Design and Implementation of Stetho — Network Sonification System

Masahiko KIMOTO [1,2] Hiroyuki OHNO [3]

Graduateschool of Information Science and Engineering, Tokyo Institute of Technology. [1]

Art Media Center, Tokyo National University of Fine Arts and Music. [2]

Emergency Communications Group, Information and Network Systems Division, Communications Research Laboratory. [3]

Abstract

We have been developing "stetho", the network traffic sonification system. We have provided NetSound service using stetho for these four years. From this experience, several problems of stetho has been appeared. In this paper, we describe the improvement of stetho system. We extend the format of configuration file so that we can describe detail settings, and we introduce expansion port to receive many events such as SNMP. We discuss the improvement of the stetho and evaluate it.

1 Introduction

Stetho system[5] which we have been developing since 1995, converts network traffic data into sound. Usually, we use visualized forms for monitoring networks and systems. On the contrary, **stetho** provides sonificated forms.

In the area of algorithmic composition, researchers try to generate sound and music from various sequence of numbers. However, in many of these trial, they don't evaluate it from the view of Recognition. **Stetho** uses network traffic as the source of sound, the output sound is useful for network administration.

In this paper, we describe the history of **stetho**, design and implementation of the improvement of it and experiment to evaluate it.

2 Methods of network observation

2.1 Visualization

Administrator's daily works — monitoring, investigation and resolving, for instance, are mandatory to keep networks stable. Monitoring networks and servers is highly tough work and it takes much amount. Generally, this work requires much experiences and intelligences. Target information have the following characteristics in this context. First, the information changes from moment to moment, because usage and topology of networks may change. Next, there are various kind of information. Recently, network systems consist of many equipments, network services with many applications. In addition, the amount of log data gets larger and larger.

Therefore, some systems to present these information are developed. MRTG[1] and TTT[2] present network traffic in graphical form. Analog[3] summarize the log file into graphs. These tools uses method of visualization.

However, by using these tools, administrators must watch monitor screen by eyesight. If many people observes, they need large size of monitor or many number of monitors.

2.2 Sonification

Target information stated in the previous section which has much kinds and quantity is thought to be easy to grasp if it is expressed as sound. This is because there are characteristics that multiple kinds of information changing continuously is easily grasped in the sense of hearing.

There are roughly three characteristics for the human sense of hearing [4]. The first characteristic is that the sense of hearing is always opened. An observating person can't notice the matter which is not within view. However, an hearing person can notice every matter because it always faces in the whole direction and input by the sense of hearing which faced and which went through the ear is being opened. This fact shows that an hearing person doesn't need to concentrate all attention toward the sound. For example, an hearing person can notice warning sound even when most consciousness is turned to other works.

Second characteristic is the point that the sense of hearing is superior in grasping of matters changing from moment to moment. It is given that a interval between sounds as a rhythm and a dynamic image can be grasped.

Third characteristic is the point that it is suitable for grasping more than one matter at the same time. It is given that people can listen to the melody of the specific instrument from an ensemble by an orchestra.

Sound is thought to be suitable for the expression of the information whitch "always changes," "shows more than one matter at the same time" and "consits of many kinds" from these characteristics. We have been trying to express the network information by using sound based on these reasonings. Stating straightforwardly, the purpose of this research is the "Generating back ground music for network administrators".

This music is generated automatically with the purpose of watching the the system from the various administrative information, and the following conditions are required.

- Be comfortable as a music.
- Migration of status is possible to be grasped.
- Accidental event is possible to be noticed immediately.

3 Stetho system

3.1 Background

We have been developing **stetho** — the network sonification system, since 1995[5]. **Stetho** is for "stethoscope" of networks. **Stetho** reads the output of tcpdump command, checks matching in regular expressions, generates corresponding MIDI events.

We have used **stetho** in NetSound which is an work of media-art for a long time. From this experience, some problems are appeared.

Target network is limited to that which are connected to one host. **Stetho** processes each packet and it can't receive more rich event such as that from IDS(Intrusion Detection System). In addition, poor configuration syntax causes only poor MIDI events generated.

