管理情報可聴化システム (stetho) の拡張とその評価

木本 雅彦[1 2]　　大野 浩之[3]
東京工業大学大学院 情報理工学研究科[1]　東京芸術大学 芸術情報センター[2]
総務省 通信総合研究所 通信システム部[3]

概要

ネットワークやシステムの監視作業に用いる情報のように、種類や量が多く時間と共に変化する傾向をもつ情報は、音で表現すると容易に挙動を把握できる。著者らは、音を利用したネットワークトラフィック観測支援機構として、トラフィック可聴化システム (stetho) を開発してきた。stetho はメディアアート作品である NetSound において長期に渡り運用されてきたが、同時に問題点も明らかになってきた。本論文では、stetho の構造の見直しを行い、再設計と再実装を行う。具体的には、一般的な GM 音源などの利用を想定して、細かな音源の制御が記述できるように設定ファイルの文法を拡張する。加えて、SNMP などのイベントを受け取るための拡張ポートを用意する。改良後の stetho を用いて機能拡張による表現力の向上を評価し、管理情報を音声に割り当てる際のアルゴリズムについて考察する。

Improvement of stetho — network sonification system

Masahiko KIMOTO [1 2]　　Hiroyuki OHNO [3]
Graduateschool of Information Science and Engineering,Tokyo Institute of Technology [1]
Arts Media Center, Tokyo National University of Fine Arts and Music [2]
Communications Research Laboratory, MPMHPT [3]

ABSTRACT

We have been developing the stetho, 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 the next step of NetSound service.

1　はじめに

ネットワークシステムの稼働状況の監視には、通常文字やグラフなどの視覚情報が用いられる。これに対して、著者らはこれまで管理情報を音声で表現する「可聴化」を試み、ネットワーク聴診器システムである stetho の開発を行なってきた。本論文では stetho の拡張と、それを用いた今後の展開について述べる。

2　ネットワーク監視手法の考察

2.1　視覚による監視

ネットワークを正常な状態で稼働させ続けるためには、管理者の日々の 1) ネットワークの監視、2) 問題の原因の究明、3) 問題の解決といった作業が必要である。このうち、トラフィックや各種サーバの挙動の監視は大きな割合を占めると同時に、管理者に大きな負担がかかる。この作業は、一般的に経験と知識が必要な上に、予測できない異常事態に対する迅速な措置が求められるためである。ここで監視対象となる情報には、以下に挙げる傾向がある。まず、これらの情報には、時々刻々と変化し続ける傾向がある。これはネットワークの利用状況が常に変化し続けることと、ネットワークの構成が変化し得ることが原因である。 次の傾向として、種類が多いことが挙げられる。ネットワークは様々なシステムから利用され、その上で様々なサービスが稼働しているため管理者は多くの種類の情報を扱う必要がある。 更に、特に近年ネットワーク規模の拡大に伴って、常に多くの種類の情報が変化し続けるために、トラフィック情報やサーバが出力するログの分量が多くなる傾向にあるという特徴が挙げられる。

そこで、これらの情報を分かりやすく管理者に提示する方法の研究が行われている。MRTG [1], TTT

[2] などはトラフィック情報を図示するものであり、Analog [3] のように各種ログ情報を整理して提示するものもある。これらの多くは視覚情報として整理した情報を提示する。

しかし、システムの状態を視覚的に監視する作業は動作モニタを常に見ていなければならない上、多人数が同時に監視を行うためには巨大な表示装置を用意したり、個別に表示方法を用意するといった対応が必要になる。

2.2 音声を用いた監視の利点

前節で述べた監視対象の情報のように種類や量が多く時間と共に変化する傾向をもつ情報は、音を利用して表現すると把握が容易になる。これは聴覚に、時間と共に変化する多種類の情報を把握しやすいという特徴があるためである。

人間の聴覚には、大きく分けて三つの特徴がある [4]。第一の特徴は聴覚が常に開かれているということである。目を通しての視覚を利用した入力は、観測者が観測対象として注目したものにのみ開かれているため、観測者は視野に入らない事象に気付くことができない。それに対して、耳を通じた聴覚による入力は常に全方位に対して開かれているので、観測者はあらゆる事象に気付くことができる。この事実は観測者が全ての注意を音に対して注ぐ必要がないことも表している。例えば、観測以外の作業にほとんどの意識を向けている場合でも、観測者は警告音に気付くことができる。

