SERIAL

Sumanth Srinivasan
Electrical Engineering
NYU Tandon
sumanths@nyu.edu

ABSTRACT

SERIAL is a wirelessly connected network instrument, setup as a site-specific installation or a performance companion, which monitors activity in a given network space to generate melodies and timbres that draw from soundscape ecologies.

This project is an attempt to create a sonic footprint of the Internet seeking to not only make people cognizant of the extent to which communication has been delegated to machines, but also present the computer network as a collective organic being that engages in redundant behavior.

1. INTRODUCTION

Every society draws analogy to, and hence identifies itself with a particular kind of machine. As Deleuze states in an interview with Antonio Negri, the old sovereign societies worked with simple machines, levers, pulleys, clocks, but recent disciplinary societies were equipped with thermodynamic machines; control societies function with a third generation of machines, with information technology and computers. [1]

In a classic reflection of the current information age, the physicality of the network is increasingly described using encoded information - so much that this kind of representation begins to act as a substitute to the actual object. Hence an aesthetic expression of this encoded information is necessary to understand the physical manifestation of an environment that increasingly relies on encoded description of itself.

Besides, sound is an important signifier for a number of machines that make up our ecosystem today to an extent that we may have learnt to rely on sonic behaviour to understand the nature of these machines. A popular example is the recent improvements in electric vehicles morphing into a need for sound department in automobile companies in order to have these silent cars make a rumbling sound for the blind.

2. SYSTEMS IN A NETWORK TOPOLOGY

Contrary to popular belief that computers have gotten smaller over the course of history, it is the containing room that has gotten bigger with time; now spanning an entire planet of interconnected devices, each functioning as aggregated pieces of hardware in a larger machine called the Internet.

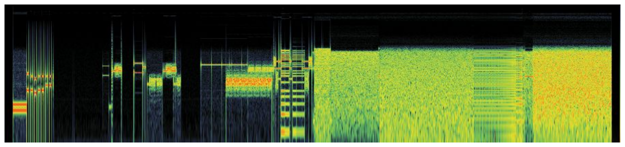


Figure 1. Time Frequency Visualization of the Dial-Up Handshake. Source [2]

As Berners-Lee describes it: *"What was often difficult for people to understand about the design was that there was nothing else beyond URIs, HTTP and HTML. There was no central computer "controlling" the Web, no single network on which these protocols worked, not even an organization anywhere that "ran" the Web. The Web was not a physical "thing" that existed in a certain place. It was a "space" in which information could exist."* [3]

While the Internet theoretically functions as distributed system of information transfer, there are certain centralized access points that guide this flow. In order to be a part of the Internet, one still needs to agree to the names and numbers protocol, servers administered by ICANN and its consensus policies. These structures do not show themselves unless they break, and deny service.

Each of the constituent processes that allow a device to communicate with the network manifests itself as a variant of the underlying bedrock: packet switching. The various protocol layers that each fragment of information may be surrounded by suggests an ostensibly ritualized process to the point where the redundancies begin to inform. It is worthwhile quoting Marshall McLuhan, *the content of every new protocol is always another protocol.* [4]

With the advent of solid-state technology, sound as an important signifier that machines have forever been associated with is lost. In some cases, signifiers have become vastly visual, and in case of the Internet, absent. It is worth noting that our idea of the Internet is increasingly in the form of the applications it supports.

This is very consistent with the general idealistic pursuit of technology to be invisible. While transparent media such as the Internet are helpful in creating conventional impressions of immediacy, it also creates a hidden plane of abstraction that is ready to contain power structures, which an end user may be oblivious to.

3. DETERMINISM IN DIGITAL MUSIC

With the advent of electronic and digital technology, there is a convergence between notions of music and soundscape ecologies. The ability to sample has enabled recreation of geophonic, anthrophonic and biophonic sounds in a structured form, creating new patterns and aesthetics. The tools for schizophonia [5] has allowed the gap between the creation and reception of sound to widen via numerous and steadily abstracting media. Meanwhile, new synthesis techniques that continue to abstract themselves from an audio signal into a control signal, have produced deeper sequences that now begin to lose the apparent determinism often associated with machine music.

