SonicData*:* Broadcasting Data via Sound for Smartphones

Aditya Shekhar Nittala[1], Xing-Dong Yang[1], Ehud Sharlin[1], Scott Bateman, Saul Greenberg[1]

[1]University of Calgary
Calgary, AB, Canada, T2N 1N4
{anittala, xdyang, ehud, saul}@ucalgary.ca

[2]University of Prince Edward Island
Charlottetown, PEI, Canada, C1A 4P3
sbateman@upei.ca

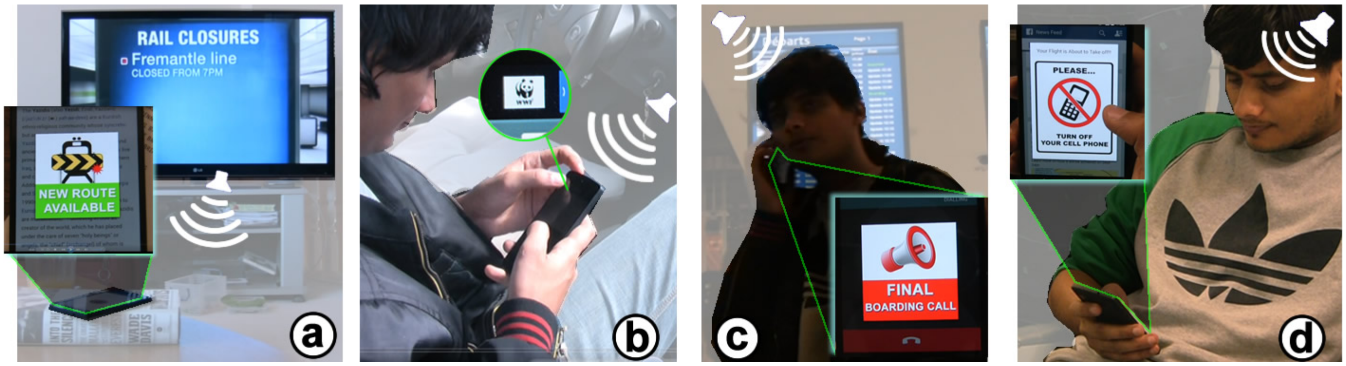

Figure 1. SonicData interaction techniques: (a) responding to traffic updates from a TV and suggesting a new route; (b) showing received charity advertisement on the status bar; (c) notifying a final boarding call; (d) suggesting powering off before taking off.

ABSTRACT

SonicData is a technique for broadcasting data to smartphones via audio streams using phone's built-in microphone. SonicData augments an audio stream in the environment with nearly inaudible high-frequencies, allowing data to be sent to any smartphone in the vicinity using regular speakers and without any need for special hardware and software infrastructure or handshaking requirements. We detail the technical implementation of the SonicData prototype, outline a technical evaluation of its capabilities, and describe the results of a preliminary study of its effect on the quality of sound streams. We designed four interaction techniques that highlight SonicData's potential as a complementary technique for broadcasting data to smartphones.

Author Keywords

Inaudible sound, mobile interfaces.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

INTRODUCTION

People are exposed to large amounts of audible information from many sources in their everyday lives, including TV, radio, online media, Muzak at shopping centers, and more. Because many people carry their smartphones with them in spaces where they encounter audible information, their smartphones are also exposed to these sounds. What if we could add hidden information-as-data to these sound channels? Could devices 'hear' the information in the background, and use it to create new possibilities for a broader range of *implicit* interactions [1]? To explore this space, we created SonicData, a *simplex* broadcasting technique that utilizes the ubiquity of audio streams already present in everyday environments. SonicData is a way of embedding data content into an existing audio stream, allowing nearby smartphones to sense and use the data in various meanings (e.g. modify interfaces). It does this by embedding meta-data into the unused high, and nearly inaudible, frequencies in audio streams. It can augment these streams either when the original audio stream is being created, or on-the-fly. Unlike other communication techniques – such as Wi-Fi, Bluetooth, or Near-field Communication (NFC) – SonicData uses a smartphone's built-in microphones and off-the-shelf speakers that are widely available in homes, offices, and public spaces. Furthermore, unlike other technologies, SonicData does not need handshaking protocols and specialized hardware or software infrastructures.

