

< 特別企画 >

NT/2000/XP/2003  
対応の完全無償ツール



本稿で紹介するフリーソフトウェアが本誌付録CD-ROMの「¥Tokubetsu¥MRTG」フォルダに収録されています。

トラフィックのグラフ表示で障害を事前にキャッチ！

# MRTGで

## Windowsネットワーク監視

MRTG (Multi Router Traffic Grapher) は、ネットワークの負荷を監視する無償ツールである。トラフィックの状態を継続的にグラフ化して、Webブラウザに表示できるので、障害予測や実効スループットの把握も直感的に行える。もともとはUNIX互換のOS向けに開発され、多くのUNIX管理者に活用されてきたが、UNIX版はユーザー自身でコンパイルする必要がある。一方、Windows版はすでにコンパイルされているので、Windowsネットワークへの導入も簡単だ。MRTGのすぐれた機能を試してもらいたい。

稗田恭磨

### CONTENTS

- Part1 情報収集の仕組みとインストール
- Part2 使いこなしの事前設定ポイント
- Part3 グラフ分析の実践ケーススタディ

Multi Router Traffic Grapher



## Part1 | グラフ表示でネットワークの問題を迅速に発見!

# 情報収集の仕組みとインストール

ネットワークのトラフィックを常時測定できるMRTGは、すぐれた機能を提供するため、そのインストール方法が少しだけ複雑になっている。まずは、ダウンロードとインストールを成功させよう。

## MRTGはどんな場面で役に立つのか?

Multi Router Traffic Grapher

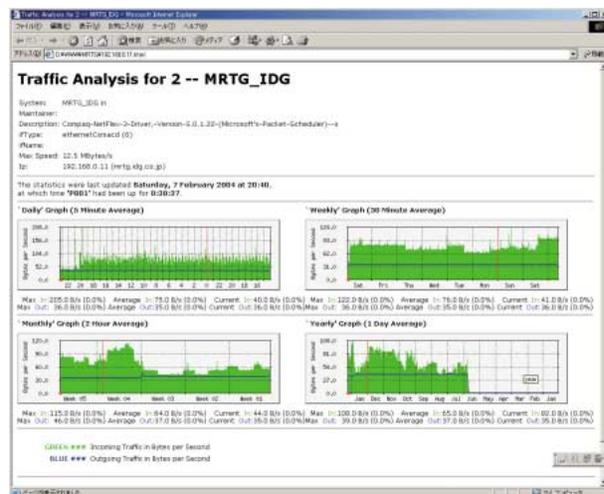
### Windows標準ツールより便利

MRTGはネットワークのトラフィックを監視する無償ツールである。ネットワークトラフィックの状況をグラフ化して、HTMLファイルを作成するので、Webブラウザを使ってネットワークの状況を直感的に把握できる。また、OSやネットワーク機器からの情報収集は、ネットワーク経由でオブジェクトを監視・制御できるプロトコルのSNMP(Simple Network Management Protocol)を利用するので、SNMPエージェントからの情報もグラフ表示できる。ネットワーク管理者は、グラフで表示されたトラフィックの推移から、現在および将来のネットワーク帯域を監視もしくは予測でき、帯域が飽和する前に対策を取れるようになる。また、トラフィックだけではなく、ネットワーク上の機器のCPU使用率な

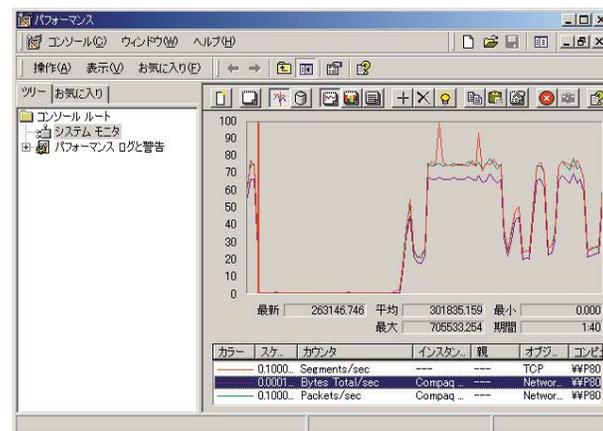
ど、さまざまな情報を監視することで、管理者は障害を予測し、すばやい対応ができるようになるのだ(画面1)。

