平成 13 年度

学士学位論文

MRTG を用いた
トラフィック監視システムの拡張

An extention of traffic monitoring system

using MRTG

1020324　　前田 明日香

指導教員　　菊池 豊

2002 年 2 月 8 日

高知工科大学 情報システム工学科

要 旨

MRTG を用いた
トラフィック監視システムの拡張

前田 明日香

情報通信ネットワークの急速な拡大と重要性の高まりの中で、ネットワーク運用・管理は重要になっている。ネットワーク管理で有用なツールに MRTG がある。MRTG は、トラフィック量をチェックし、その変化を見やすいグラフにして WWW で表示することができる。

しかし MRTG は、マシンのトラフィック量の異常判定ができないことや定期的にしか値を取得することができない。よって、本研究では、通信機器の警報通知機能と最新情報表示機能を提案する。実装として前者は、マシンの異常トラフィックの警報をメールで通知する機能であり、後者は、最新のトラフィック量を取得し、表示する機能である。

結果は、警報通知機能は、閾値を超えると管理者にメールが通知されることが実地で確認した。また、最新情報表示機能は、MRTG のグラフに加えて、最新の値の表示を実現した。

今後の課題は、警報通知機能は、閾値をどのように定めていくのかである。これはトラフィック量を長期に渡り、収集していかなければならない。また、最新情報表示機能は、使いやすいものなのか、知りたい情報なのか調べ、機能をさらに拡張していくことが今後の課題となる。

キーワード　ネットワーク運用・管理, MRTG, PHP

Abstract

An extention of traffic monitoring system using MRTG

MAEDA Asuka

In the rise of a rapid expansion and importance of an information communication network, network employment and management are important. MRTG is in a useful tool by network management. MRTG can check the amount of traffic, can make the change legible graph, and can display it by WWW.

However, the unusual judging of the amount of traffic of a machine cannot be performed, and MRTG cannot acquire a periodical value. Therefore, in this research, the notice function of an alarm of communication apparatus and the newest information display function are proposed. As mounting, the former is a function which notifies the alarm of the unusual traffic of a machine by mail, and is a function which the latter acquires the newest amount of traffic and displays.

In the result, when the Threshols dose was exceeded, that an administrator is notified of mail checked the notice function of an alarm by practice. Moreover, in addition to the graph of MRTG, the newest information display function realized the display of the newest value. A future subject is about how the notice function of an alarm defines a Threshols dose.

This must collect the amounts of traffic over a long period of time. Moreover, the newest information display function investigates whether it is the information which wants to know whether it is what it is easy to use, and it becomes a future subject to

extend a function further.

key words network employment and management,MRTG,PHP

目次

図目次

表目次

第 1 章

はじめに

情報通信ネットワークの急速な拡大と重要性の高まりの中で、ネットワーク運営・管理は重要になっている。

そこで本研究では、管理者に、簡単にネットワーク管理できるようにわかりやすい情報、見やすい表示で情報を公開することを目的とする。ネットワークの負荷を監視する MRTG を用いる。

MRTG の改善点を把握し、拡張を目指す。MRTG の拡張としては、通信機器の警報通知機能と最新情報表示機能を提案する。そして、システムの実装と結果について報告する。

まず、この章ではネットワーク管理による重要性、本研究の目的、そのためにはどのような既存ツールを用いるのかを述べる。第 2 章では、MRTG の説明と改善点、第 3 章では、MRTG による拡張の提案と提案による実装を述べる。第 4 章では、結果と考察を行う。そして、第 6 章で本研究のまとめについて述べる。

1.1　ネットワーク管理の重要性

この節では、ネットワーク管理の必要性、ネットワーク管理に対する目的、トラフィック監視の既存ツールについて説明する。

1.1.1　背景

情報通信ネットワークにおいて、ネットワークの運用・管理は重要である。ネットワークの運用・管理とは、システムを常時監視することで、停止を未然に防ぎ、万一停止した場合

には迅速に対応するということが求められる。

それは、コンピュータ・ネットワークの利用が大きくなり、そのためネットワークやサーバが停止すると業務が止まってしまうことやさらに多大な信頼を無くすことになるからである。

こういった社会状況のなかで、コンピュータ・ネットワーク監視の重要性は非常に大きくなっているということがいえる。

1.1.2　ネットワーク管理に対する目的

ネットワーク管理の重要性に着目して、本研究では、管理者に使い勝手の良い仕組みを提供し、簡単にネットワーク管理ができることを目的とする。

ネットワーク管理ができるように、ルータなどの通信機器の情報を以下のように管理者に提示する。

- わかりやすい情報
- 見やすい表示
- 使い勝手の良い仕組み

1.1.3　トラフィック監視

少しでも目的を満たせるようなツールはないのかと考えた。そこで、これまでのネットワーク状況を見ることができるトラフィック監視 MRTG (Multi Router Traffic Grapher), Seafelt, PyNG (the Python Network Grapher), RRDTools+ (Remstat, Cricket, ORCA, NRG)[*1]の既存ツールなどがあげられる。

MRTG というツールが一般的に使われいるので、本研究では、MRTG を用いてネットワーク監視することにする。MRTG は、WEB 画面でトラフィック監視、フリーソフトなので小規模な LAN でも、手軽にネットワーク監視、アレンジが可能であり、トラフィックの

[*1] http://www.soi.wide.ad.jp/iw99/iw99_tut/slides/15/

状態を統計で見れるという利点がある。MRTG については、次の章で詳しく説明していく。

第 2 章

MRTG によるトラフィック監視

この章では、トラフィック監視のツールである MRTG について詳しく説明していく。まず、MRTG の構成として SNMP、MIB、UCD SNMP について説明し、次に MRTG について述べる。最後に、MRTG の改善点を挙げる。

2.1 SNMP

この節では、ネットワーク管理プロトコルである SNMP について説明する。

2.1.1 SNMP の概要

SNMP は、TCP/IP プロトコル群で標準化されたネットワーク管理の機能を提供するプロトコルで、1989 年から 1990 年に RFC 化された [1]。ネットワークの規模が小さいうちは、ネットワーク管理者の人手によって管理できるかもしれないが、ある程度大きくなると、人の手にはおえないということからこのプロトコルが提唱された。

SNMP は、ネットワークの構成や、ハブ、ルータ等のネットワーク機器に関しての管理情報のやり取りに使用される。

SNMP の動作

SNMP (Simple Network Managing Protocol) ベースのネットワーク管理システムは「マネージャ」、ネットワーク管理対象機器は「エージェント」、および管理情報ベース

(Management Information Base: MIB) と呼ばれる「データベース」の 3 要素で構成されている (図 2.1)。エージェントが保持しているデータベースが MIB である。MIB については、第 2.2 章で説明する。

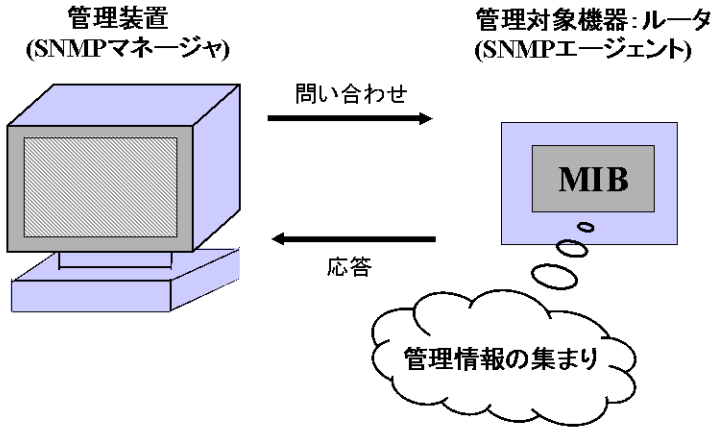

図 2.1　SNMP の動作

SNMP バージョン

1. SNMPv1

   現在利用されているエージェントの多くは SNMPv1 を実装している。もともとは単に SNMP と呼ばれていた。SNMPv1 では、5 種類のメッセージ (プロトコルデータ単位：PDU) を使った基本的なネットワーク管理フレームワークが定義されている。

2. SNMPv2

   SNMPv1 の欠点を補い SNMP を拡張するために、努力が重ねられてきた。

3. SNMPv3

   SNMPv2 に欠けていた部分を補うことにあてられており、その他は SNMPv2 を継承している。

SNMP のプロトコル

マネージャとエージェント間の情報交換は、以下の 5 つのプロトコルである。図 2.2 を参照。

- マネージャからエージェントへの問い合わせ要求および、設定要求
  - Get Request
  - Get Next Request
  - Set Request
- エージェントからマネージャへの応答および、イベント通知
  - Get Response
  - Trap