In this paper, we outline the details of how SonicData works. We present findings of a study that shows embedding data in high-frequency sound is a feasible way of transmitting data to a smartphone, and that its effect on the audible quality of the audio stream is negligible. We also demonstrate the potential of SonicData with four prototype systems, all of which use commonly available speakers to provide important data to mobile applications.

RELATED WORK

There is a long history of encoding information in sound. Both humans and animals, commonly use sound to send information to one another; through speech, whistling or drumming (e.g. [7]). In the digital realm, digital data is

Cite as: Nittala, A., Yang, X-D., Sharlin, E., Bateman, S., and Greenberg, S. (2014) SonicData: Broadcasting Data via Sound for Smartphones. Research Report 2014-1064-15, Department of Computer Science, University of Calgary, Calgary, AB, Canada T2N 1N4.

encoded into sound by modulating sound wave properties, such as its frequency, amplitude, or phase [2, 4, 6]. The use of sound for interactive tasks is also well explored. Most rely on matching a heard sound to a known sound, e.g., via audio classification machine learning techniques [3] or via a database search of pre-sampled audio streams [8]. While these methods are technically robust within certain application domains (e.g. identifying songs or voice recognition), they are based on a-priori assumptions on what sounds are being heard (e.g., music, voice, etc). These assumptions mean that smartphones can only react to a limited set of potential inputs. This means that most apps on smartphones are special purpose (e.g., those that listen to music in order to identify the name of a song).

Today, smartphones have a wide range of communication channels, from their telecommunications network, to Wi-Fi, NFC and Bluetooth. These provide phones with duplex, high speed and reliable communication channels. However, these channels rely on specialized hardware and software infrastructure, and elaborate handshaking protocols between the phone and the communication channel. SonicData is not intended to compete with these existing data channels but rather to act as an alternative or complement. SonicData is a simple and flexible method, suitable for applications relying on light-weight messages, which can be transferred via existing speakers in the environment and a phone's built-in microphone.

SYSTEM IMPLEMENTATION

Our SonicData prototype consists of an encoder and a decoder. The encoder can be integrated in the infrastructure emitting the SonicData messages, where messages can be composed on the fly. Alternatively, the encoder can be part of an offline process of preparing messages ahead of time. The decoder, an app that runs on the smartphone, listens to the sounds around it and extracts messages (if any).

Encoding Frequency. For encoding, we used a simple version of Frequency-Shift Keying (FSK) modulation to encode information by varying the instantaneous frequency of a sound wave [2]. FSK allows us to encode data into the higher end of the frequency spectrum (Figure 2) that is less sensitive to human ears and more robust to the interference caused by music and human speech. The high frequencies we chose are hardly audible by adults [5], especially when the data is mixed with music or other noise. As we will show later, these frequencies are within the capabilities of smartphone microphones and off-the-shelf speakers.

Message Encoding. SonicData allows a user to enter an ASCII message, then translates the message into a binary stream. The binary stream is encapsulated into packets of 8 ITA2 characters (5 bits/character). A four bit Hamming code is used to provide forward error correction. If the information is longer than 8 characters, it is separated into multiple packets, each of which has control bits that specify its ID and length (Figure 2).

Transferring Message. The messages can be sent in binary or other forms, e.g. decimal. The latter allows the messages to be compact in length. In our implementation, we transformed the message into decimal and mapped them to discrete high, nearly inaudible frequencies ranging from 18 to 19.8 kHz in 200 kHz intervals. Dedicated frequencies of 17.8 kHz and 20 kHz are added to the message in order to robustly indicate its start and end, respectively.