実は、Windowsでも「パフォーマンスモニタ」を使えば、ネットワークインタフェースのトラフィックを監視して、その状況をグラフ表示でき、ファイルとしても保存できる。また、WindowsXPからは、ネットワークトラフィックを同じくグラフ表示できる「タスクマネージャ」が用意されている。しかし、これらの機能で収集できる情報の対象は、WindowsというOSでのトラフィックに限られ、監視期間も1つにすぎない(画面2)。

一方、MRTGでは、「日・週・月・年」という4種類の監視期間を設定して、トラフィック状況を収集する。また、これらの収集された情報は1ページのWebブラウザに1度に表示されるので、比較・検討が簡単に行え、ネットワークの問題点を発見しやすい。さらに、無償で利用できるため、世界中の多くの管理者によって活用されている。



画面1 Webブラウザを使ってグラフ表示が可能なMRTG。そのトラフィックの推移からネットワーク帯域を監視して、障害予測ができる



画面2 Windows付属の管理ツール「パフォーマンス」では、表示できる情報が1期間に限られているので障害予測に利用しづらい

## 多項目の情報を収集

MRTGは、SNMPを利用してネットワーク機器の情報を収集する。そのため、監視対象もWindowsだけではなく、SNMPを利用しているルータやスイッチなどの機器(SNMPエージェント)からも情報を収集して、グラフ表示できる。MRTGでは、SNMPの「ifInOctets」と「ifOutOctets」というMIB(Management Information Base)に対して、ポーリング(要求確認)を行う。MIBは、SNMPによって管理されるオブジェクトデータベースで、一般的なネットワーク機器のMIBには、「CPU使用率」「不正パケット数」などが用意されている。MIBに用意されていなくても、SNMPに対応している監視対象のネットワーク機器にMIBを追加設定すれば、収集する情報を増やせる。例えば、「メモリ容量」「ハードディスクの空き容量」「スパムメール受信数」「温度」などの計測にも対応できる(図1)。

## リモートから監視できる

MRTGが収集したトラフィック情報は、HTMLファイルとして作成され、Webブラウザで表示できる。Webサーバと組み合わせると、リモートからの監視にも使える。また、MRTGはもともとUNIX互換のOS向けに作られているので、作成されるHTMLファイルは汎用性が高い。監視する側のPC端末では、Internet Explorer(IE)やNetscapeなどのWebブラウザがあれば、収集した情報にすぐにアクセスできる。

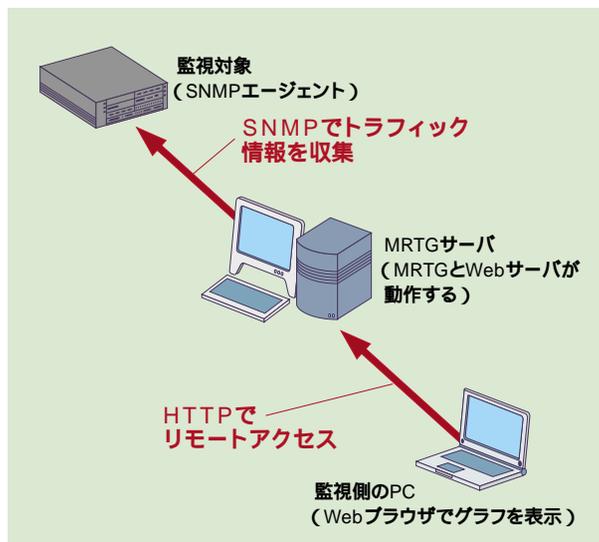


図1 MRTGはSNMPで監視対象にアクセスして、トラフィック情報を収集する

## “負荷”の判断材料に活用

このようにMRTGは、Windowsやネットワーク機器を通過するトラフィックの推移を示す。ネットワーク上のさまざまな情報を監視できるので、管理者が障害を事前に予測して対応すればよい。特にMRTGが役に立つ場面は、ある期間内に「何がピーク(最大の負荷)に達しているのか?」を判断するときである。例えば、Webサイトの閲覧が重くなったと感じたら、システムの拡張が必要なのかどうかを見極めなければならない。その際の判断材料を提供してくれる。帯域が飽和している場合は回線を増速するとか、サーバのメモリが飽和している場合は、メモリを増設するなどの対策が取れるだろう。逆に、システムがそれほど利用されていないことが判明すれば、回線やメモリを減らすなどして、節約できるかもしれない。