3.2 Design of stetho system

We have re-designed **stetho** system to solve these problems (figure 1). The goals were the followings.

- Enhancement of support of MIDI device.
- Adding extension port to receive events from outside.
- Enhancement of configuration syntax.

3.3 Implementation of stetho

Stetho is written in C language and is about 7000 lines.

We employed TiMidity++ which was a software MIDI sound generator as default output device. TiMidity++ 2 or later can receive real-time events via networks. We re-implemented **stetho** to cope with it.

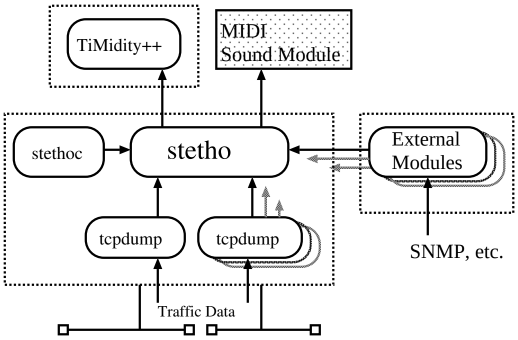

Figure 1: Improved stetho system.

We also supported sequencer device of OSS which was famous sound driver for UNIX, raw MIDI device and serial MIDI device. As the result, **stetho** became possible to be used on almost all UNIX platform.

In the case of NetSound described later, it was completed by using the sampling sound which is made by a musician. General GM sound module or other generic modules are used when a general user uses **stetho**. In this case, user should be able to describe various expressions in the configuration file without editing sampling file. Thus, the grammar of the configuration file was expanded aiming the precise control of the diverse sound devices. The expanded points are the followings.

- The sound sequences of more than one track (chord) can be assigned toward one phrase.
- Initialization sequences for each devices can be described.
- The process which is executed as the child process can be specified rather than tcpdump.
- When an event occurs continuously, it can be specified whether it is piled up including the maximum number of notes.

A policy from former version of the pattern match of each lines toward the input, was followed. In the process of the re-design, a direction to include the function of tcpdump into the inside of **stetho** by using libpcap[11] was examined. There are some advantages in this, buffering in tcpdump can be excluded and the problem that the output of tcpdump is different in many OS can be solved. However, implementation of external module is easy if a former policy was used. We have chosen to execute tcpdump.

An example of configuration file is shown in figure 2.

The function which accepts TCP connections was added to develop the external module. Pattern match

```
# # is comment line
inhibit_midi_channel=10
oss_sequencer_device=1
timidity_buffer=1.0
tcpdump_options=-n -e

define midi_initialize
A EXx41,x10,x42,x12,x40,x00,x7f,x00,x41,xf7
 # MIDI Initialization
end

define 1
 # ``define'' defines sound sequence.
a CH @2 v110 k110
 # Init. MML if head is lower letter.
b CH @14  v110
A o4c4,110 d4,110
 # MML if head is from A to K
B o3e4,110 f4,110
end

when /.*\.(80|8080|http)[: ].*/
        # specify regex after when.
        #
play 1  # play specifies sound.
mono    # mono or poly. In case of poly,when
poly 4  # some packets are observed,play for
        # each. Argument is maximum of sound
        # In case of mono, only one sound is
        # played.
vshift +10
        # elevate velocity against traffic.
nshift +1
        # elevate note against traffic.
end
```

Figure 2: Example of configuration file.

for each line is also done for input from the external module. The occurrence time of the event is the time of the moment of **stetho** internal when input was accepted. It is a premise that there are little delay in network between the external module and **stetho**.