This abstraction of control signals from actual sound, coupled with the idea of reflecting on current soundscapes as a means to engineer hidden activities on a sonic realm is an important motivation in the design of SERIAL.

4. SERIAL

SERIAL works in four different modes:

- Interface Mode: The device reads meta-data of incoming and outgoing packets at the interface of the local device. This allows it to access higher level protocol details such as IP Address, Port Numbers and Secure Socket Layer details

- Monitor Mode: The device periodically sends requests to the nearest gateway in the network and receives a sample of network activity. In case of a WPA/WEP encrypted network, only physical and link layer information is readable.

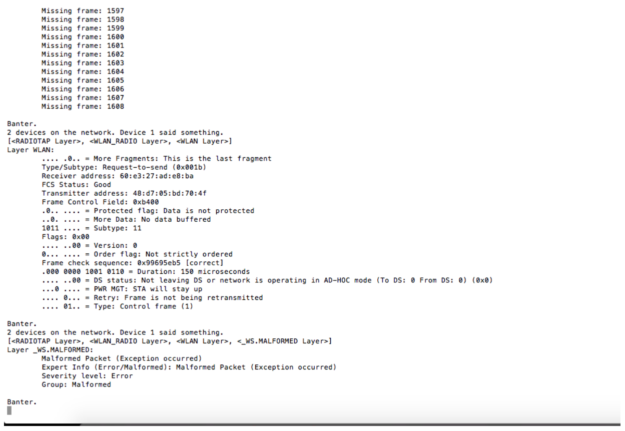
- Soundscape Mode: In this mode, the device continuously reads packets and instantly sonifies them to create a site-specific soundscape.

- Score Mode: In this mode, there is a set score that the software follows while interpreting network meta-data and creating music. This score informs the form of the composition thus unfolding.

4.1 PyShark and Unwrapping Packet Data

Network activity in SERIAL is sniffed and interpreted using Wiresharks tshark and the Python API, PyShark [6]. Depending on the mode of operation listed above, SERIAL is capable of obtaining some or all packet layers as per the OSI model of network protocols. Native PyShark is, however not capable of parsing information obtained via Monitor Mode. Special modifications were made to the library invoke tShark in this mode and decode information appropriately.

Most of the header properties, including addresses, port numbers and flags are returned by PyShark as string values (as shown in figure 2), simplifying the process of passing them as parameters to the SoundModule via a callback function that is triggered for each packet received.



```
        Missing frame: 1597
        Missing frame: 1598
        Missing frame: 1599
        Missing frame: 1600
        Missing frame: 1601
        Missing frame: 1602
        Missing frame: 1603
        Missing frame: 1604
        Missing frame: 1605
        Missing frame: 1606
        Missing frame: 1607
        Missing frame: 1608

Banter.
2 devices on the network. Device 1 said something.
[<RADIOTAP Layer>, <WLAN_RADIO Layer>, <WLAN Layer>]
Layer WLAN:
        .... .0.. = More Fragments: This is the last fragment
        Type/Subtype: Request-to-send (0x001b)
        Receiver address: 60:e3:27:ad:e8:ba
        FCS Status: Good
        Transmitter address: 48:d7:05:bd:70:4f
        Frame Control Field: 0xb400
        .0.. .... = Protected flag: Data is not protected
        ..0. .... = More Data: No data buffered
        1011 .... = Subtype: 11
        Flags: 0x00
        .... ..00 = Version: 0
        0... .... = Order flag: Not strictly ordered
        Frame check sequence: 0x99695eb5 [correct]
        .000 0000 1001 0110 = Duration: 150 microseconds
        .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x0)
        ...0 .... = PWR MGT: STA will stay up
        .... 0... = Retry: Frame is not being retransmitted
        .... 01.. = Type: Control frame (1)

Banter.
2 devices on the network. Device 1 said something.
[<RADIOTAP Layer>, <WLAN_RADIO Layer>, <WLAN Layer>, <_WS.MALFORMED Layer>]
Layer _WS.MALFORMED:
        Malformed Packet (Exception occurred)
        Expert Info (Error/Malformed): Malformed Packet (Exception occurred)
        Severity level: Error
        Group: Malformed

Banter.
```