ただし、MRTGは大手ベンダーから提供されているようなネットワーク管理システムではないので、問題の発見と解決を行わない点には注意が必要だ。MRTGには警報を発したり、トラップを作成する機能はない。あくまでネットワーク上の値をグラフ表示するだけで、その問題の判断は管理者が行う。

## インストールの流れ

MRTGは「MRTG公式日本語サイト」から無料で入手できる。MRTGはUNIX互換のOS向けに開発されたが、Windowsでも動作するものが提供され、必要な言語ライブラリ(Perl)も無料で入手できる。

そのほか、Windowsで必要な設定としては、Windowsそのものを監視対象にする場合、OS標準の「SNMPエージェント」を利用する。また、WindowsをMRTGサーバとして構成し、リモートからアクセスしたいなら、IIS(インターネットインフォメーションサービス)をWebサーバとして利用する。OS標準の「タスク」で設定すれば、MRTGを自動スケジュールで運用できる。

まずは、MRTGを利用する前に、次の5ステップを踏む必要がある。詳細については、次ページ以降を参照していただきたい。なお、MRTGの「設定」については、Part2で紹介する。

- Step1 MRTGのダウンロード
- Step2 Perlのダウンロード
- Step3 SNMPエージェントのインストール
- Step4 Perlのインストール
- Step5 MRTGのインストール

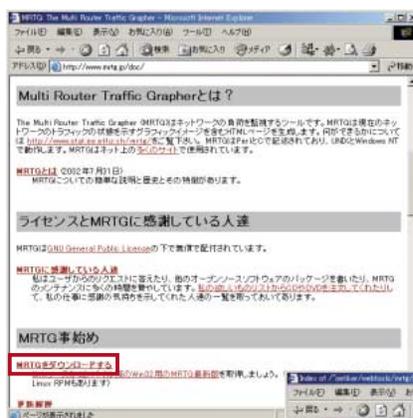
# Part1 情報収集の仕組みとインストール

## MRTGの実行環境を構築する5ステップ

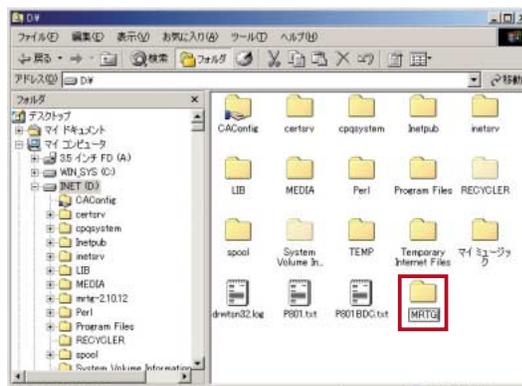
Multi Router Traffic Grapher

### Step1

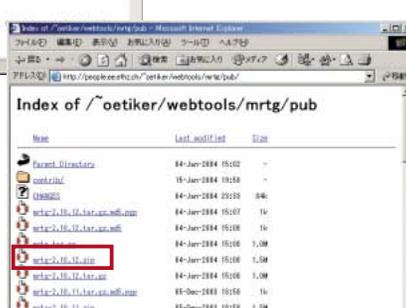
## MRTGのダウンロード



1 「MRTG公式日本語サイト」(<http://www.mrtg.jp/doc/>)にアクセスして、「MRTGをダウンロードする」をクリックする



2 最新版のアーカイブをクリックして、ダウンロードを開始する。さまざまなOS向けのMRTGが格納されているが、「.zip」の拡張子を持つファイルがWindows用である