WebMelody[15] which will be introduced as related work later, plays SMF (Standard MIDI File) corresponding to events. Note number is made to change with **stetho** corresponding to the amount of traffic. This process is difficult with method to play SMF like WebMelody, because if the note number of “note-on” is changed, corresponding “note-off” must be looked up and changed. Therefore, in **stetho**, users describe MMLs in configuration file, and **stetho** converts them into intermediate codes. The intermediate code is composed of sequence of play time, note and length. The timing of events is managed “since time” since the start of **stetho**. Occurrence of event is managed with queue which includes play time and events. Phrase and information on playing phrase are assigned toward the pattern of the regular expression. The following operations are done in every 1/50 second.

- Update of clock.
- Search playing phrase and check events to insert to queue at that time.
- Add the shift of pitch and velocity corresponding to the amount of traffic to the intermediate codes, translate them to events, and push them into queue.
- Look up event queue and play all events which should be played at that time.

Insertion of phrase is asynchronous and is done when input from tcpdump or external module is received. An internal data structure of **stetho** is shown in figure 3.

4 Stethocast

4.1 Overview of stethocast

We introduce the case study of **stetho**. Stethocast is a package which streams sound of **stetho**. There are stethocast/RealSystem and stethocast/MP3. Stethocast/MP3 sends MP3(MPEG Audio/Layer3) stream by using icecast [12]. Figure 4 is structure of stethocast/MP3.

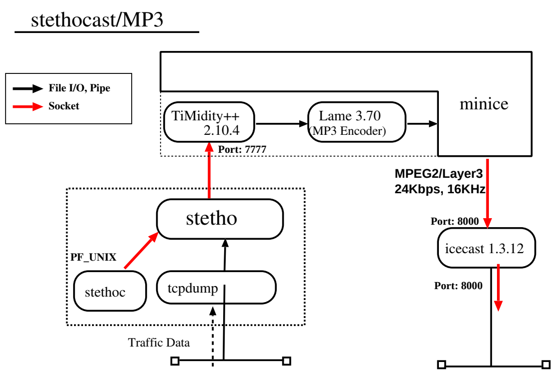

Figure 4: Structure of stethocast/MP3.

4.2 Minice

We developed minice[13]. Minice is a simple sound stream source for icecast. Goal of minice is to provide simple facility and enough stability.

Minice creates a UNIX pipe and connect player process and encoder process, and relays the sound stream from the encoder to icecast server. Arguments of each programs and bit-rate are specified in configuration file manually. Minice itself doesn't care about the format of data streams. Minice is

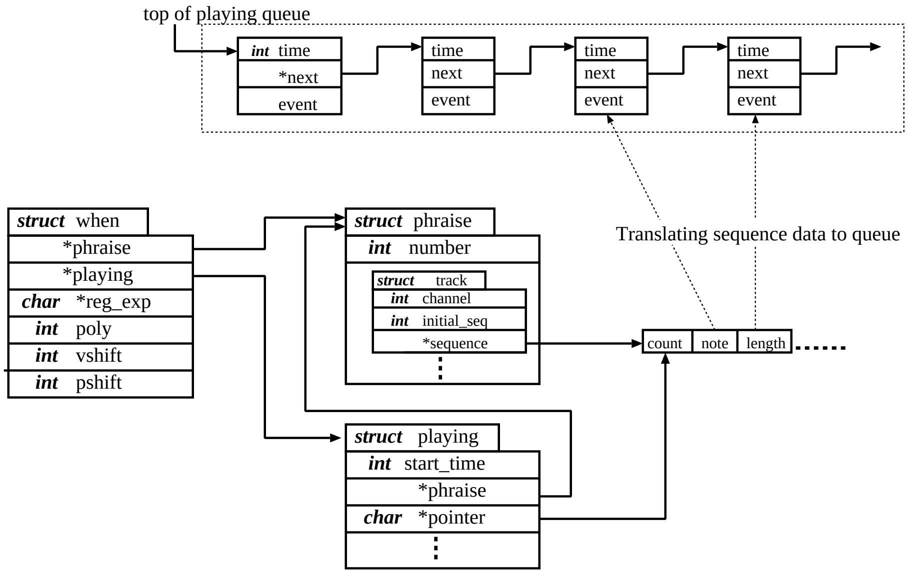

Figure 3: Internal data structure of stetho.

small and stable, and is about 1000 lines in C language.