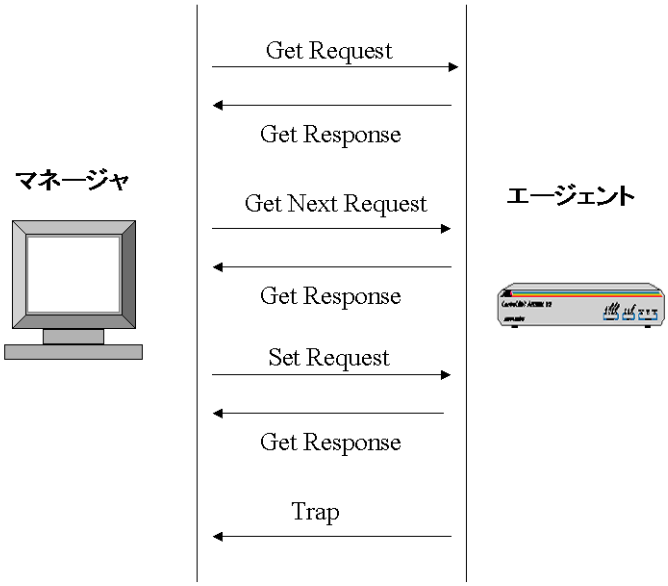

図 2.2　SNMP のプロトコル

SNMP コミュニティ

ネットワーク管理システムである SNMP エージェントは特定の SNMP マネージャだけから要求を受け付けるようになっている。SNMP では、そのための認証を” コミュニティ” によって行う。コミュニティの名前は、コミュニティ名という。コミュニティ名はパスワードの役割を果たす。マネージャはエージェントに問い合わせるときに、コミュニティ名を渡す。エージェントはコミュニティ名を判断し、それが正当なコミュニティ名であると、その操作を実行 (Set や Get) する。

2.2 MIB

SNMP マネージャがエージェントから取得または操作できる管理対象オブジェクトは、MIB (Management Information Base: 管理情報ベース) というデータベースで定義されている。MIB は情報の倉庫であり、一般的に数千のオブジェクトが記述されている。

管理情報データベースの構造は、SMI (Structure of Management Information) で定義されている。

2.2.1 標準 MIB と独自 MIB

ネットワークデバイスが、SNMP プロトコルとの互換性を持たせるためにサポートしなければならないオブジェクトは、共通の標準 MIB として定義されている。最も広く実装されている標準 MIB は、MIB-Ⅱ である。

このほかにも、さまざまなネットワークコンポーネントや技術に合わせて、多くの MIB が開発され、RFC として文書化されている。

さらに、多くのベンダが独自の MIB を開発している。これらの MIB は、主に標準 MIB には用意されていない追加機能を提供するために開発されている。

2.2.2 MIB ツリーとオブジェクト名

MIB は、大量のオブジェクトがいくつかのグループに分けてまとめられている。この構造を使うことで、SNMP マネージャは簡単で簡潔な方法で、MIB オブジェクトを参照することができるようになっている。

MIB オブジェクトは、階層型のツリー構造に編成されており、最上部がツリーのルートになっている。図 2.3 は、標準 MIB の構造の一部を示している。

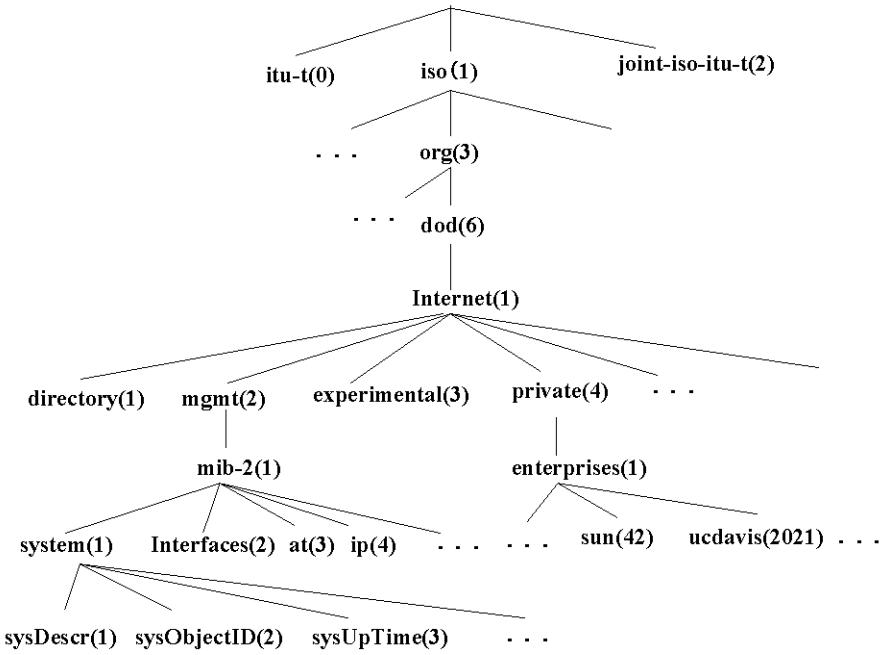

図 2.3 MIB の構造図

ルートから別れている 3 つのノードのうち、itu-t(0) は国際電気通信連合の電気通信標準化部門が管理し、iso(1) は国際標準化機構が管理している joint-iso-itu-t(2) は両者が共同で管理している。利用する MIB は、インターネットに割り当てられたノード internet(1) である。

internet(1) のうち、mgmt(2) の mib-2(1) ノードが標準 MIB-Ⅱ になる。また各ベンダの独自 MIB は private(4) の enterprises(1) の下に、ベンダごとのプライベートエンタープライズコードで割り当てられている。

MIB 内のオブジェクトを参照するには、

1. ツリーのルートから始まって、目的のオブジェクトに至るまでの木を下にたどる
2. それまで通ったオブジェクトを順に並べる
3. オブジェクトをドット (.) で区切って指定する
4. オブジェクトには、数字と名前のどちらでも使うことができる
   - 例

     iso.org.dod.internet.mgmt.mib-2.system

     1.3.6.1.2.1.1

このようにして、それぞれのオブジェクトは、ドットで区切られた一連の英単語または数字の羅列で表現される。これを、オブジェクト識別子 (OID: Object Identifier) という。

2.3 UCD SNMP

この節では、UCD SNMP がサポートされている独自 MIB オブジェクトや UCD SNMP パッケージに含まれるコマンドに関する説明をします。

2.3.1 UCD SNMP の概要

UCD SNMP パッケージには、ほかの SNMP パッケージと同様、MIB オブジェクトの取得と設定という機能を持つ。つまり、標準 MIB に対する基本操作 (Get,GetNext,Set,Trap) がサポートされている。

UCD SNMP エージェント (snmpd) の機能を簡単にまとめると、次のようになる。 [1]

- MIB-Ⅱ オブジェクトのサポート

  標準 MIB-Ⅱ オブジェクトをサポートしている。
- 独自の UCD-MIB による管理機能のサポート

  独自の追加 MIB オブジェクトをサポートし、エージェントを通じて UNIX ホストのさ

まざまな情報を入手できる。

- 外部コマンド実行による機能拡張

  特定の SNMP 要求をトリガーして、外部コマンドを実行する機能をサポートする。

- 自由度の高い設定ファイル

  設定ファイルの融通性の高い設定オプションやディレクティブを使って、動作状況やエージェントの動作を制御できる。

- SNMPv2/v3 サポート

  SNMPv2 および SNMPv3 プロトコルをサポートしている。

UNIX ホストの管理は、UCD-MIB を使って、システムの動作状況を表す重要な統計データ (システムの平均負荷など) を監視することができる。たとえば、次のような作業を実行できる。

- システム負荷の監視
- ディスク使用量やメモリに関する統計データの表示
- システムプロセスの一覧表示や制御
- UNIX コマンドやシェルスクリプトの呼び出し
- エージェント情報と状態の監視

UCD SNMP エージェントでは、標準 MIB のほかに独自 MIB が実装されている。独自 MIB オブジェクトへのアクセスは、private.entrprises サブツリーで行われる。

UCD-MIB ツリーの完全なパスは、.iso.org.dod.internet.private.enterprises.ucdavis (.1.3.6.1.4.1.2021) から始まる。図 2.4 に示す。

UCD-MIB の主なオブジェクト

UCD-MIB の主なオブジェクトは、図 2.5 のとおりである。これは、ucdavis サブツリーを起点としたオブジェクトの階層を示している。

UCD SNMP ユーザーコマンド

UCD SNMP を用いて、SNMP 対応のネットワーク機器を監視し、管理することができる。たとえば、snmpget コマンドを使用することで、 ネットワーク機器のネットワークインターフェイスを監視し、それが正常に動作しているかを確認することができる。UCD SNMP パッケージは、以下のコマンドを提供する。

snmpget, snmpgetnext, snmpwalk, snmpbulkget, snmpset, snmptable, snmpdelta, snmpstatus, snmptest, snmpnetstat, snmpdf, snmpconf, snmptranslate