Figure 2. Serial functioning on Monitor Mode

4.2 Mapping and Audio Handling

Audio handling in SERIAL is done using PyAudio, a python binding around PortAudio, which is a cross-platform audio I/O library built in C.

The mapping of parameters in SERIAL was done with a focus on the need to develop an aesthetic while describing an increasingly coded medium.

The lower level meta-data from the wireless and ethernet layers are mapped to colored noise created by passing white noise through the filter. The frequency of the channel in a given network (which is either 2.4Ghz or 5Ghz) results in picking a sinetone oscillator that either gives sub-frequency sounds or a clear pitched tone, hence creating textural backdrop to the soundscape.

Link-layer protocols, in both secured and unsecured networks provide details about the MAC address, which helps uniquely identify packets sent or received by a device. This information is used to create a dynamic list of devices in the network, while also attaching a unique frequency modulated sound for each device. When a new address is discovered, it is added to this list and a randomly generated parameters for the oscillator and modulation are assigned to it. This gives each device its own distinct chirp.

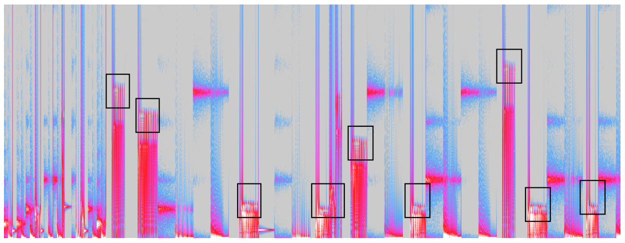


Figure 3. MAC address driven chirps in a public wi-fi network

Wireless networks in homes, privately owned spaces such as offices and university libraries are secured using WPA encryption. This ensures privacy to the user regarding any information that is passed via the network to and from remote servers. Public networks provided in Airports, cafes and train stations, however are often not secured and hence allow easy sniffing of higher-layer details in the packets exchanged in this network. In such a scenario, SERIAL invokes an Interval array that emulates the traditional minor blues scale with root frequency informed by the source IP address of the packet. In the larger scheme, this revelation is sonified as a sudden burst of tonal melody in an otherwise noisy and rhythmic soundscape informed by wireless channel frequencies and hardware addresses. Figure 4 shows a spectrogram of harmonic sequences from an unsecured network.

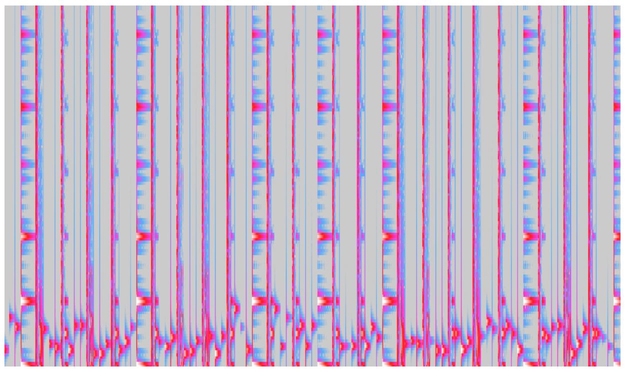


Figure 4. Harmonic melodies created by IP and TCP layer information. Recurring tones with wide harmonics signify the Secure Socket Layer protecting payload.