第二の特徴は時間と共に変化する事象の把握に、視覚よりも聴覚の方が優れている点である。その根拠としては、音からは動的なイメージを連想しやすいことや、音の間の時間差をリズムとして細かく把握できることなどが挙げられる。

第三の特徴は複数の事象を同時に把握するのに適しているという点である。これは、人間が音の方向に敏感であることや、さまざまな音のパターンを記憶して異なるパターンを識別できることから導かれる。この特徴の例としては、オーケストラによる合奏の中から、特定の楽器の旋律を聴き分けられることが挙げられる。

これらの特徴から、音は「常に変化し」「同時に複数の事象を表し」「種類が多い」情報の表現に適していると言える。この点に基づいて、著者らはネットワーク管理情報の音声による表現を試みている。この研究の目的を端的に述べると、「ネットワーク管理者用のバックグラウンド音楽の生成」である。この音楽はシステムの挙動を監視する目的で、様々な管理情報から自動生成されるものであり、以下の要件が求められる。

- 音楽として心地よいものであること
- 定常的な状態の変化を大局的に把握できること
- 突発的なイベントを即座に正確に把握できること

3 stethoシステムとNetSoundの概要

3.1 stetho システムの概要

著者らは、ネットワークトラフィックの可聴化システムとして、1995 年より stetho システム (以下 stetho) を開発している [5]。stetho は tcpdump を入力とし、行単位で正規表現によるパターンマッチを行い、当てはまった行に対応する MIDI イベント列を発生する。stetho の基本的なシステム構成を図 1 に示す。

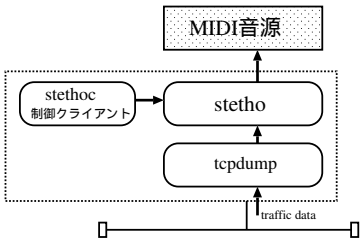

図 1: stetho システム

3.2 NetSound の概要

1996 年に開催された Internet World Exposition '96 のテーマパビリオンとして "sensorium" というサイトが設けられた [7]。著者らはこの一環として、ネットワークトラフィックの変化を音楽で表現するコンテンツである "NetSound" を設置した [6]。NetSound は、stetho が生成した音声を RealAudio を用いて配信するコンテンツである。

NetSound は、当初 2 台の PC (Windows95 と BSD/OS 2.1) と AKAI 社のサンプラー音源を用いて構築された (図 2)。この構成は後 (1997 年後半) に Linux と BSD/OS 3.1, ソフトウェア音源に置き換えられた (図 3)。ソフトウェア音源には、timidity 0.2i にリアルタイムイベントを扱うための改造を独自に施したものを利用した。この変更によって、Windows 機とフロッピーディスクでしか音声データを交換できないサン

プラー音源を排除できたため、遠隔地からネットワーク経由での保守管理が可能になった。

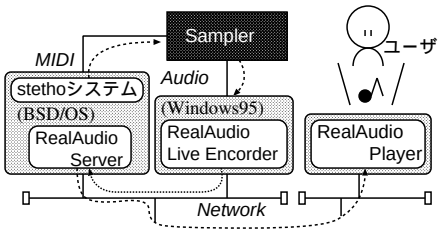

図 2: netsound 機器構成

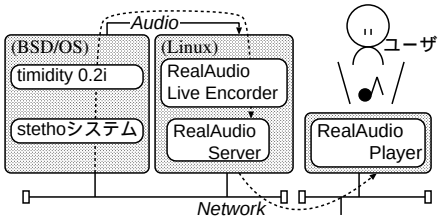

図 3: netsound 機器構成 2

NetSound 用の音源は音楽家との共同作業によって作成された。観測点となるネットワークのトラフィック傾向を解析しながら音の割り当てを決めていった。HTTP などの流量が多いトラフィックについては、発音数を制限するなどの工夫を行った。当初はサンプラーの機能で細かな発音の操作を行っていたが、timidity に変更する際、stetho 側に発音数の制限の機能などを取り入れていった。

NetSound は、開設した 1996 年 7 月 24 日から 1997 年 1 月 7 日までの 171 日間で、延べ 2872 人から利用された。ホスト数は 1227 ホストであり、そのうち 44%のホストからは 2 回以上のアクセスがあった。