基本的な構文

snmpcomand [共通オプション] ホスト [コミュニティ] コマンドパラメーター

使用する構文の説明は、表 2.1 に示す。

(例) MIB オブジェクト system.sysContact を参照する Get リクエストの内容の場合

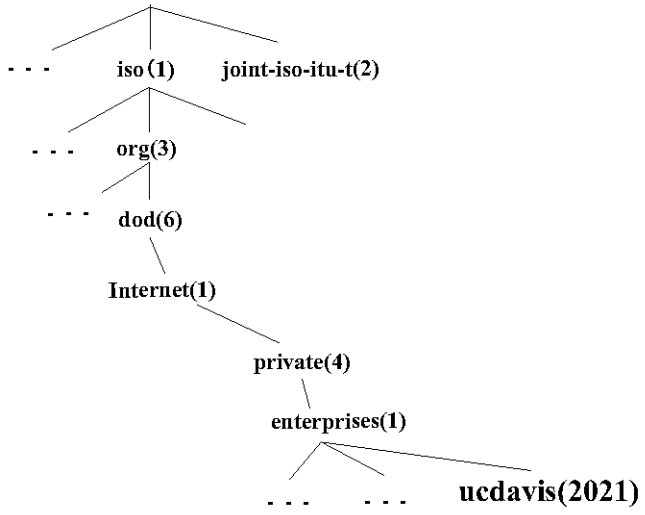

図 2.4 UCD 独自 MIB への MIB ツリー図

%snmpget -d localhost private system.sysContact.0

ダンプ情報

-d オプションを使用すると、SNMP リクエストおよび返答のパケットについて、その内容が表示 (ダンプ) される。パケットの内容は 16 進数と ASCII コードで表示される。

トラップ関連のプログラム

snmptrap コマンドは、これまで紹介した SNMP コマンドと同様にユーザーコマンドであるが、SNMP マネージャに情報を送信するためのものである。

snmptrap コマンド

特定の SNMP マネージャに SNMP のトラップを送るためのコマンドである。シェルスクリプトなどのプログラムでこのコマンドを使用することで、エージェント側からマネージャに情報を送ることができる。

snmptrap [共通オプション] エンタプライズ OID エージェント 汎用トラップ
固有トラップ アップタイム [オブジェクト ID　データ型　値 ・・・・・]

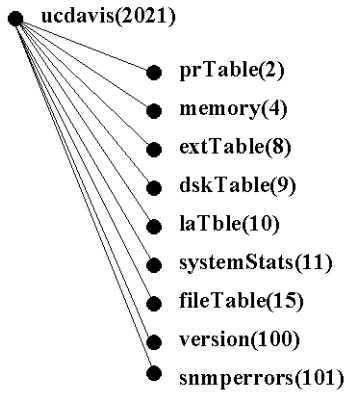

図 2.5　UCD-MIB のサブツリー図

snmptrap コマンドで使用する構文の説明は、表 2.2 に示す。

2.4 MRTG

ネットワークの監視ツールのひとつに MRTG(Multi Router Traffic Grapher) というソフトウェアがある。

MRTG は、SNMP を使ってルータを流れるデータのトラフィック量をチェックしてその変化を見やすいグラフにして WWW 上で表示してくれるツールである。

2.4.1 MRTG の動作

MRTG の利用環境は、監視される機器、MRTG を実行するサーバ、その結果を閲覧する端末の 3 つで構成されている。ネットワーク装置の監視には、SNMP (Simple Network Management Protocol) というネットワーク監視用プロトコルが使われている図 (2.6)。

2.4.2 MRTG の特徴

MRTG には、以下の特徴がある [2]。

- ほとんどの Unix プラットフォームと WindowsNT 上で稼動

表 2.1 使用する構文の説明

snmpcomand	コマンド名
共通オプション	コマンドの動作の制御などに使用。-d,-h など
ホスト	対象エージェントの IP アドレスまたはホスト名
コミュニティ	エージェントに設定されいるコミュニティ名
コマンドパラメーター	指定したエージェント内部の操作対象となる MIB オブジェクトの識別子 (オブジェクト ID) を指定

- 計測結果を HTML ファイルおよびグラフに出力してくれる
- MRTG が動作しているコンピュータ上で Web サーバを動かしておけば、離れたところからネットワーク機器の状態を見ることができる
- 独自に SNMP を実装。外部の SNMP Package は不要
- 定期的にログをサマリーするデータ管理を行っており、ログファイルのサイズが大きくならない
- 半自動のコンフィグ作成ツールが付属
- ２系列のデータを基に集計を行い、グラフ化する
- 日・週・月・年ごとにデータを集計した WEB ページを結果として生成する
- 生成する WEB ページの外観は、詳細に設定することが可能
- コンフィグから index を簡単に生成するツールが付属

詳細

MRTG は、SNMP を使用してルーター上のトラフィックカウンターを読み取る Perl スクリプトと、トラフィックデータを収集して監視しているネットワークのトラフィックをグラフにする C のプログラムで構成されている。これらのグラフはどんな Web ブラウザからでも読めるように、WEB ページに埋め込まれる。

表 2.2 snmptrap コマンドで使用する構文の説明

エンタプライズ	OID トラップを生成したネットワーク管理サブシステムを指定する
エージェント	トラップを送信するホストを指定する
汎用トラップ	あらかじめ定義されている SNMP トラップを指定する
固有トラップ	トラップの性質をより詳しく表す値を指定する
アップタイム	デバイスが最後に初期化された時点からトラップの生成時までの経過時間として使う

MRTG は日ごとの詳細なグラフに加えて、それぞれ過去 7 日間、4 週間、12 ヶ月のトラフィックを視覚化する。これは、MRTG がルーターから収集してきた全てのデータをログとして保持するため可能となっている。このログは自動的に整理されるため、時間の経過とともに肥大することがなく、過去 2 年間におけるトラフィックに関するデータを保持することができる[*1]。

2.4.3　MRTG の改善点

MRTG の改善点を挙げる。

異常時の場合の改善点 1

- マシンのトラフィック量の異常判定ができない。

  MRTG は、データをグラフ表示にするだけで、マシンのトラフィック量が異常であるかどうかの判定ができない。

  MRTG の表示の改善点 2

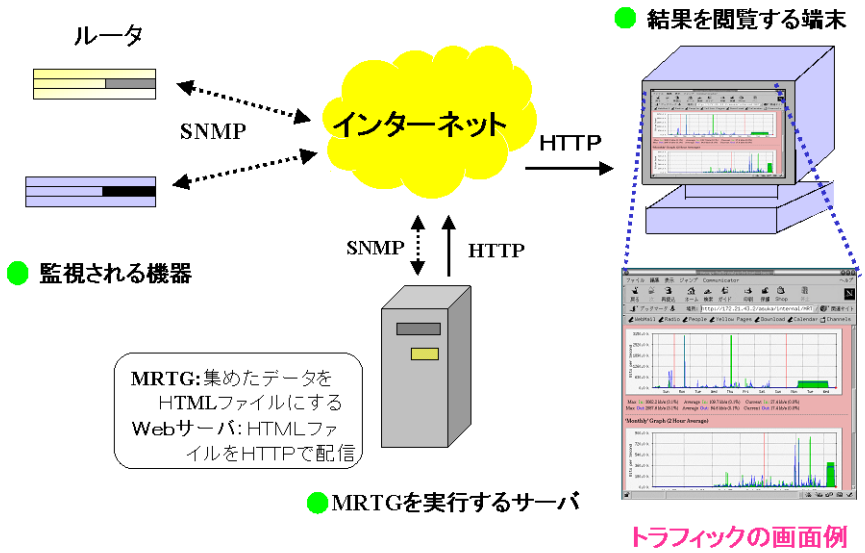

図 2.6　MRTG の動作

[*1] http://www.mrtg.jp/doc/mrtg.html

- MRTG は、定期的にしか値を取得することができない。

  MRTG は、cron で定期的に実行されており、cron は 1 分以内での実行はできないといえる。

第 3 章

MRTG の拡張

この章では、第 2.4.3 項で述べた MRTG の改善点に対して、警報通知機能と最新情報表示機能を提案する。また、提案した機能の実装を行う。

3.1 機能の提案

機能の提案を述べる。

3.1.1 警報通知機能

第 2.4.3 項で述べた改善点 1 を基に、警報通知機能の提案をする。

警報通知機能とは、マシンの異常トラフィックの警報をメールで通知するものである。これは、MRTG に閾値を設定し、トラフィック量が閾値を超えるとメールを通知する機能である。それにより、障害を予側できるようなシステムにする。障害を予測することで、トラブルを未然に防ぐことができる。

3.1.2 最新情報表示機能

第 2.4.3 項で述べた改善点 2 を基に、最新情報表示機能を提案する。

最新情報表示機能は、最新の値を取得し、表示する機能であり、表を用いて表示する。また、更新ボタンを押すと、最新情報を表示でき、さらに定期的に自動で更新できる。それによって、統計情報と現在のシステムの状況をより厳密に知ることができる。