本誌付録CD-ROMの「¥Tokubetsu¥MRTG」フォルダには、「mrtg-2.10.13.zip」を収録しているが、MRTGは頻繁にバージョンアップが行われているので、「MRTG公式日本語サイト」でバージョンアップの有無を確認して、最新版があれば、そちらをダウンロードして使ったほうがよい。

### 3

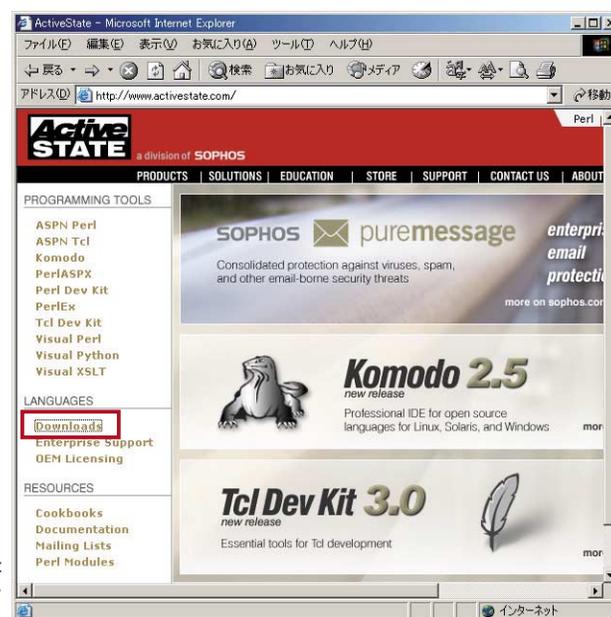
ダウンロード完了後、解凍ツールで展開すると、ダウンロード時のファイル名のフォルダが現れる。これができるかぎり「ルートディレクトリ」に移動して、ファイル名を「MRTG」に変更する(今回の例では「D:¥MRTG」)。この際、あとでMRTGを動作させるのに苦労するので、デスクトップ上への移動は避ける

### Step2

## Perlのダウンロード

通常のフリーソフトと異なり、MRTGの動作には「Perl」という言語ライブラリを必要とする。Perlは、「C言語」に似た「インタプリタ型」の言語で、コンパイルを行わなくても実行でき、WebサイトのCGI開発などによく使われる。また、「Windows Scripting Host(WSH)」でも利用可能だ。ここでは、Windowsプラットフォームで動作するPerlとして有名な、米国Active Stateの「Active Perl」を利用する。本誌付録CD-ROMの「¥Tokubetsu¥MRTG」には、「ActivePerl 5.8.2 build 808」を収録しているが、最新版の有無は米国Active StateのWebサイトで確認できる。なお、既存の環境でPerlを利用している場合、新たなインストールの必要はない。

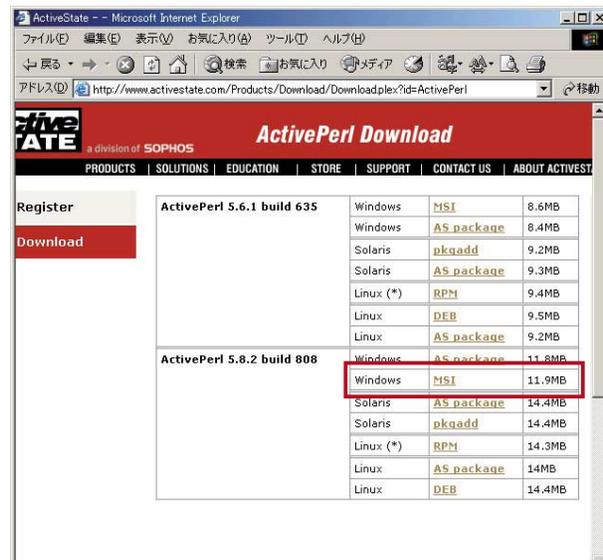
1 米国Active StateのWebサイト(<http://www.activestate.com/>)にアクセスして、「Downloads」をクリックする



## < 特別企画 > MRTG で Windows ネットワーク監視



- 2 「ActivePerl」右隣の「Download now」をクリックする。「E-Mail Address」や「Name」などの入力画面に移動するが、必須ではないので、そのまま「NEXT」をクリックする。