Netsound を含めた Sensorium パビリオンは、Arc Electronica Center [1] 主催の Arc Electronica Festival '97 において金賞を受賞し、芸術作品として高い評価を得た。コンピュータの動作を音声で表現する従来のシステムの多くでは、音楽として耳に心地よい音声の出力が難しいという問題があったが、NetSound は芸術作品としても成功を収めたものと考えている。

3.3 stetho の問題点

これまでは stetho が動作しているホストで tcpdump を実行していたため、観測対象がそのホストだけに限られていた。また、パケットという粒度の細かい情報だけを扱っているという問題もあった。昨今 IDS(侵入検知システム) などのように、様々な情報から知識ベースを用いて抽象度の高いイベントを発生させるシステムが広く使われるようになってきており、様々な複数のイベントを音声で通知する機能が求められるようになった。

加えて、これまでの stetho の出力の品質は、サンプリング音の品質に依存するところが大きい。これは設定記述の抽象度の低く、発生できる MIDI イベントの記述が脆弱であることに起因する。

[1] http://www.aec.at

4 stetho システムの拡張

4.1 概要

1996 年以降、stetho は長期に渡り定常運用がなされてきた。しかし同時に問題点も指摘されており、それを改善する試みも行ってきた。ところが、これらは実際の運用には反映されず、ことごとく立ち消えた形となっている。この原因はパッケージ作成とリリースエンジニアリングにあると考え、stetho の機能を整理し必要な箇所は拡張し、一般向け配布を意識した修正とパッケージ化を目指すことにした。

具体的には、GNU Autoconf への対応による移植性の向上と導入の簡易化、昨今のサウンドデバイスへの対応、設定ファイルの記述の拡張による表現の多様化、制御プログラムの改良などである。また、トラフィック以外の情報もイベントして処理できるように、外部から接続可能な拡張ポートを用意する。拡張後の stetho システムの構成は図 4 のようになる。

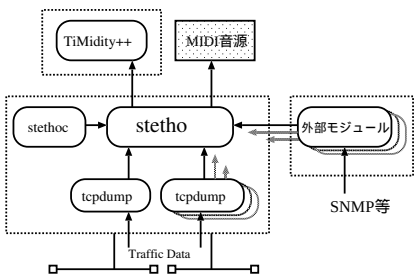

図 4: 改良後の stetho システム

4.2 出力デバイスの拡張

netsound では、ソフトウェア音源に改造した timidity 0.2i を採用していた。timidity 0.2i はフリーソフトウェアであるが既に開発は停止し、1999 年からは開発を引き継いだチームが TiMidity++[8] という名

称で継続して改良を続けており、TiMidity++ 2.x からリアルタイムのイベントをネットワーク経由で読み込み、再生できるようになった。これに対応するための改造を stetho に施した。結果的に、従来の timidity への独自改造の保守を行う労力を削減できることになった。

また、UNIX 用サウンドドライバとして著名な、OSS[9] のシーケンサーデバイスへの対応も行った。従来通りの raw MIDI デバイスと、シリアル MIDI も利用できる。OSS 対応は PC-UNIX を意識したものであるが、RS-232C 入力に対応した MIDI 音源があれば、RS-232C 端子をもつ機器上で stetho を利用できる。この結果として、stetho はほぼすべての UNIX プラットフォームで利用可能になったと考えている。

4.3 設定ファイルの記述の拡張

NetSound では、音楽家の手によるサンプリング音を用いたことによって、作品としての完成を見たが、多くの利用者が stetho を使う場合は通常の GM 音源や同等の音色を用いることが想定できる。したがって、サンプリング音の編集ではなく設定ファイルの記述の段階で多様な表現が出来たほうが望ましい。そこで、デバイスの厳密な制御と表現の多様化を目標として設定ファイルの文法の拡張を行った。拡張した点は以下である。

- 一つのパケットパターンに対して、複数トラック (和音) の音を割り当てられる。
- デバイスごとの初期化シーケンスを記述できる。
- 外部起動するプロセスを tcpdump 以外にも複数指定できる。
- 拡張ポート, 外部プロセスごとのパターンマッチを記述できる。
- 連続してイベントが発生した場合に、重ねて発音するか否かを最大発音数を含めて指定できる。

音の発生は入力に対する行単位のパターンマッチという旧来の方針を踏襲した。再設計の過程では、libpcap[10] を用いて tcpdump の機能を stetho 内部に組み込む方向も検討された。これは、tcpdump 内での処理を排除できるという利点がある。しかし文字列に対するパターンマッチを行うという方針と、tcpdump 以外の外部プロセスや後述する拡張ポートの実装が容易であるという理由から、従来通り tcpdump を呼び出すこととした。