3.2　機能の実装

この節では、第 3.1 節に基づいて、警報通知機能と最新情報表示機能についてそれぞれ実装を行う。

3.2.1　実装環境

実装環境は、表??に示す。

表 3.1　実装環境

OS	FreeBSD 4.3-RELEASE
IP	172.21.43.250
WWW	Apache-1.3.22, PHP-3.0.18
SNMP	ucd-snmp-4.2.2

3.2.2　警報通知機能の実装

MRTG で収集している値が一定の値を超えたら管理者にメールで通知する機能について実装する。

警報メール送信スクリプト作成

警報をメールで通知するスクリプトを作成する。Perl で、スクリプトを作成する。

まず、日本語表示に設定する。シェル上で操作。

Alt-x set-file-coding-system(enter)

File coding system : euc-jp-unix(enter)

warning.pl の作成

```
#!/usr/bin/perl
 open(SENDMAIL,"|/usr/sbin/sendmail -t -oi");
  print SENDMAIL "From: asuka\@kikuken.org\n";
  print SENDMAIL "To: asuka\@kikuken.org\n";
  print SENDMAIL "Subject: MRTG の警報通知\n\n";
  print SENDMAIL "監視対象機器が設定した閾値を超えました。\n";
  print SENDMAIL "\n";
  print SENDMAIL "監視対象名: ".$ARGV[0]."\n";
  print SENDMAIL "設定した閾値: ".$ARGV[1]."\n";
  print SENDMAIL "現在の値: ".$ARGV[2]."\n";
close(SENDMAIL);
```

入力トラフィックと出力トラフィックで見分けをつけるため、2 つスクリプトを用意した。出力トラフィックには、MRTG の警報通知 (out) と付け加えた。

MRTG の閾値の設定

MRTG が生成するファイルに追加する設定項目は、閾値そのものを設定する ThreshMaxI・ThreshMaxO と、それを超えた場合に起動するプログラムを指定する ThreshProgI・ThreshProgO である [3]。

まず、自分のマシンのトラフィックの監視を行う。MRTG を作成したファイルを thinkpad.cfg とする。設定した閾値を超えた場合に、メールが通知されるかどうか確か

めるため、入力トラフィックが 400000 を超えたら危険とみなし、警報を発するように設定する。thinkpad.cfg に追加する。出力トラフィックも同様に設定する (図 3.2.2)。

```
ThreshMaxI[172.21.43.250_9]: 400000
ThreshMaxO[172.21.43.250_9]: 400000
ThreshProgI[172.21.43.250_9]:/home/lab/kiku02/asuka/Warning/Perl/warning.pl
ThreshProgO[172.21.43.250_9]:/home/lab/kiku02/asuka/Warning/Perl/warningout.pl
```

図 3.1　閾値設定

MRTG の実行を、crontab ファイルに追加して実行する。これで、しばらく閾値メッセージが届くか確認する。

3.2.3　最新情報表示機能の実装

最新の値を取得し、表示させるため実装を行う。

最新の値を取得するには、ucd-snmp を用いて確認する方法がある。しかし、ucd-snmp は、毎回コマンドを実行して確認しなければならない。そこで、MRTG が生成した Web ページと一緒に表示するために、PHP[*1]を使用する。

PHP(図 3.2) は、HTML 文書に直接埋め込めことができる点、SNMP をサポートしている点から採用した [4]。

PHP の特徴

- スクリプト

  コンパイルを必要としないので、ソースを修正してすぐにテストというサイクルを繰り

[*1] http://www.php.net

返すことができる。

- HTML 文書への埋め込み型言語 (表 3.2)

  ファイル全体をスクリプトとして作成する必要はなく、HTML 文書のうちの必要な部分だけを PHP で書くといった使い方ができる。

- 確実なエラーハンドリング

  文法エラーが発生した場合には、発生した行番号やエラー内容などを表示するための HTML 文が自動的に生成され、エラー情報がブラウザ上に表示されるのでデバッグが容易である。

- Cのような文法

  If、for、while、do-while など、C 言語などの制御構文が用意されている。

- Perl のような機能

  Perl のような文字列演算子や関数が豊富で、リスト変数、連想配列、多次元配列もサポートされている。

- Apache のモジュールとして動作

  無駄なリソースを消費せず、処理が高速である。

- 各種データベースのインタフェース

  Oracle や、PostgreSQL、MySQL、mSQL といったフリー DBMS へのイ] ンタフェースを標準で備えている。データベースを使うと、目的のデータを高速に見つけることができる。

PHP で使える ucd-snmp コマンド

表 3.3 にまとめる。

MRTG はデフォルトでは .html ファイルを生成する。Extension オプションを使用して MRTG に違う拡張子を使うよう指示することができる。拡張子を php3 にして、PHP タグを出力に含ませる。

thinkpad.cfg ファイルに Extension[172.21.43.250_9]: php3 と追加する。これで、ファイルに php のコマンドも挿入できる。挿入の仕方は、スペースを空けずに (PageTop などの後に)、<?php 〜 ?> と php 文を入力。すると、172.21.43.250_9.php3 と php のファイルが生成される。thinkpad.cfg ファイルの設定を表 3.2.3 に示す。

最新の値を表示する場合、cfg ファイルに PHP を作成し埋め込ませる方法と、個々で PHP を作成する方法の 2 通りがある。前者は、改行を入れずに作成しなければならない。よって本研究では、後者の個々で作成する方法を用いたので、この cfg ファイルには、PHP で作成したページに飛ぶようにリンクを貼る。

PHP での実装方法

PHP を用いて取得したい項目を含むファイルを作成する。取得したい項目を PHP を用いて取得する。snmpget を利用してホスト名、コミュニティ名、オブジェクト名を指定する。オブジェクトの中身が一つならばオブジェクトの最後に 0 を指定する。二つ以上ならば snmpwalk でオブジェクトを指定する。

インターフェイスに関する情報を取得するため interfaces サブツリー (.1.3.6.1.2.1.2) を指定する。.1.3.6.1.2.1.2.1 (ifNumber) から ucd-snmp のコマンドである snmpget や snmpwalk で実行し確認して、取得できる OID を設定する。今回の実装で用いたのは、9 個

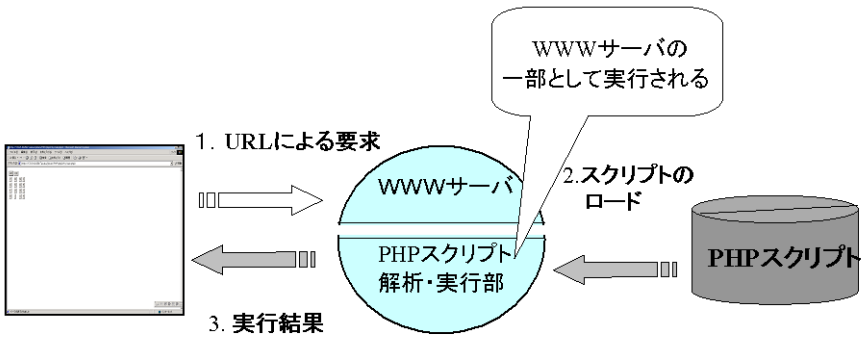

図 3.2　PHP の実行形態

の値である。interfaces オブジェクトの中身は 9 個ずつあったので、snmpwalk で指定する。

```
<?php
$host = "172.21.43.250";
$community = "kikuken";
$sysDescr = snmpget("$host","$community","system.sysDescr.0");
$ifIndex = snmpwalk("$host","$community",".1.3.6.1.2.1.2.2.1.1");
    •
    •
 for ($i=0; $i<count($ifIndex); $i++) {
 print "<tr><th></th>";
 print "<td><center>$ifIndex[$i]</center></td>";
print "</tr>";
?>
只今、<?php echo date("Y 年 m 月 d 日 (D) h:i:s:A"); ?>です。
```

見やすい形として取得した値を表にして表示する。そして、更新時刻も表示する。取得した値の名前と説明をする。これで PHP を用いたファイルで、2 つの情報以外でも情報を即座に見ることができる。

さらに、情報を更新する機能を導入する。そのため cfg ファイルを参照して、情報を定期的

表 3.2　HTML 文書への埋め込み型言語の例

<HTML>
<BODY>
<?php echo(”PHP”)?>
</BODY>
</HTML>

に自動で更新できるように設定する。

```
<!-- Begin Head -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
<TITLE>INTERFACES</TITLE>
</HEAD>
<META HTTP-EQUIV="Refresh" CONTENT="180">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="Cache-Control" content="no-cache">
  PHP のスクリプト
</BODY>
</HTML>
```