4.3 NetSound

Web site of "sensorium" is constructed in Internet World Exposition '96 as one of the pavilions[8]. We provided "Netsound" which provides sonificated network traffic by through of RealAudio streaming [6].

At that time, NetSound consisted of two PCs, Windows95 and BSD/OS 2.1. We also used AKAI sampler module. We changed them into Linux, BSD/OS 3.1 and software sampler on BSD/OS in 1997.

Though the order of development is out of sequence, NetSound is one of the contents which stethocast was used, for convenience' sake. What we want to argue here is the evaluation toward NetSound, and is simultaneously the evaluation toward **stetho** as a media-art platform. A gold prize was awarded to Sensorium in Arc Electronica Festival '97 of the Arc Electronica Center[1] auspices. Thus Sensorium pavilion including NetSound got evaluation as a work of an art. Though there was a problem that the output of the pleasant sound was difficult in most of the other systems, NetSound is thought to settle a success as a work of an art.

In 1999, we lost the observation point because of the re-construction of our organization, we currently only provide a sample sound file [2] We are planning to provide new NetSound at Art Media Center of Tokyo National University of Fine Arts and Music. The plereminaly URL is the below:

`http://stetho0.noc.geidai.ac.jp:8000`

5 Experiment

5.1 Objectives

We mentioned requirements for **stetho**, to be pleasant as music and administrators can recognize the status of system. We think the success of NetSound is respectable, though the evaluation is difficult because a listener's favorite is contained. Next, it is become a point that the condition of traffic can be grasped from the sound of **stetho** or not. Evaluation about this point isn't done enough even in the related researches.

When we aim to develop it as an automatic composing system from now on, the information how listeners analyse the sound and recognize it could support the development.

So, we had an simple evaluation experiment, though it is not enough as a HCI(Human Computer Interaction) experiment. The objectives are investigation how traffic can be grasped correctly from sound and rough investigation of recognition of listeners.

5.2 Experiment

In this experiment, subjects listen to sound translated from pseudo traffic, and we examine how they recognize it.

Three hosts (host A, B, C) were prepared for the experiment. Each host's specification and its part are shown in the table 1. All hosts are connected with one 10BASE-T hub. This hub is repeater-hub,

[1] http://www.aec.at
[2] http://netsound.ohnolab.org

and all traffics in the network are observed in the observation host where **stetho** works. 100MBit/sec is estimated to be too fast for a connectivity to outside in ordinal organization, that is why we used 10BASE-T hub.

All the programs stated in the following were written in ruby[14]. The program transfers 8 kinds of size from 1KB to 128KB in random intervals from 5 seconds to 20 seconds, at Host C from WWW server (Host A). Four processes are executed concurrently. The main part of the program is shown in figure 5.

```
def mainloop
    while(1)
        n = rand(5) + 1
        for (i = 0; i < n;i ++)
            l = int(rand(8))
            system("wget urls[l]")
            sleep(rand(3))
        end
        sleep(rand(15)+5)
    end
end
```

Figure 5: Traffic generating program.

Subjects operated an interface in figure 6 with listening to the sound of **stetho** raising from Host B. Each button from 1 to 5, means the amount of the traffic.

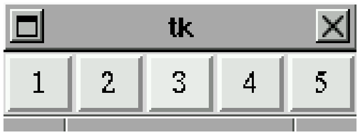

Figure 6: User interface for observation.

A traffic to observe is only HTTP. The sound assigned to HTTP is that used in NetSound. It is like a bell ringing for each packet. Note number rises following to traffics' increasing. This configuration is also same as that in NetSound.

The subjects (four persons: A to D), are familiar with network administration, have vague image about the network traffic pattern, have the experience to listen sound of **stetho** and NetSound. However this is their first time to listen the sound pattern of the experiment.

They operated the interface for three minutes after they listened to the sound for two minutes to memory the steady condition, especially maximum and minimum.