拡張後の設定ファイルの記述例を図 5 に示す。

```
# #から行末まではコメント
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 初期化イベント
end

define 1
 # define で音のパターンを定義する。番号を指定。
a CH @2 v110 k110
 # 先頭が小文字の場合はトラックの初期化用 MML
b CH @14  v110
A o4c4,110 d4,110
 # 先頭が A~K の大文字の場合は通常の MML。
B o3e4,110 f4,110
end

when /.*\.(80|8080|http)[: ].*/
  # when の後に tcpdump の出力に対するパターンを記述。
play 1        # play で再生する音の番号を指定。
mono|poly 4   # mono か poly を指定する。poly の場合は
              # 同じパケットが複数来たら、その度に
              # 音を鳴らす。数字は最大同時発音数。
              # mono の場合は一度に一音しか鳴らさない。
              # 但し発音中に同じパケットが流れたら
              # もう一度繰り返す。
vshift +10    # パケットの流量に応じて volume を変化。
nshift +1     # パケットの流量に応じて note を変化。
end
```

図 5: 設定ファイル記述例

4.4 入力情報の拡張

外部に観測モジュールを追加するために、TCP での接続を受け付ける機能をつける。外部のモジュールからの入力も、tcpdump 同様行単位のパターンマッチを行う。イベントの発生時刻は 1 行分の入力を受け付けた時点の stetho での時刻になる。このため、通常は外部モジュールと stetho の間には、ネットワーク上の遅延やそのゆらぎが少ないことを前提としている。

5 NetSound の運用と展開

2000 年末現在 NetSound は、これまで運用してきた組織の改組に伴い観測点を失ったため、サンプル音声のみの提供となっている。NetSound のあらたな運用基盤として、2001 年以降は東京芸術大学芸術情報センターでの運用を予定している。

NetSound を開設した 1996 年当初と異なり、現在のほとんどの LAN には Ethernet スイッチが用いら

れている。このため、あるセグメントのトラフィック全てを観測するという方法が取りにくくなった。そこで、対外ルータを通過するトラフィックを観測するという方針に変更するものとした。

東京芸術大学は、学内はプライベートアドレスを用い、NAT ルータを経由して SINET に 6Mbps で接続されている。ここで NAT ルータの外側のトラフィックを観測するものとする。NAT の外側のため、観測した情報からは内部ネットワークのどのホストが外部のどこと通信しているか判別できず、結果的に利用者のプライバシー保護に繋がる。

同時にシステム構成を変更し、FreeBSD 3.4 をインストールした 2 台の PC を利用するものとする。2 台の OS を統一したのは、管理作業を軽減するためである。RealProducer(旧 RealEncoder) は Linux エミュレーション上で動作させる。標準の FreeBSD では RealProducer は動作しなかったが、OS 側に修正を加え動作するようにした[2]。

新たな NetSound のサイトは論文執筆時点では準備中であるが、2001 年第一四半期中の公開を目指している。

6 評価と考察

6.1 拡張機能の評価

従来の NetSound では、音楽家が音を WAV 形式で作成し、再生するための音源を生成していた。拡張後の stetho によって、音の重ね合わせや旋律の作成が設定ファイルの変更だけで容易に行えるようになった。

拡張ポートを利用することで、SNMP や IDS からのイベントも受け取れるようになった。この機能は現在のところ試験的に追加されたものであるが、外部モジュールの作成は容易であり、今後はいくつかのモジュールの試作を行っていきたい。

またパッケージについては移植性の向上が図れたことから、今後 stetho を広く配布し多くの人に利用してもらえる用意が整ったと考えている。

6.2 音声割り当て手法の考察

音声割り当ては大別して連続的に発生するパケットと、断続的に発生するパケットとに分けられる。前者のパケットに対しては、弦楽器などの減衰の長い音や、アルペジオの連続などが適している。後者のパケットに対しては、ベルなどの音が適している。また、全体として不協和音にならないように音を割り当てなければならない。