Message Decoder. The decoder was implemented as an Android application. The app runs in the background and checks the presence of frequencies assigned to the data scheme above. A sound is identified for decoding if it appears longer than a certain amount of time (500ms) and its amplitude is greater than a threshold value (30% above the mean amplitude of the frequencies in the range between 15.8 and 17 kHz). Once a packet is received, it is converted into binary form. Single-bit errors could be corrected using the Hamming Code. If more errors occur, the entire packet is dropped. Since SonicData is a simplex system (the encoder only transmits and the decoder only listens), the transmitter continuously plays the message, broadcasting it until the end of a predetermined time. The encoder has no knowledge of whether it was received. The decoder, on the other hand, is aware of a packet being lost, and will then attempt to receive it in the next loop. Lost packets are identified by their IDs. Our current implementation allows a speed of 8 bits per second. The current prototype was designed to serve as a reliable proof-of-concept rather than a speedy optimal solution and can be further improved in both capabilities and speed.

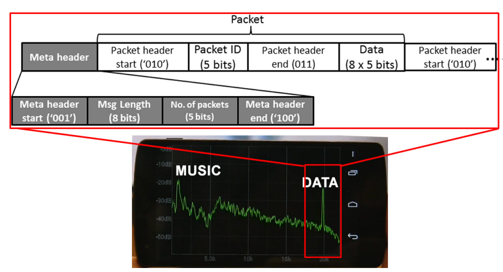

Figure 2. SonicData message template (hamming code is not shown in the diagram).

Adding SonicData to Other Sounds. SonicData messages can be easily added to sound files normally played on speakers (e.g., music, broadcast radio/TV, etc.). Data to be transferred can be modulated on-the-fly or saved in the WAV format, which can be incorporated into any existing video/audio file using free editing software (e.g., Audacity or Windows Live Movie Maker). Mixing data with an existing file may affect the quality of the original audio stream. However, based on our testing, conflicting harmonies (which impacts the quality of the original sound) can be avoided by playing the data on mono or adjusting the amplitude ratio between the inaudible data track and the original sound. This step may not be necessary, depending on the quality of the original sound. Alternatively, SonicData can be played as a standalone audio stream.

EVALUATION

Our evaluation had two goals: 1) to measure the robustness of the SonicData technique; and 2) to understand the impact embedding SonicData into other sound files has on quality.

Phase 1. We evaluate the robustness of the system in three common smartphone usage conditions, each placing the phone in different real-world positions. In the first condition, the phone is located in a user's pant pocket. In the second, the phone is held in front of the user's chest, in a texting position. In the third, the user is talking with the phone pressed against their ear. For each condition, we tested successful data transfer rates from various distances by repeating a data message 10 times. The message we used for the study contained a single character sent 10 times, which is similar to sending a 10-character message once. This allowed us to test the raw performance of the system.

The data was embedded into a song (Kelly Clarkson's "Stronger") using Audacity and a music video of the same song using Windows Live Movie Maker. Three audio files, e.g. a WAV and 2 MP3 files, and a MP4 video file were generated for testing. The MP3 and MP4 formats (lossy compressed formats) and WAV (a lossless format) were used to test the performance of the system. The two MP3 files were generated using a compression rate of 128 and 320 kbps respectively. To remove harmonic effects, the data was only played on the left channel for the three audio files, and played on Logitech Gigabyte T40 speakers. Volume ratio between data and sound was adjusted to 140% for all the tested files to ensure sufficient data volume. Finally, the volumes of all the tested files were calibrated to be equal (65-72 dB-SPL), when measured from a Nexus 5 smartphone located 2 meters away.