Refresh で情報を更新する。180 秒で毎回更新される。

PHP でファイルを作成する前に、interfaces サブツリーのどれが取得できるかどうかマシンで snmpwalk コマンドを使って調べた。

表 3.3　PHP のコマンド

snmpget	SNMP オブジェクトを取得
snmpset	SNMP オブジェクトを設定
snmpwalk	エージェントから全ての SNMP オブジェクトを取りだす
snmpwalkoid	ネットワークエンティティに関する情報ツリーの検索
snmp_get_quick_print	UCD ライブラリの quick_print の現在の設定値を取得
snmp_set_quick_print	UCB SNMP ライブラリで quick_print の値を設定

```
Extension[172.21.43.250_9]: php3
 <TABLE>
   <TR><TD>System:</TD>     <TD>kiku017.kikuken.info.kochi-tech.ac.jp
   in ThinkPad Server</TD></TR>
   <TR><TD>Maintainer:</TD> <TD>asuka@kikuken.org</TD></TR>
   <TR><TD>Description:</TD><TD>ep0  </TD></TR>
   <TR><TD>ifType:</TD>     <TD>ethernetCsmacd (6)</TD></TR>
   <TR><TD>ifName:</TD>     <TD></TD></TR>
   <TR><TD>Max Speed:</TD>  <TD>1250.0 kBytes/s</TD></TR>
   <TR><TD>Ip:</TD>         <TD>172.21.43.250 (dhcp250.kikuken.
   info.kochi-tech.ac.jp)</TD></TR>
 </TABLE><p><?php echo "php"; ?><br><A HREF="../../../../jikken/PHP/
php3/snmp/mrtg/interfaces.php3">インターフェイスに関する情報</A> \\ \hline
\end{tabular}
```

表 3.4 拡張子 php3

表 3.5　interfaces サブツリー (1.3.6.1.2.1.2)

OID	サブツリー名	取得できる
1.3.6.1.2.1.2.1	ifNumber	○
1.3.6.1.2.1.2.2	ifTable	○
1.3.6.1.2.1.2.2.1	ifEntry	○
1.3.6.1.2.1.2.2 1.1	ifIndex	○
1.3.6.1.2.1.2.2.1.2	ifDescr	○
1.3.6.1.2.1.2.2.1.3	ifType	○
1.3.6.1.2.1.2.2.1.4	ifMtu	○
1.3.6.1.2.1.2.2.1.5	ifSpeed	○
1.3.6.1.2.1.2.2.1.6	ifPhysAddress	×
1.3.6.1.2.1.2.2.1.7	ifAdminStatus	○
1.3.6.1.2.1.2.2.1.8	ifOperStatus	○
1.3.6.1.2.1.2.2.1.9	ifLastChange	○
1.3.6.1.2.1.2.2.1.10	ifInOctets	○
1.3.6.1.2.1.2.2.1.11	ifInUcastPkts	○
1.3.6.1.2.1.2.2.1.12	ifInNUcastPkts	○
1.3.6.1.2.1.2.2.1.13	ifInDiscards	○
1.3.6.1.2.1.2.2.1.14	ifInErrors	○
1.3.6.1.2.1.2.2.1.15	ifInUnknownProtos	○
1.3.6.1.2.1.2.2.1.16	ifOutOctets	○
1.3.6.1.2.1.2.2.1.17	ifOutUcastPkts	○
1.3.6.1.2.1.2.2.1.18	ifOutNUcastPkts	○
1.3.6.1.2.1.2.2.1.19	ifOutDiscards	×
1.3.6.1.2.1.2.2.1.20	ifOutErrors	○
1.3.6.1.2.1.2.2.1.21	ifOutQLen	×
1.3.6.1.2.1.2.2.1.22	ifSpecific	×

第 4 章

結果と考察

この章では、実装した警報通知機能と最新情報表示機能について実行した結果を示し、結果をもとに考察を述べる。

4.1 警報通知機能

警報メールが届いた結果を入力トラフィック (図 4.1)、出力トラフィック (図 4.1) に示す。

Delivered-To: asuka@kikuken.org

Date: Thu, 17 Jan 2002 00:31:16 +0900 (JST)

From: asuka@kikuken.org

To: 020324p@ugs.kochi-tech.ac.jp

Subject: MRTG の警報通知

監視対象機器が設定した閾値を超えました。

監視対象名: 172.21.43.250_9

設定した閾値: 400000

現在の値: 411226

図 4.1 入力トラフィックのメール通知

4.1.1　考察

図 4.1、図 4.1 により閾値を超えるとメールが送られてくることを実地で確認した。よって、閾値を設定することにより、マシンのトラフィック量が異常であるかどうかの判定ができるということがいえる。しかし、予想通りメール通知の結果は、どの情報が閾値を超えているいるのか判断することができない。今回は、スクリプトを 2 つ作成し現在の値を表示することで判断することができた。今後は、1 つのスクリプトを作成するだけで判断できるように、詳細に情報を得られるものにしていく必要がある。また、トラフィック量の異常警報を通知するには、閾値を変えて再度試していく必要がある。マシンの異常が起こった時に、メールが送られてこなくなる可能性がある。

今後、閾値をどのように定めていくのかが課題である。トラフィック量を長期に渡り収集し、適正な値を自動で得るなどの方法が有効であると考える。

Delivered-To: asuka@kikuken.org

Date: Wed, 30 Jan 2002 17:56:57 +0900 (JST)

From: asuka@kikuken.org

To:020324p@ugs.kochi-tech.ac.jp

Subject: MRTG の警報通知 (out)

監視対象機器が設定した閾値を超えました。

監視対象名: 172.21.43.250_9

設定した閾値: 400000

現在の値: 873226

図 4.2　出力トラフィックのメール通知

4.2　最新情報表示機能

最新情報表示機能の結果を示す。図 4.3 は今回実際に設定した MRTG のトラフィックの監視の実行結果である。図 4.4 は今回提案した PHP を用いて作成した最新情報表示である。図 4.5 は、図 4.4 で示した表の項目の説明を示している。

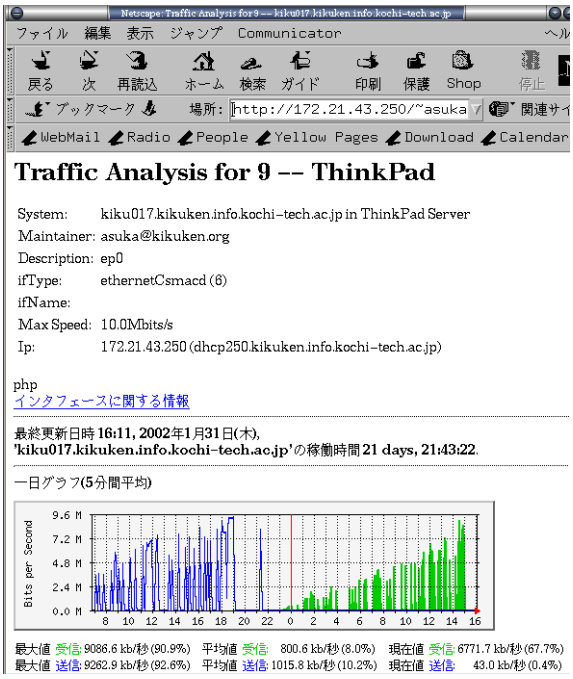

図 4.3　トラフィック監視

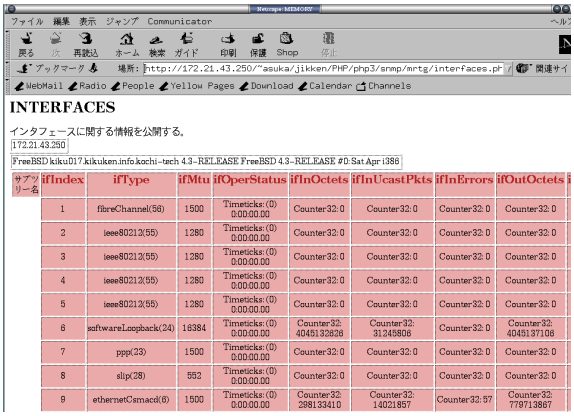

図 4.4　最新情報表示

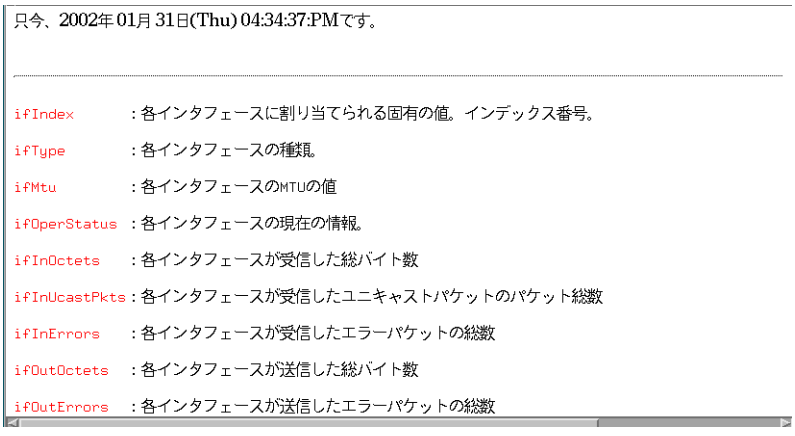

図 4.5　最新情報表示の説明文

また、トラフィック以外の監視も重要であると考え、以下の 6 つも PHP ファイルで作成

した。

- mib-2 サブツリー

  system、icmp、snmp

- ucdavis サブツリー

  memory、laTable、systemStats

ここでも PHP でファイルを作成する前に、mib-2 と ucdavis サブツリーを取得できるかどうか snmpget と snmpwalk で調べた。MIB ツリーには、多くのオブジェクトがあるが、mib-2(表 4.1) と ucdavis(表 4.2) サブツリーに絞って調べた。