5.3 Result

Result of the experiment is shown in figure 7. Number of packets per second and transition of button are shown in vertical axis. Transition of button is slided 5 seconds because **stetho** has 5 seconds buffer in it. Result from 120 second after starting to 300 second is shown in horizontal axis.

When a graph is surveyed, it is understood that the tendency of the operation varies in the listener.

The frequency of the operation increases as for the subject C in latter half though operation times are a little. This is thought that subject C obtained an experience toward the operation. Result of subject A and subject D are similar in the point of frequency of operation and phased elevation.

We draw additional lines to the graph at the points of maximum. They help us to pay attention to a characteristic points.

- Peak of recognition is delayed against the peak of traffic. For example, there are 2 second delay for subject A and D, 4 second delay for subject B around 126 second. These delay is seen at other peak points. We think it is the delay of operation of interface and buffering in listener's head.
- In the latter half, delays of traffic and recognition become big. For example, 5 to 6 seconds delay for subject A and D, 2 seconds delay for subject B and C, is seen at the peak around 288 second.
- The operation elevates step by step after peak of traffic in case of subject A and D.
- The distinction of the peak by the subjects seems to depend on the amount of traffic of back relatively. For example, around from 200 second to 218 second and around from 220 second to 230 second, though these difference is small increase subject A, B and D recognize the increase.
- Comparing appearing peaks, in points of disappearing peaks, the operation seems to be more delayed.

And, the following mentions were presented from subjects after the experiment optionally.

- "Noisiness" was made a standard for the amount of traffic.
- An pitch of sounds didn't necessarily correspond to the amount of traffic which could be recognized by "noisiness".
- About 1 second sound is accumulated in the head for the distinction of "noisiness".

Host	CPU	Memory	OS	Act
Host A	VIA C3 700MHz	256MB	FreeBSD 4.3	WWW Server
Host B	Pentium 200MHz	128MB	FreeBSD 4.1	Observer(stetho)
Host C	PentiumII 333MHz	128MB	NetBSD 1.5	Traffic Generater

Table 1: Hosts for the experiment.