Result of Phase 1. We recorded the number of received packets starting from 2m away from the speakers. Using the lossless WAV file, our system successfully received all the packets up to 14m and 18m with the phone being held in front of the chest and against the ear respectively. The talking (ear) position led to longer distance because in this position, the phone's microphone was pointed towards the sound source. Surprisingly, even when placed in the pocket, the phone could still receive ≥ 70% of the packets from up to 6m from the speakers. The results suggest that the SonicData is feasible given the interaction techniques we propose in the next section. However, audio compression impacted the performance of the system and as expected, data transfers completely failed with the 128 kbps MP3. Using the 320kbps MP3, the phone could receive ≥ 60% of the packets when held in front of the chest and against the ear, from up to 2m and 8m respectively. Although Figure 3 (right) shows that data could still be received from 32m, the system suffered significant packet loss. We considered it a failure if more than half of the packets were lost.

Compression data using MP4 also impacted data transformation. The system completely failed when placing the phone in the pocket. When held in front of the chest or against the ear, the system received ≥ 70% of the packets from up to 10m away from the speaker. The result can be considered good as 10m is farther than the distance at which people usually watch a video.

Phase 2. In the second phase, we tested the quality of the data-embedded song with 31 adults (between 22 and 40) and an 11-year old boy. We presented the song in its original and data version (without compression). The participants were asked if they found a difference between the two. The order of the presentation was balanced.

Result of Phase 2. Overall, the adult participants did not find SonicData impacted the quality of the original song. 19 told us that they did not find a difference between the two versions. The rest had split views – 6 liked the original version and 6 liked the data version. The 11 year-old boy could hear the 'ring' tone of the data, but he did not find it bothersome.

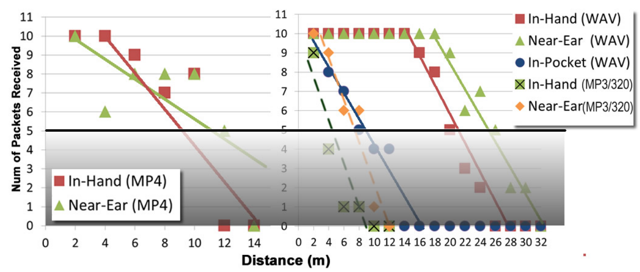

Figure 3. Number of received packets on MP4 (left), WAV, and MP3 (320kbps) (right). The black line indicates half of the tested packets.

SONICDATA INTERACTION TECHNIQUES

To show the possibilities and flexibility of SonicData, we implemented a set of four interaction techniques in an Android app. Some of the techniques are novel and others demonstrate a novel implementation of existing techniques.

Context-based recommendation: People use TV and radio news about public events, traffic conditions or the weather to make plans. For example, people may change their travel plans when informed about a traffic jam on their route on the radio. SonicData can capture these events and provide ad-hoc recommendations for users. In our implementation, SonicData allows a smartphone to respond to traffic updates from TV and radio audio streams, and suggests alternative routes based on the user's current travel plan. It shows a recommendation window on the screen of the phone, and clicking it shows alternative routes in Google Maps (Figure 1a).

Ambient information acquisition: SonicData listens to the audio stream from TV or radio programs, and retrieves details, e.g. channel, program name, song title, artist, host, guest, etc. Unlike existing methods that require databases of potential sound classification samples [8], SonicData is much simpler and more flexible; it just decodes the explicit meta-data in the sound, which could be any message. Additionally, supplemental information (e.g., a web link) of what is currently being played can also be 'heard' by SonicData. In our implementation, we update the status bar of a smartphone interface to show the currently received information (Figure 1b). Users can then selectively tap the notification to get more details. SonicData is aware of all the songs or programs the user watches or listens to. This allows the phone to create models of user preference that could be used in personalizing services.

Enhanced notifications: In public spaces, audio announcements are commonly used to broadcast important messages, which can be missed by people whose attention may be focused elsewhere. For example, at an airport, a user may miss their final boarding call when talking on the phone. With SonicData, the phone could capture the message sent with the boarding announcement, check the user's flight information, and finally notify the user when the boarding call is announced (Figure 1c). In this scenario, using communication channels such as Wi-Fi or Bluetooth would require an established handshaking protocol between the phone and the location infrastructure prior to using the service. By comparison, SonicData is always listening, so the chance of missing important information is reduced.