表 4.1　mib-2 サブツリー (1.3.6.1.2.1)

OID	サブツリー名	取得できる
1.3.6.1.2.1.1	system	○
1.3.6.1.2.1.2	interfaces	○
1.3.6.1.2.1.3	at	○
1.3.6.1.2.1.4	jp	○
1.3.6.1.2.1.5	icmp	○
1.3.6.1.2.1.6	tcp	○
1.3.6.1.2.1.7	udp	○
1.3.6.1.2.1.8	egp	×
1.3.6.1.2.1.9	transmission	×
1.3.6.1.2.1.11	snmp	○

4.2.1　考察

最新情報表示機能は、MRTG のグラフに加えて、最新の値の表示を実現した。ブラウザのボタンを押すことで、最新の値を即座に得ることができる。これは、即座に情報がほしいときに役立つことができる。また、統計情報と現状把握ができる。

何かエラーが生じても MRTG では、すぐに情報を見ることができないかもしれないが、最新情報表示機能は回復作業後に即座に、状況が確かめることができると考えた。

今後は、マシンの管理者にとって、使いやすいもので、知りたい情報は何であるのか調べていく必要がある。そして、トラフィック以外の値、例えば mib-2 や ucdavis サブツリーなどの値も取得できるので、今後最新表示機能を利用することが可能である。また、表示が見にくいならば、機能をさらに拡張していくことが今後の課題となる。

表 4.2　ucdavis サブツリー (1.3.6.1.4.1.2021)

OID	サブツリー名	取得できる
1.3.6.1.4.1.2021.2	prTable	×
1.3.6.1.4.1.2021.4	memory	○
1.3.6.1.4.1.2021.8	extTable	×
1.3.6.1.4.1.2021.9	dskTable	×
1.3.6.1.4.1.2021.10	laTable	○
1.3.6.1.4.1.2021.11	systemStats	○
1.3.6.1.4.1.2021.14	ucdDemoMIB	×
1.3.6.1.4.1.2021.15	fileTable	×
1.3.6.1.4.1.2021.100	version	○
1.3.6.1.4.1.2021.101	snmperrs	○
1.3.6.1.4.1.2021.102	mrTable	○

第 5 章

まとめ

本研究では、警報通知機能と最新情報表示機能を実装した。

MRTG では、マシンのトラフィック量の異常判定ができなかった。提案した警報通知機能を用いると、閾値を超えるとメールが送られてくることを実地で確認した。また、最新情報表示機能は、MRTG のグラフに加えて、最新の値の表示を実現した。

今後警報通知機能は、閾値をどのように定めていくのかが課題である。これは、トラフィック量を長期に渡り収集し、適正な値を自動で得るなどの方法が有効であると考える。また、最新情報表示機能は、使いやすいものなのか、知りたい情報なのか調べ、機能をさらに拡張していくことが今後の課題となる。

謝辞

本研究において、多くの方々に御協力いただき、ありがとうございました。

藤岡さん、澤田さんには、席が近いということでよく研究やパソコン操作について教わりました。澤野くんには、いびられながらも困った時には、助けてくれました。小川さんには、良き話し相手になってもらいました。豊島くんは、優しく対応してもらい、本も貸してくれました。

広瀬さんには、研究のアドバイスをいただき、多くの情報を教えてもらいました。正岡さんには、研究に関する本を見つけてくれました。田渕、西内さん、舟橋さん研究指導や応援ありがとうございました。

研究の指導には、菊池豊助教授にアドバイスをいただきお世話になりました。

参考文献

[1] Steve Maxwell. SNMP ネットワーク管理ツール. 株式会社 翔泳社, October 2001.

[2] 片岡厳. パワーアップ FreeBSD. 株式会社技術評論社, April 2001.

[3] 冨成章彦. オープンソースを使ったネットワーク監視術. 有限会社セレンディップ, August 2001.

[4] 堀田倫英, 石井達夫, 広川類. PHP 徹底攻略. ソフトバンク パブリッシング株式会社, August 2000.

付録 A

MRTG の設定方法

MRTG での監視までには、次の 4 つの手順が必要である。

1. MRTG の導入
2. 監視対象の SNMP 設定
3. mrtg.cfg の作成
4. mrtg の実行

A.1 MRTG の導入

必要なソフトウェアは以下の 4 つである。

A.1.1 導入手順

1. zlib の導入
2. libpng 導入
3. gd 導入
4. mrtg 導入

A.1.2 監視対象の SNMP 設定

研究室内の kaeru (IP アドレスは 172.21.43.2) というルータのトラフィックを監視することとする。これは、SNMP が実装されていて、設定もされているので特に設定はしない。

自分のマシンだと設定を行う (付録 C)。

A.1.3　mrtg.cfg の作成

MRTG が実行されるときに読み込まれるファイルをが mrtg.cfg ある。MRTG には cfgmaker という Perl で記述されたプログラムが用意されており、監視対象機器の MIB(Management Information Base) 情報を取得し、出力ファイルの生成に使う mrtg.cfg を作成してくれる。ファイル名を kaeru.cfg とする。

A.1.4　cfgmaker の構文

cfgmaker コミュニティ名@監視対象機器のホスト名 (または IP アドレス) > ファイル名

kaeru.cfg の作成。

> cfgmaker public@172.21.43.2 > kaeru.cfg

kaeru.cfg の中身

```
##################################################################
# System: kaeru
# Description: Sun SNMP Agent, Ultra-60
# Contact: System administrator
# Location: System administrators office
##################################################################
### Interface 2 >> Descr: 'hme0' | Name: '' | Ip: '172.21.43.2' | Eth:
 '08-00-20-b2-4e-ac' ###
Target[172.21.43.2_hme0]: \hme0:public@172.21.43.2:
SetEnv[172.21.43.2_hme0]: MRTG_INT_IP="172.21.43.2" MRTG_INT_DESCR="hm
e0"
```

```
MaxBytes[172.21.43.2_hme0]: 12500000
Title[172.21.43.2_hme0]: Traffic Analysis for hme0
PageTop[172.21.43.2_hme0]: <H1>Traffic Analysis for hme0 </H1>
 <TABLE>
   <TR><TD>System:</TD><TD>kaeru in System administrators office</TD>
</TR>
   <TR><TD>Maintainer:</TD><TD>System administrator</TD></TR>
   <TR><TD>Description:</TD><TD>hme0 </TD></TR>
   <TR><TD>インタフェースタイプ:</TD><TD>ethernetCsmacd (6)</TD></TR>
   <!--<TR><TD>ifName:</TD><TD></TD></TR>-->
   <TR><TD>最大伝送速度:</TD><TD>12.5 MBytes/s</TD></TR>
   <TR><TD>Ip アドレス:</TD><TD>172.21.43.2 ()</TD></TR>
 </TABLE>
```

#ではじまる行はコメントと解釈され、空白行は無視される。

A.1.5 kaeru.cfg ファイルのカスタマイズ

kaeru.cfg に設定を付け加える。kaeru.cfg は、キーワードを使う。

WorkDir:定義を追加する。Web サーバで閲覧できるところを指定する。

Directory[ターゲット名]:を使って監視対象機器ごとにディレクトリを分けて出力させる。

自動的に生成されないためあらかじめ作成しておく。

ディレクトリの作成

> cd public_html

> mkdir MRTG/kaeru

kaeru.cfg に設定を付け加える。

WorkDir: /home/lab/kiku02/asuka/public_html/MRTG

Directory[172.21.43.1_hme0]:kaeru

A.1.6　mrtg の実行

実行は mrtg という Perl のプログラムに kaeru.cfg を渡している

> mrtg kaeru.cfg

3 回実行する。

A.1.7　cron の実行

WWW でグラフが出るのを確認する。あとは kaeru.cfg を 5 分ごとに実行してやればよい。これは Unit に標準的に備わっている cron を用いる。

crontab ファイルに書き込み

/5 * * * * /usr/local/bin/mrtg /home/lab/kiku02/asuka/MRTG/kaeru.cfg

crontab で定期的に実行

cron の実行

> crontab crontab

A.1.8　日本語を表示する

Title や PageTop などに日本語で文字を埋め込んでも MRTG が生成する.html ファイルでは日本語フォントが正しく表示できない。この.html ファイルは Perl で書かれた mrtg で生成されるので、このプログラム自体を直接書き換える。

mrtg ファイル

> su

xemacs /usr/local/bin/mrtg

mrtg ファイルの変更 (890 行目)