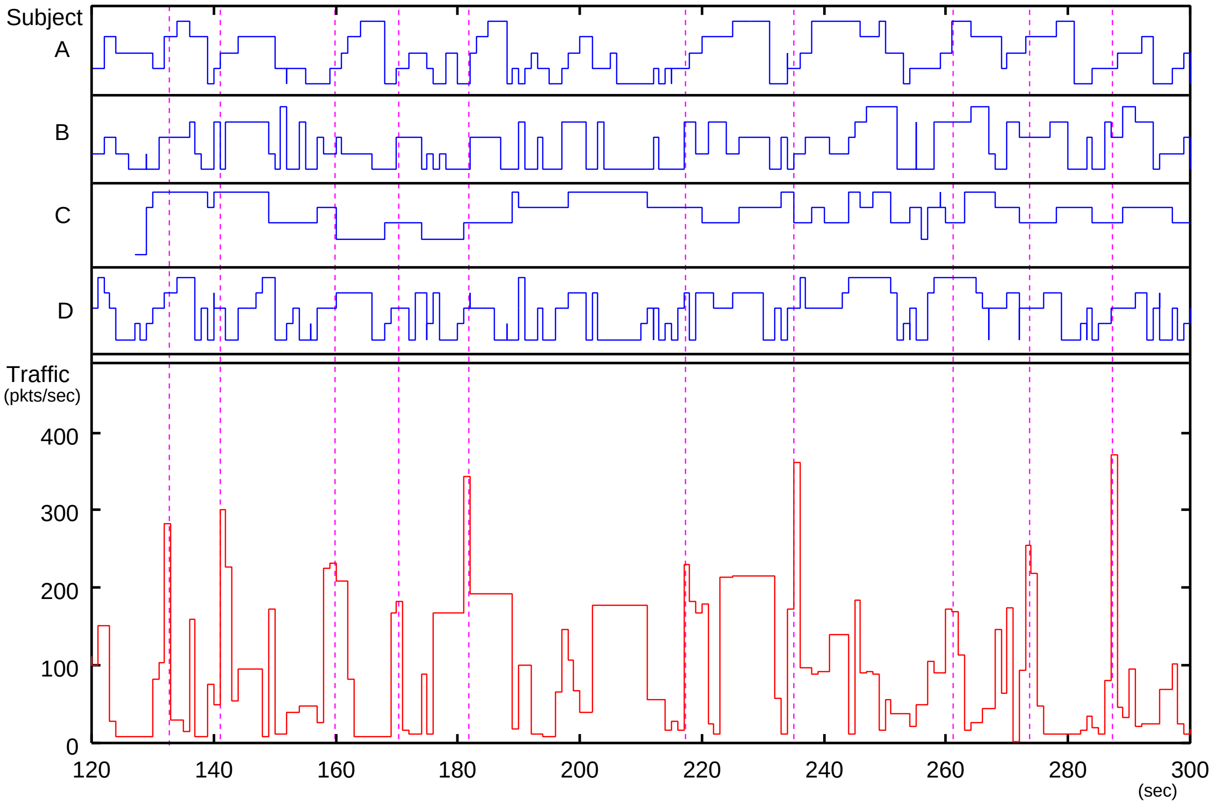

Figure 7: Graph of the results.

- Sound-less condition is defined as traffic level 1.
- When sound was intermittently, it was made 2 or 3.
- Even the same pitch, noisiness, it is recognized as large quantities when it continues for a long time
- It seems that the timing which becomes quiet doesn't reach consciousness though it reaches consciousness when sound begins.

6 Discussion

6.1 Evaluation of re-design

The WAV sound files are made by musician for NetSound. In configuration file of improved **stetho**, users can easily describe the chord and melody. Using extension port, **stetho** can receive any events from IDS or SNMP agent. Though this facility is currently still under testing, we will develop some prototypes as external module.

6.2 Evaluation of experiment

From result of the experiment, the peak of the subject's recognition aligns with the peak of the traffic mostly, although there are delays of the recognition. When noisiness is made a judgment standard, it becomes a relative judgment toward the past traffic for one second. When an pitch is made a judgment standard, the value which can be recognized from noisiness doesn't necessarily correspond with the note number which **stetho** plays. Though the change in the pitch and the noisiness was oriented toward the amusement as a media art in NetSound, it isn't so suitable from the view of the recognition. The method which uses continuation sound and changes pitch corresponding with traffic for past one second is suitable if interval is used.

By this configuration, **stetho** is suitable for purpose to recognize the happening of change and to graspe traffic vaguely. It shows a tendency to delay the recognition for disappear of sounds.

6.3 Related works

WebMelody[15] converts log file of the WWW server into sound. It plays SMF corresponding to the contents of log file. It is tried that listeners do not get tired with the sound. However, target object is limited to log file and it doesn't support so various sound devices as **stetho**.

In Peep[16], observation part and sound generation part are separated, and they communicate via networks. It can generate sounds corresponding to pattern in log file and load of CPU. Such a function would be implemented as external module for **stetho**. It plays not MIDI event but PCM sound, there is a disadvantage that is hard to express information changing dynamically as sounds with Peep

In the area of algorithmic composition, there are some researches vto try to translate numeric sequence, for instance fractal set[18][19] or pi[20], into music. By recent researches, network traffic has self-similarity, so trial with **stetho** is one of the typical work of algorithmic composition. However the output sound of **stetho** is not a meaningless sequence of sound but what listeners can recognize corresponding information with.

7 Future works

We describe three future works for **stetho** here.

First is the improvement of expression. A change in time series, accumulation, and so on can't be expressed with present **stetho**. Though the elevation of note and velocity corresponding to traffic is being made possible to express imitatively, for example, "If the total amount of certain type of packets gets larger than certain threshold, play this sound" can't be expressed. There are two methods for this improvement, more enhancement of configuration file and implementation as external module.

