Data Over Sound

Sergey Philippov, Alexander Sheptovetsky

Lapka, Inc., November 2013

Abstract

When we talk, we transmit data using sound waves. We encode information using specific sounds associated with each letter, where a *sound* is a short combination of frequencies with some additional acoustic parameters. The same approach is used in wireless technology, where information is transmitted using radio waves. What would it take to transpose radio-based near-field communication into the sound-over-air domain?

This paper discusses the details, advantages, and challenges of designing and implementing a data-over-sound protocol for the near-field device-to-device communication.

Radio

Radio is a wireless transmission of information through space by systematically changing some property of the radio waves, such as their amplitude, frequency, or phase.

Sound

In physics, the sound is a vibration that propagates as a typically audible mechanical wave of pressure and displacement, through a medium such as air.

When people talk, they use specific sounds to encode letters — this is how they transmit information over the air. By nature, a specific sound is a short sequence of frequencies; therefore, talking is essentially a frequency-modulation process.

The radio world is very familiar with this concept, where it is called Frequency-Shift Keying (FSK) and follows an identical concept — assigning specific frequencies to specific "letters" and transmitting information by modulating these frequencies in the signal.

Hardware

If we want to keep drawing an analogy, smartphones would be the right match for a human today. Among other hardware devices, they became most popular, smart, adaptive, and predisposed for evolution.

To communicate with each other, smartphones mostly use radio-based protocols: GSM, LTE, WiFi, NFC, Bluetooth. However, to communicate with people, they use regular human media-domains: visual and sound. That means they already have the hardware required to transmit and receive information in these domains. In the case of the sound domain, that would be speaker and microphone.

So in theory, we could use existing hardware to make devices talk to each other in a sound domain, almost the same way they talk and listen to people.

Regulations and Standards

All conventional radio protocols are highly regulated and standardized. They were designed to work in all conditions, suit all needs, and conform to all requirements (security, etc.). Therefore, using the same protocol one could make police-grade devices and connected toys. Such a unified approach is better for regulations and chip makers, but when it comes to simple use cases (just to make one device talk to another, no extra security or pairing) it becomes overwhelming both from an implementation perspective and with regards to user experience.

On the other hand, the sound domain is merely a subject of regulations, with only some safety limits (volume) and human capabilities (frequency range of audible sound). In other words, one can simply talk.

Inaudible

Based on human capabilities, the sound is divided into infrasound, audible, and ultrasound, where the border between audible and ultrasound is 20 kHz.

However, most people can hear sounds just up to 18 kHz. So there is a range that technically is audible but in fact, is not. We call it inaudible sound.

What is special about inaudible sound is that most of the existing audio hardware (microphones and speakers) still include it into the operational range, even though people can't hear it. That makes it perfect for our needs — to transmit data over sound using existing hardware without intrusion into the people medium.

Alphabet

To make devices talk and understand each other, we must define a common alphabet that will be used to encode and decode data.

Modulation

First, we have to choose the modulation we are going to build our alphabet on. In telecommunications, modulation is the process of varying one or more properties of a carrier signal.

The most fundamental digital modulation techniques are based on keying: PSK, FSK, ASK (phase, frequency, and amplitude-shift keying) — where a finite number of phases, frequencies, or amplitudes is used. Bluetooth, for example, uses phase-shift keying to exchange information between devices.

When talking, people use frequency-shift keying to encode information; so we will do the same.

Frequency-Shift Keying

One of the aspects we want to achieve is a certain speed of transmission. Taking into account the physical limits of the sound domain, we cannot transmit faster than a certain amount of symbols per second. Therefore, we have to make each symbol carry as much information as possible in order to increase the transmission speed.

By increasing the number of frequencies, we can increase the information density of the symbol. For example, using two frequencies will enable us to encode two values per symbol; by increasing the number of frequencies to 16, we will bring the density of information up to 16 *bits* of information per symbol.

FFT

What else can have an impact on our choice of the alphabet? The ability to receive and understand data encoded in sound.

Receiving is relatively easy, most microphones have a frequency response up to 24 kHz which includes all audible sound. Understanding is more complex: to retrieve specific frequencies from the received signal, one usually uses Fourier Transformation. When it comes to real-time signal processing on mobile devices, efficiency is extremely important, so FFT (Fast Fourier Transformation) will be the best variation of the algorithm. One of the features of FFT is that one can calculate amplitudes for the number of frequencies with equal incremental step, where the lesser frequencies are faster to process. Therefore, the best set of frequencies can be calculated by running FFT with different settings.

Best choice

Based on experiments, the best choice for the alphabet will be Frequency-Shift Keying using 18 equally distributed frequencies in an 18–20 kHz range, to encode 16 digits and 2 markers. In this case, we will be able to transmit hexadecimal data one digit per transmitted symbol plus have separate service markers to mark different messages with only one symbol.

Distance versus Doppler

One of the main issues we ran into using data-over-sound protocol was the Doppler effect.

On the one hand, to improve the noise immunity of the signal over distance, one would want to narrow the acceptable dispersion range for each frequency. One the other hand, the narrower the acceptable dispersion range will be — the higher the chance that the Doppler effect will shift the frequency out of its range when a transmitter or a receiver is moving.

The solution for that would be to understand how fast a transmitter and a receiver might move, calculate the frequency dispersion range the Doppler effect will make, and use this range in the process of defining the alphabet.

Practical Implementation

This technology was developed and used in Lapka Breathalyzer (https://mylapka.com/bam/ceramic/) and proved to be a viable and competitive solution based on market success and customer experience.

Resources

- Steven Smith, The Scientist & Engineer’s Guide to Digital Signal Processing, California Tech Pub, 1997
- K. Daniel Wong, Fundamentals of Wireless Communication Engineering Technologies, Wiley, 2012
- William C. Elmore, Physics of Waves, Dover Publications, 1985
- iOS Developer Library, vDSP Programming Guide, Using Fourier Transforms, Apple Inc, 2013