'<META HTTP-EQUIV-=”Content-Type” CONTENT=”text/html;'.&$LOC

('charset=x-jis-jp').” \”> \ n”;

A.2　SNMP の指定の仕方

MRTG の Target 指定について説明する。

- Keyword: Targer　データ収集項目を指定
- SNMP データの収集
- 外部コマンド結果の埋め込み収集

SNMP データの収集　Target[<target name>]:<target kind>:<community>@<address>

<target name>：測定機器の名称

<target kind>：測定項目

<community>：測定機器に設定している community string

<address>：測定機器のアドレス・ホスト名

A.2.1　SNMP データ収集指定方法

- Port 指定 (ifIndex 指定)
- SNMP OID 指定/SNMP MIB symbol 指定
- Interface Address 指定
- 組み合わせ指定

A.2 SNMP の指定の仕方

Port 指定 (ifIndex 指定)

SNMP Client 側で管理している Port 番号 (ifIndex) を使ってデータ照会する。

IfInOctets と ifOutOctets を測定例 1 Target[gw1-3]： 3：public@gw1.foo.co.jp

gw1.foo.co.j に収容されている ifIndex=3 の Interface に関して ifInOctets/ifOutOcters を測定

例 2　Target[gw1-3]： － 3：public@gw1.foo.co.jp

例 1 の In/Out を逆にしてデータ収集する

SNMP OID 指定/SNMP MIB symbol 指定

SNMP OID(Object ID) または MIB symbol を指定し、データ照会する。変数 1、変数 2 は”&” で連結指定する

例 3　Target[gw1-3]： ifInErrors.3&ifOutErrors.3：public@gw1.foo.co.jp

ifInErrors/ifOutErrors を測定 例 4 Target[gw1-3]：1.3.6.1.2.1.2.2.1.14.3&1.3.6.1.2.1.2.2.1.20.3public@gw1.foo.co.jp →例の OID 指定

Interface Address 指定

機器の構成変更の度に設定変更をさけるためにインターフェースに割り振られたアドレスをキーにしてデータ照会を行う。

デフォルトでは ifInOctets と ifOutOctets を測定

例 5 Target[gw1-3]：/172.21.43.1：public@gw1.foo.co.jp

例 6 Target[gw1-3]：－/172.21.43.1：public@gw1.foo.co.jp

例 5 の In/Out を逆にしてデータ収集する

組み合わせ指定

Interface addres 指定と OID/MIB symbol 指定を組み合わせる

例　7　Target[gw1-3]：ifInDiscards/172.21.43.1&ifOutDiscards/172.21.43.1：public@gw1.foo.co.jp

ifOutDiscards/ifOutDiscards を測定

例　8　Target[gw1-3]：1.3.6.1.2.1.2.2.1.13/172.21.43.1&1.3.6.1.2.1.2.2.1.19/172.21.43.1：

A.2　SNMP の指定の仕方

public@gw1.foo.co.jp　→例 7 の OID パターン

　コマンド埋め込み指定 Target[<target name>]：｀<command>｀

<targer name>：測定機器の名称

<command>：測定コマンド

Target キーワードにプログラムのパスを渡す。そのパスを”｀“：バックシングルコーテーションでくくる。

付録 B

MRTG のキーワード

mrtg.cfg に最低限記述しなければならないキーワードは 5 つある。

- WorkDir：
- Target[ ]：
- MaxBytes[ ]：
- Title：
- Page Top[ ]：

B.1　文法

MRTG の設定ファイルの書き方は、以下のようにいくつかの簡単なルールになっている。

- キーワードは行の先頭から始まらなければならない。
- キーワード行に続いて空白から始まる行は、そのキーワード行に追加される。
- 空白は無視される。
- #で始まる行はコメントである。

表 B.1　グローバルパラメータ

Workdir	ログファイルと Web ページが生成される場所を指定 例　WorkDir: /home/lab/kiku02/asuka/public_html/MRTG

B.1.1　ターゲットごとの設定

各ディレクティブの後に、[ターゲット名] という記述があるが、これはその設定がどのターゲットに所属しているかの識別子になるとともに、MRTG が生成するファイルの名前ベースなどになる (表 B.3)。

表 B.4　対象パラメータごとのオプション

表 B.2　オプショナルなグローバルパラメータ

IconDir	MRTG で共通に利用されるアイコンの URL を指定 例　IconDir: /img
Refresh	ブラウザに、何秒後にリロードすればいいかを指定する。デフォルトでは 300 秒。 例：Refresh: 600
Language	出力フォーマットを選択した言語に切り替える 例：Language: iso2022jp

AbsMax[ターゲット名]: 数値	取得する値が MaxBytes で設定した値を超える可能性があるあ場合、ここで実際の最大値を設定する。
Unscaled[ターゲット名]: ymwd	通常はグラフの縦軸は取得する値の最大値によって変化するが、ここで設定した対象のグラフについては、MaxBytes により縦軸が決定される。y=year,m=month,w=week,d=day
WithPeak[ターゲット名]: ymw	平均化された値の他にピークの値を y,m,w グラフで表示する。
Suppress[ターゲット名]: ymwd	MRTG は日・月・年の 4 つのグラフを生成するが、この設定で不要なグラフを作成しないようにする。
Extension[ターゲット名]: 拡張子	デフォルトでは MRTG は.html ファイルを生成する。この設定で他の拡張子にすることができる。

表 B.3　ターゲットごとの設定

Target[ターゲット名]: ターゲット	どこの情報を参照するのかを記入する。
MaxBytes[ターゲット名]: 数値	インターフェイスのトラフィックの最大量を設定する。取得した値がこれより大きくなった場合、MRTG はその値を無視する。
Title[ターゲット名]: タイトル文字列	グラフ表示用に生成される HTML ページのタイトル。
PageTop[ターゲット名]: 文字列	MRTG が生成する HTML のトップの文字列の設定。

Directory[ターゲット名]: ディレクトリ名	WorkDir のサブディレクトリに MRTG のファイルを生成するよう指示することができる。サブディレクトリは事前に作っておく必要がある。
Xsize[ターゲット名]: 数値 Ysize[ターゲット名]: 数値	グラフサイズをピクセルで指定。Xsize は 20 から 600 の間、Ysize は 20 より大きい必要がある。
YticsFactor[ターゲット名]: 倍数の数値	取得した値のグラフ上の縦軸の値の倍数を指定。
Factor[ターゲット名]: 倍数の数値	取得した値に対してグラフの下に表示されるサマリの値の倍率を指定する。
Options[ターゲット名]: オプション、オプション,…	growright：グラフの右端が現在時刻になる。 bits：バイトで取得した値をビット (つまり 8 倍) にして表示する。 nopercent：利用率のパーセント表示をしない。 Gauge：取得した値そのものを表示する。
Kilo[ターゲット名]: 数値	MRTG の初期値では、キロ・メガを計算する際に 1000 を用いる。この値をたとえば 1024 に変更する場合に指定する。
Background[ターゲット名]: 色	Background で生成される HTML ページの背景色を指定することができる。
Ylegend[ターゲット名]: グラフ縦軸の説明の文字列	グラフ内に表示されるグラフ縦軸の説明の文字列を指定する。
ShortLegend[ターゲット名]: 単位の文字列	グラフの下に表示される数値の単位の文字列を設定する。初期値は b/s。
LegendI[ターゲット名]: グラフの説明文字列	グラフの下のサマリ行に表示される値の説明の文字列を指定することができる。LegendI・O がそれぞれあるが、1 番目に取得される値 (通常 IN のトラフィック)・2 番目に取得される値 (通常 OUT のトラフィック) に対応している。

Legend1[ターゲット名]: グラフの説明文字列	最後のグラフのさらに下に各グラフの線の説明をする文字列が入るが、その説明文字列を設定する。Legend1・2・3・4 がそれぞれあるが、1 番目に取得される値 (通常は IN のトラフィック) のグラフ、2 番目に取得される値 (通常は OUT のトラフィック) のグラフ、WithPeak が設定されている場合に 1 番目に取得される値の最大のグラフ・WithPeak が設定されている場合に 2 番目に取得される値の最大値のグラフに対応している。

