDT, P., VERMA, A., CARTER, H., AND TRAYNOR, P. (sp) iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM conference on Computer and communications security* (2011), ACM, pp. 551–562.

[28] MAYRHOFER, R., AND GELLERSEN, H. Shake well before use: Intuitive and secure pairing of mobile devices. *Mobile Computing, IEEE Transactions on 8*, 6 (2009), 792–806.

[29] MILUZZO, E., VARSHAVSKY, A., BALAKRISHNAN, S., AND CHOUDHURY, R. R. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (2012), ACM, pp. 323–336.

[30] MORRIS, D., TAN, H. Z., BARBAGLI, F., CHANG, T., AND SALISBURY, K. Haptic feedback enhances force skill learning. In *WHC* (2007), vol. 7, pp. 21–26.

[31] NANDAKUMAR, R., CHINTALAPUDI, K. K., PADMANABHAN, V., AND VENKATESAN, R. Dhwani: secure peer-to-peer acoustic NFC. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (2013), ACM, pp. 63–74.

[32] NIWA, M., YANAGIDA, Y., NOMA, H., HOSAKA, K., AND KUME, Y. Vibrotactile apparent movement by DC motors and voice-coil tactors. In *Proceedings of the 14th International Conference on Artificial Reality and Telexistence (ICAT)* (2004), pp. 126–131.

[33] ROBERGE, J. K. *Operational amplifiers: theory and practice*. John Wiley & Sons, 1975.

[34] ROY, N., GOWDA, M., AND CHOUDHURY, R. R. Ripple: Communicating through physical vibration. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)* (2015), pp. 265–278.

[35] RUMNEY, M., ET AL. *LTE and the evolution to 4G wireless: Design and measurement challenges*. John Wiley & Sons, 2013.

[36] SAXENA, N., AND WATT, J. H. Authentication technologies for the blind or visually impaired. In *Proceedings of the USENIX Workshop on Hot Topics in Security (HotSec)* (2009), vol. 9, p. 130.

[37] SMARAGDIS, P., RAJ, B., AND SHASHANKA, M. Missing data imputation for spectral audio signals. In *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on* (2009), IEEE, pp. 1–6.

[38] STUDER, A., PASSARO, T., AND BAUER, L. Don't bump, shake on it: The exploitation of a popular accelerometer-based smart phone exchange and its secure replacement. In *Proceedings of the 27th Annual Computer Security Applications Conference* (2011), ACM, pp. 333–342.

[39] TANNER, P., LOEBACH, J., COOK, J., AND HALLEN, H. A pulsed jumping ring apparatus for demonstration of Lenzs law. *American Journal of Physics 69*, 8 (2001), 911–916.

[40] TSE, D., AND VISWANATH, P. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[41] WAITE, A. D., AND WAITE, A. *Sonar for practising engineers*, vol. 3. Wiley London, 2002.