- 3 ダウンロードサイトが表示される。Windows 2000 の以降では、「Windows Installer」が標準装備されているので、「MSI」ファイルの最新版をクリックして、ダウンロードを開始する。保存先パスを「D:\setup¥ActivePerl」などにすると管理しやすい。

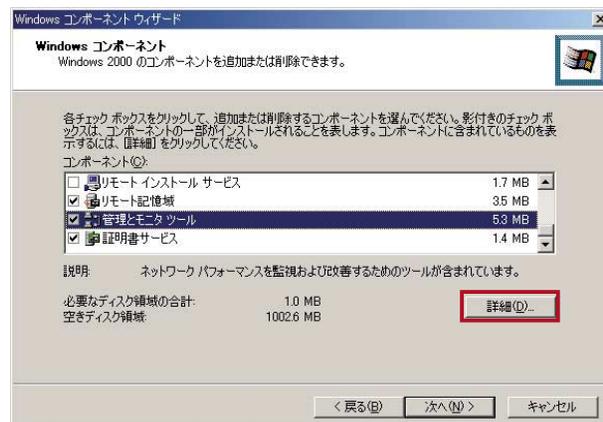
### Step 3

## SNMP エージェントのインストール

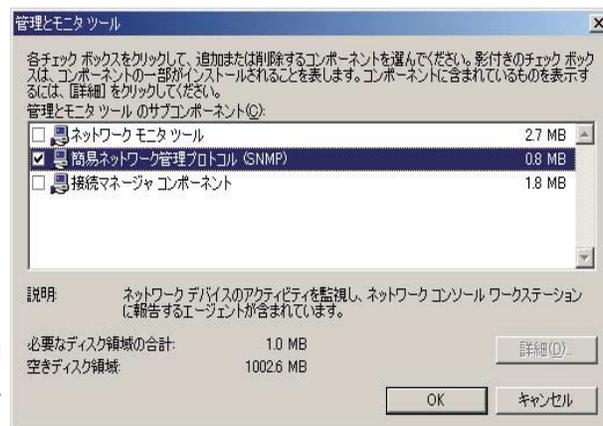
Windows には、標準で SNMP エージェントが添付されているのでこれを利用する。SNMP は、ネットワーク機器をネットワーク経由で監視・制御するプロトコルである。既存の環境で、SNMP を利用している場合、新たなインストールの必要はない。



- 1 「コントロールパネル」の「アプリケーション(プログラム)の追加と削除」を開いて、「Windows コンポーネントの追加と削除」をクリックする



- 2 「Windows コンポーネントウィザード」が表示されるので、「コンポーネント」の一覧から「管理とモニタツール」を選択して、「詳細」をクリックする



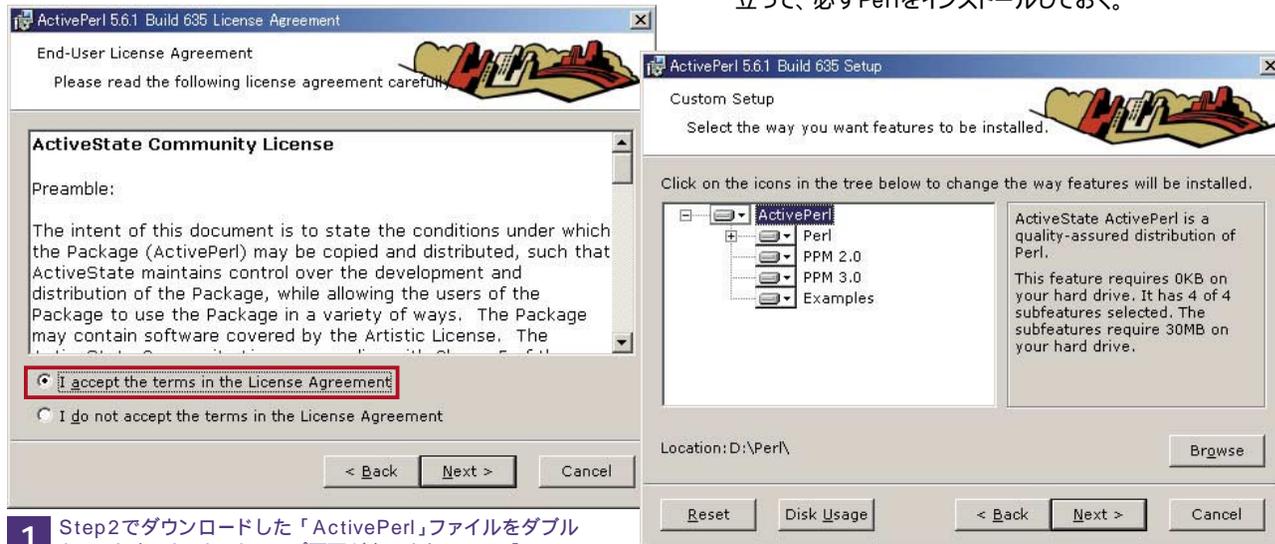
- 3 「管理とモニタツール」の「簡易ネットワーク管理プロトコル(SNMP)」にチェックを入れて、「OK」をクリックすると、ファイルのコピーが始まる。途中で、インストール CD-ROM の挿入を求められる場合がある