表 B.5　スレッシュホールドのチェック

ThreshDir[ターゲット名]: ディレクトリ名	ThreshDir を書き込み可能なディレクトリに設定することで、MRTG はスレッシュホールドを超えた時のみ警告を発す。
ThreshMinI[ターゲット名]: 下限値 ThreshMinO[ターゲット名]: 下限値 ThreshMaxI[ターゲット名]: 上限値 ThreshMaxO[ターゲット名]: 上限値	閾値を設定する。値に%をつけると MaxBaytes に対する割合になる。Max・Min はそれぞれ、上限と下限、I・O はそれぞれ、1 番目に取得される値 (通常 IN のトラフィック)・2 番目に取得される値 (通常 OUT のトラフィック) に対応している。
ThreshDesc[ターゲット名]: 値	この値は、以下で設定されるプログラムが呼び出される前に環境変数 THRESH_DESC として設定される。プログラムはユーザフレンドリな出力をするためにこの値を使用することができる。
ThreshProgI[ターゲット名]: 実行ファイル ThreshProgO[ターゲット名]: 実行ファイル	設定した Max の閾値を超えた、または、Min の閾値を下回った場合に実行するプログラムを指定する。

付録 C

ucd-snmp の設定方法

ucd-snmp の設定の手順を述べる。

まず、http://not-snmp.sourceforge.net から ucd-snmp-4.2.2.tar.gz を取得し、インストールする。

>cd /usr/local/src

>tar xzvf ucd-snmp-4.2.2.tar.gz

>cd ucd-snmp-4.2.2

>./configure –with-libwrap=/usr/local/lib

(管理者のアドレス、システムの場所 (名前)、ログファイルのパスを聞かれる。)

-press return to continu (Enter)

system contact information:メールアドレス asuka@kikuken.org

system location:ThinkPad Server

location to write logfile (/usr/log/snmpd.log):(Enter)

location to write persistent information (/var/ucd-snmp):(Enter)

>make

>su

#make install

インストールされたのは以下になっている。

/usr/local/sbin/snmpd

/usr/local/bin/snmpget

/usr/local/bin/smpwalk

/usr/local/share/snmp/mibs

snmp.conf は、ソースディレクトリに EXAMPLE.conf としてあるのでこれを/usr/local/share/snmp/に snmp.conf としてコピーする。所有権を snmpd を起動しユーザにして、属性も変える。

#cp EXAMPLE.conf /usr/local/share/snmp/snmp.conf

#chmod 600 /usr/local/share/snmp/snmp.conf

環境設定

設定ファイル/usr/local/share/snmp/snmp.conf を編集

#vi snmp.conf

sec.name source community

com2sec local localhost private

com2sec mynetwork 172.21.43.0/24 kikuken

セキュリティグループの設定

group sec.model sec.name

#group MyPwGroup v1 local

#group MyPwGroup v2c local

#group MyPwGroup usm local

group MyROGroup v1 mynetwork

group MyPOGroup v2c mynetwork

group MyPOGroup usm mynetwork

ビューの設定

incl/excl subtree mask

view all included .1 80

view system included system fe

view mib2 included .iso.org.dod.internet.mgmt..mib-2 fc

アクセス権限の設定

context sec.model sec.level match read write notif

access MyROGroup "" any noauth exact all none none

access MyRWGroup "" any noauth exact all all none

システム情報の設定

syslocation ThinkPad Server

syscontact Asuka Maeda<asuka@kikuken.org>

snmpd の起動

>su #cd /usr/local/sbin

#snmpd

#exit

実行確認

>su

>ps -ax | grep snmpd

正常情報取得

デーモンがきちんと反応してくれるかをテストする。

>snmpwalk -v 1 localhost private system

:

稼動環境の確認

#cd /etc

#vi services

snmp 161/tcp

snmptrap 162/tcp snmp_trap

162/udp

付録 D

Apache の設定方法

Apache の設定の手順を述べる。

まず、http://httpd.apache.org から apache_1.3.22.tar.gz を取得し、インストールする。

>mkdir /usr/local/src

>cd /usr/local/src

>tar xzvf apache_1.3.22.tar.gz

>cd apache_1.3.22

>env OPTIM=-O2 ./configure –enable-module=so

(パフォーマンス向上) (DSO 使用可能) >make

>su

#make install

設定

設定は、httpd.conf で行う。

#cd /usr/local/apache/conf

#xemacs httpd.conf

ポート番号とユーザ名

port 80 ←ポート番号 80

User nobody ←ユーザ名・ユーザ権限

Group nobody ←グループ名・グループ権限

サーバ名と管理者名

serverAdmin asuka@kikuken.org ←管理者のメールアドレス

#serverName www.kikuken.org　　←サーバ名

使用するディレクトリの設定

DocumentRoot ”/usr/local/apache/htdocs” → http://localhost/でアクセスするディレクトリ

Alias /icons ”/usr/local/apache/icons/” → http://localhost/icons/でアクセスするディレクトリ

Script Alias /cgi-bin/ ”/usr/local/apache/cgi-bin/” → http://localhost/cgi-bin/でアクセスするディレクトリ

UserDir public_html → http://localhost/ asuka/で、ユーザ asuka のディレクトリにある public_html ディレクトリ

ファイルやディレクトリの設定

<Direcory ”/usr/local/apache/htdocs”> ← DocumentRoot 用の設定

Options Indexes FollowSymLinks ExecCGI

AllowOverride None

Drder allow,deny

Allow from all ←すべてのクライアントからのアクセスを許可

</Directory>

<Directory /home/*/public_html> ←各ユーザ用の設定

AllowOverride Authconfig FileInfo Indexes Limit

Option ExecCGI ← CGI を実行可能にする

Option Indexes

Deny from all

Allow from 172.21.43.0/24

Allow from 127.0.0.1 ← localhost と 172.21.43.0/24 以外からの接続を拒否

</Directory>

Directory index.html index.htm index.shtml index.cgi index.php ←ディレクトリ名だ

けでアクセスしたときに表示するファイル

ErrorDocument 401/401.html ← 401(Unauthorized) のときのメッセージ

ErrorDocument 403/403.html ← 403(Forbidden) のときのメッセージ

ErrorDocument 404/404.html ← 404(NotFound) のときのメッセージ

CGI の設定

AddHandler cgi-script .cgi

使用文字コード

<Directory /home/*/public_html>

：

AddCharset EUC-JP .html ←この箇所に追加

</Directory>

起動と終了

#/usr/local/apache/bin/apachect/ start ←起動

#/usr/local/apache/bin/apachect/ stop ←終了

#/usr/local/apache/bin/apachect/ restart ←再起動

起動の確認

http://172.21.43.250 で確認する。

Apache によるメッセージが表示されればよい。

付録 E

PHP の設定方法

PHP の設定の手順を述べる。

まず、http://www.php.gr.jp/project/i18n/から php-3.0.18-j18n-ja.2.tar.gz を取得し、インストールする。また、http://www.php.net/downloads.php から php-4.0.6.tar.gz も取得し、インストールを行う。

php3 のインストール

>cd /usr/local/src

>tar xvzf php-3.0.18-i18n-ja.2.tar.gz

>cd php-3.0.18

>./configure –with-pqsql –with-zlib

–enable-track-vars

–with-apxs=/usr/local/apache/bin/apxs

–with-snmp –enable-ucd-snmp-hack

–enable-i18n –enable-mbregex

>make

>su

#make install

configure の指定は、-with-apxs=/usr/local/apache/bin/apxs は Apache の DSO モジュール版としての構築を意味する。–with-snmp,–enable-ucd-snmp-hack は SNMP にアクセスする関数を利用するための指定である。

日本語用の設定

#cd /usr/local/src/php-3.0.18-i18n-ja-2/php3.ini-dist php3.ini /usr/local/lib/php3.ini

php4 のインストール

>cd /usr/local/src

>cd php-4.0.6

>./configure –with-pgsql –wit-zlib

–with-apxs=/usr/local/apache/bin/apxs

–with-snmp

–enable-mbstring –enable-mbstr-enc-trans

–enable-versionning

>make

>su

#make install

–enable-versioing は、PHP3 と PHP4 の共存を実現するために指定した。

Apache の設定 (修正・追加)

PHP に関する設定が追加されている

#cd /usr/local/apache/conf

#vi httpd.conf

PHP モジュールを組み込む設定

LoadFile /usr/local/lib/libpg.so

LoadModule php3-module /usr/local/apache/libexec/libphp3.so

LoadModule php4-module /usr/local/apache/libexec/libphp4.so

AddModule mod-php3.c

AddModule mod-php4.c

ファイルタイプの設定

#を削除し、設定を有効にする

AddType application/x-httpd-php3 .php3

AddType application/x-httpd-php3 .html

AddType application/x-httpd-php3-source .phps

AddType application/x-httpd-php .php

AddType application/x-httpd-php .html

AddType application/x-httpd-php-source .phps

Directory Index index.html index.htm index.shtml index.cgi index.php index.php3

設定のテスト

#/usr/local/apache/bin/apachect/ configtest ←テスト

#/usr/local/apache/bin/apachect/ restart ←再起動

PHP の作成

>cd public_html/jikken/PHP/php3/test.php3

<?php

echo phpinfo();

?>

PHP の情報の確認

www で http://172.21.43.250/ asuka/jikken/PHP/php3/test.php3 で情報を得られることを確認する。

php4 は、インストールしたが日本語が利用できなかった。よって、日本語を使う時は、php3 で設定した。