4.3 SoundModule

The SoundModule is a library of functions designed to perform real-time signal processing upon blocks of audio. Each function takes inputs and outputs in a similar fashion, hence allowing them to be aggregated into a signal chain in an ad-hoc fashion. The various functions currently available in SoundModule were developed keeping in mind their use in building the aesthetics necessary for this project.

The module contains two types of sources: a sine tone oscillator and a white noise generator. Both the functions take in parameters such as duration, sampling rate, decay time, and frequency (in case of the oscillator) and return a block of audio of length corresponding to duration and sampling rate.

There are a number of effects and filter functions. The vibrato function performs Frequency Modulation on a block of input signal and takes additional LFO frequency and depth as parameters. The filterbank_22k function builds a filterbank with a set of five fourth-order band pass filters with center frequencies as follows: 500Hz, 1000Hz, 2000Hz, 5000Hz, 10000Hz. This function is inspired by the aesthetics of the filter bank in the analog synthesizer Buchla 100e.

There are two utility functions for clipping and mixing. The clip function allows for non-linear compression by taking compression ratio and input gain as parameters. It offers the aesthetics of hard clipping and related harmonic distortion.

The mix function allows adding multiple tracks of the same length into one channel, hence allowing additive synthesis. The pan_stereo function enables spatially panning a given track in a stereo channel. The encoding of output in this function is optimized for PyAudio to be able to handle the struct.

```
import SoundModule as sm

Fs = 44100
osc = sm.oscTone(1,0.9,700,Fs)
x = sm.vibrato(osc,10,5,Fs)
str_out = sm.pan_stereo(x, 0.3, 0.7)
# Returns a packed struct ready to write
stream.write(str_out)
```

By importing the SoundModule into the main program, it is easier to patch a chain of modules to create sounds with network meta-data as parameters in real-time. This also allows the resulting sound to be site specific, since the parameters depend on properties of the network it is reading from.

5. SETUP

5.1 Serial as an installation

Every space has a unique network topology and comprises of different set of communicating devices from time to time. Besides, the combination of technology put together to build the local network infrastructure may be specific to the space. These details are carried as information by the packets in the network. SERIAL is programmed to respond to these details in the process of sonifying packet information. This makes it a site-specific installation that produces a unique soundscape depending on the network it monitors and the activity on it.

5.2 Serial as a performance

In its Score Mode, SERIAL can traverse through changes in a number of different parameters, adding performative elements in its interpretation of Internet activity. This comes via its mapping of frequency as well as control over passage of time during as it streams these sounds. This feature allows it to act as a dynamic companion to a human performer who may improvise alongside the computer to make traditional or experimental form of music that takes a granular as well as a macroscopic view of all the communication happening in the space.

6. CONCLUSIONS AND FUTURE WORK

This paper outlines the social motivations behind constructing a network enabled instrument such as SERIAL as well as the the technology involved in building it.

The paper also explains in detail the various functioning modes and components of SERIAL and how they work together to transform invisible and inaudible network activity of everyday space into a soundscape of redundant but organic behaviour.

SERIAL can be put to use as a tool for education of computer networks and the infrastructure that supports it, as well as a medium of surveillance intervention in a modern state that contains both corporate and government organizations capitalizing on the wealth of data available to them by the mere act of people using devices to communicate with each other.

7. REFERENCES

[1] G. Deleuze, “Negotiations.”

[2] “Dial Up Handshake Visualization.” Freesound. [Online]. Available: https://www.freesound.org/people/Jlew/sounds/16475/

[3] T. Berners-Lee and M. Fischetti, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*, 1st ed. Harper San Francisco, 1999.

[4] A. Galloway, *Protocol: How Control Exists After Decentralization*, ser. A Leonardo Book. MIT Press, 2006. [Online]. Available: https://books.google.com/books?id=4BWmPwAACAAJ

[5] R. M. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*. Distributed to the Book Trade in the United States by American International Distribution, 1977.

[6] D. Green, “PyShark - Python Packet Parser for tShark.” [Online]. Available: http://kiminewt.github.io/pyshark/