さらに、これらのパケットの発生傾向は観測場所によって大きく異なる。例えば前述の東京芸術大学のトラフィックを試験的に観測したところ、パケット数にして HTTP が約 22%, FTP が約 72%と、そのほとんどを占める。その他に POP と、図書館の OPAC サービスを利用するための TELNET を合わせて 5%程度になる。HTTP と FTP に関しては表現を多彩にするために、パケットの種類以外に送受信アドレスなどの情報を用いて音に変化をつけた方が良い。

stehto での音の割り当てを行う際には、これらの条件を加味する必要があるが、tcpdump の出力結果から適度に適切な設定ファイルを自動生成する機構を作成し、この作業を支援することを考えている。

6.3 関連研究との比較

WebMelody[11] は、WWW サーバのログを音声に変換する。ログの内容に応じて、SMF を再生する。聞いていて疲れない音を生成するように工夫して 2 いるが、システム監視ツールとして見た場合監視対象が限定されている。また、stetho ほど多様な音源に対応していないという差異がある。

Peep[12] は、観測部と発音部が分離され、ネットワーク経由で通信する。ログに対するパターンマッチや、CPU 負荷などの条件に応じて音をだせる。stetho でこのような機能は外部モジュールに依存する形になる。Peep は MIDI イベントではなく PCM 音を再生するため、動的に変化する情報を音声として表現しにくいといった欠点がある。

7 今後の課題

stetho の今後の課題について、三点を挙げる。

最初は、表現力の向上である。現在の stetho では、時系列での変化もしくは累積などを表現できない。トラフィック量の変化については擬似的に表現可能にしているが、例えば「あるパケットが一時間内に流れた合計が一定量を超過したら音を発生させる」などの表現ができない。設定ファイル中で「演算」の記述もできないため、トラフィック情報になんらかの演算処理を施した結果に応じた発音もできない。この改良については、設定ファイルの表現力の向上を図る方針と、外部モジュールとして実装する方法の二通りが考え

[2] http://www.ohnolab.org/~kimoto/freebsd/

られる。

つぎに、自動作曲システムという方向性の検討である。いわゆる自動作曲システムの中には、作曲理論に基づいてコード進行や旋律の遷移の情報を知識ベースとして持ち、乱数などのイベント列に応じてコードや旋律を自動生成するものがある。このようなシステムの入力情報として、ネットワークトラフィックを用いる試みを検討している。これは、先に挙げたstethoの目標のうちの、「音楽として心地よいものである」ことを目指したものである。

最後に、stethoが発生する音の認知科学的評価が挙げられる。Algorithmic Compositionと呼ばれる分野では、フラクタル集合や円周率などの数列を音楽に変換する試みなどが一般的に行われている。近年の研究でネットワークトラフィックは大局的には自己相似性を持つことが知られており[13]、stethoでの試みはAlgorithmic Compositionの典型例一つであると言える。しかしstethoが発生する音声は単なる音の羅列ではなく、管理情報に対応した意味づけができるという点で、通常のAlgorithmic Compositionとは異なる。そこで今後は、stethoが発生した音声が「心地よいか」という点だけでなく、「聞いた人が意味づけを行い、大局的な管理情報として認知できるか」という評価を行う必要があると考えている。

8 おわりに

管理情報可聴化システムであるstethoとこれを用いたNetSoundのこれまでの経緯を述べ、stethoの拡張について述べた。この拡張によりstethoはより使いやすいシステムになった。今後は東京芸術大学でのNetSoundの運用を行うとともに、stethoを配布し多くの人に利用してもらうことを目指す。特に芸術大学という活動基盤を得たことから、多くの音楽専門家の協力が期待できるものと考えている。

謝辞

東京工業大学 旧大野研究室の過去のstetho, NetSoundの開発運用関係者に謝意を表する。また、Sensoriumの西村 佳哲氏にはNetSoundの運用にあたり、山口 優氏には音声の製作にあたり多大な協力を得た。東京芸術大学 芸術情報センター長の伊藤 隆道氏と助手の古積 新一氏には、新たな実験運用基盤を与えて頂いた。ここに感謝する。

参考文献

[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] 成田哲也 and 大野 浩之, "stetho:ネットワークトラフィック可聴化システム", 社団法人 情報処理学会 マルチメディア通信と分散処理研究会, Nov. 1995, DSM-951150

[6] 成田哲也 and 大野 浩之, "ネットワーク可聴化システムの利用例", 第 52 回 (平成 8 年後期) 全国大会 公演論文集 (3), pp. 473, 社団法人 情報処理学会,

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

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

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

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

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

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

[13] 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