'Voice' command: In public spaces, audio announcements are also commonly used to request important actions from people. For example, before planes takeoff and touchdown, an audio announcement requests that electronic devices be turned off or switched to flight mode. However, it is difficult for airlines to enforce, and inconvenient for passengers to comply. With SonicData, as the passengers receive an action request from a recording, so do their phones. In our implementation, the phone displays a message explaining that the phone will need to be turned off (Figure 1d).

DISCUSSION AND CHALLENGES

Our SonicData prototype and example interaction techniques highlight the strong potential of the approach to serve as a new and simple data channel that provides many new interaction possibilities for mobile applications. While our current implementation and study has shown that SonicData works well, it has also identified several challenges that we will address in the future.

Speed: The speed of our current SonicData implementation is slow (8 bits/second). Future work will focus on techniques allowing faster data transfer. While sending data over sound is inherently limited in bandwidth, acoustics data transfer has been shown to be robust for sending hundreds of bits per second [6], thus SonicData still is very far from reaching its ceiling limit. We will also investigate the use of mark-up schemes to improve the semantic richness of messages with minimal data requirements.

Security risk: With our current scheme, it is possible for mischievous or erroneous data to be broadcasted, which could be misleading and inconvenient to a user. This raises significant security risks that warrant careful investigation. One solution could use data signing of messages, similar to what is done on the Web with HTTPS, to certify trusted data.

Occlusion and Environmental Noise: Although, our study showed that it is possible for data to be received when a smartphone is in a pocket, there are other conditions, in which the microphone might be covered or where environmental sound might interfere. Further, investigation is required to find out how such challenges can be overcome. One simple scheme might be to broadcast SonicData at different frequencies in its play loop rather than repeatedly playing it at the same frequency.

Audible by kids and pets: Although our implementation uses audio frequencies that can be barely heard by adults, they can still generate noticeable sounds for children and pets. Whether this causes important problems for the technique or not is an open question. A potential alternative solution is to use much higher frequencies, e.g. ultrasonic waves. However, this will require a microphone that can detect higher frequencies, which is not currently available in today's smartphones.

CONCLUSION

SonicData enables machine 'hearable' data to be sent through existing audio infrastructure to smartphones, with little-to-no noticeable impact on the quality of the audible sound. SonicData is a simplex communication technique that is extremely flexible because encoded data can contain any text-based message. Further, SonicData doesn't require any specific hardware and software infrastructure or handshaking protocols. In this paper, we have shown that SonicData is a feasible, powerful and simple new data broadcast technique that enables new interaction possibilities with smartphones.

REFERENCES

1. Buxton, W. Chunking and Phrasing and the Design of Human-Computer Dialogues. *IFIP'86*. 475-480.
2. Gerasimov, V. and Bender, W. Things that talk: using sound for device-to-device and device-to-human communication. *IBM Syst. J.,* 39 (3-4), 530-546, 2000.
3. Harrison, C. and Hudson, S. E. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. *UIST'08*, 205-208.
4. Madhavapeddy, A., Scott, D. and Sharp, R. Context-Aware Computing with Sound. *UbiComp'03*. 315-332.
5. Moore, B. An Introduction to the Psychology of Hearing.
6. Nandakumar, R., Chintalapudi, K. K., Padmanabhan, V. and Venkatesan, R. Dhwani: secure peer-to-peer acoustic NFC. *SIGCOMM'13*, 63-74.
7. Stern, T. Drum and Whistle "Languages": An Analysis of Speech Surrogates. *American Anthropologist*, 59 (3), 487-506, 1957.
8. Wang, A. An Industrial-Strength Audio Search Algorithm. *ISMIR'03*. 7-13.