Next is the orientation of the automatic composing system. Some automatic composing systems are based on theory of composition, and uses random sequence as input. We are planning to use sequence of numbers generated from network traffic for input of these systems. This trial is oriented to "be comfortable as music" which is one of the aims of **stetho**.

More verification of the validity of **stetho** in the network administration is given at the end. From the experiment in this paper, recognition of administrative information from sound of **stetho** is appeared to be basically efficient. However, adjustment of parameters of velocity and assignment of sound are needed to recognize the change of traffic correctly. We will engage the further experiment to estimate an assignment of sounds. Moreover, we must verify that listeners can recognize an unusual condition as well, and are planning experiments using DDoS simulator[21].

8 Conclusion

We introduced the history of **stetho** and NetSound. We also described implementation of **stetho** and evaluated it. By the result of the experiment, it is appeared that administrators may use the sound of **stetho** to grasp the condition of networks. As the future works, we will provide new NetSound contents at Tokyo National University of Fine Arts and Music. In the art university, we hope we will have many supports by artists.

Acknowledgement

The authors are grateful to all members who have worked on **stetho** and NetSound. We thank Mr. Nishimura on working of NetSound, Mr. Yamaguchi on creating sounds. We also thank Prof. Itoh and Mr. Kozumi in AMC, Tokyo National University of Fine Arts and Music, who give us new experiment platform.

References

[1] The Multi Router Traffic Grapher (MRTG), http: //www.mrtg.org/

[2] TTT: Tele Traffic Tapper (version 1.6), http:// www.csl.sony.co.jp/ person/ kjc/ software.html#ttt

[3] Analog: WWW logfile analysis, http:// www.analog.cx/

[4] Carla Scaletti and Alan B.Craig, "Using sound to extract meaning from complex data", 1991, Extracting Meaning from Complex Data:Processing, Display, InteractionII

[5] Tetsuya Narita and Hiroyuki Ohno, "**stetho**: network traffic sonification system", SIGDSM, IPSJ, DSM-951150, Nov. 1995,

[6] Tetsuya Narita and Hiroyuki Ohno, "Utilization of network sonification system", Proceeding(3) of 52nd National Convention of IPSJ, pp. 473, IPSJ

[7] Masahiko Kimoto and Hiroyuki Ohno Improvement of stetho — network sonification system, DSM Symposium, IPSJ, Feb. 2001

[8] Sensorium, http://www.sensorium.org

[9] TiMidity++: MIDI to WAVE converter / player, http:// www.goice.co.jp/ member/ mo/ timidity/

[10] OSS: 4Front Technologies, http:// www.opensound. com/

[11] The Packet Capture library, http://www-nrg.ee.lbl.gov/ nrg.html

[12] Open Source Streaming Audio, http:// www. icecast.org

[13] Masahiko Kimoto, minice – Icecast stream source with minimum functions, FreeBSD PRESS NO.7, Mainichi Communications

[14] Ruby: Object Oriented Script Language, http://www.ruby-lang.org/

[15] M. Barra, T. Cillo and A. Santis, "WebMelody: Sonification of Web servers", at WWW9 (May 2000)

[16] "Peep(The Network Auralizer): Monitoring Your Network With Sound", 2000 LISA XIV, Dec. 2000, http:// peep.sourceforge.net/ docs/ lisa2000.pdf

[17] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Willson. "On the Self-Similar nature of Ethernet Traffic (Extended Version)", IEEE/ACM Transactions on Networking, 2(1):1-15, 1994

[18] Hearing the Mandelbrot Set, http:// members.aol.com/ dshp3/ mandelmaps.html

[19] Fractal Tune Smithy Home Page, http://www.tunesmithy. connectfree.co.uk/

[20] π is music, http://web.kyoto-inet.or.jp/ people/ haselic/ pi/pai.htm

[21] Hiroyuki Ohno, Hiroshi Takechi and Hidemi Nagashima, "Vulnerability Data Base and DDoS Attack Simulator", DSM Symposium, IPSJ, Feb. 2001