# Part 1 情報収集の仕組みとインストール

## Step 4

### Perlのインストール

MRTGは、その本体とインストーラ部分がPerlで書かれたプログラムになっている。そのため、インストールにもPerlの実行環境が必要なので、MRTGのインストールに先立って、必ずPerlをインストールしておく。



- Step 2でダウンロードした「ActivePerl」ファイルをダブルクリックすると、セットアップ画面が表示されるので、「Next」をクリックする。次の「License Agreement」の確認画面では、「I Accept...」を選択し、「Next」をクリックする

- インストール先を確認して、変更の必要があれば行う。通常は、このままでよい(例では「D:\Perl」)。次の「New features in PPM」の確認画面では、何もせず「Next」をクリックする



- オプション設定画面が表示される。インストールするモジュール(4つすべて)にチェックを入れて、「Next」をクリックする。なお、事前にIISがインストールされていない場合、下位2つのモジュールは利用できないが、IISで「Perl Script」を利用しなければ必要ない。続く画面で「Install」をクリックする



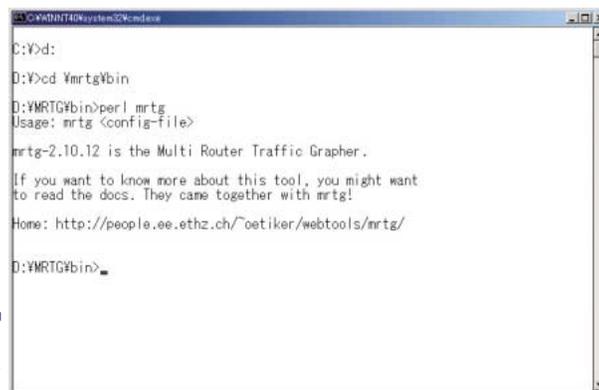
- インストールが終了したら、スタートメニューから「ファイル名を指定して実行」を立ち上げ、「CMD」と入力して「コマンドプロンプト」を開く。「echo %path%」と入力して、「コマンド検索パス」に手順で確認した「ActivePerlのインストール先フォルダ」が追加されていることを確認する。インストールが終わると、すぐにWSHやHTMLでPerlが利用できる

## Step 5

### MRTGのインストール

前述のとおり、MRTGのインストーラはPerlで書かれたスクリプトなので、インストールには「Perl」コマンドを利用する。実行ファイルをクリックしてもインストールはできない。

- 「コマンドプロンプト」を開く。「cd」コマンドを使って、Step 1で作成した「MRTG」の「bin」フォルダに移動する。その後「perl mrtg」と入力する。この画面のようなメッセージが表示されたらインストールは完了である



## Part2 自動運転で継続的に情報収集できる！

# 使いこなしの事前設定ポイント

MRTGのインストールが完了したら、「監視対象」「自動運用」「リモートアクセス」「ネットワーク機器のSNMP」に関する4つの設定を行い、MRTGを効率よく活用できるようにする。

### 設定ポイント1

## 監視対象を設定して、HTMLファイルを出力する

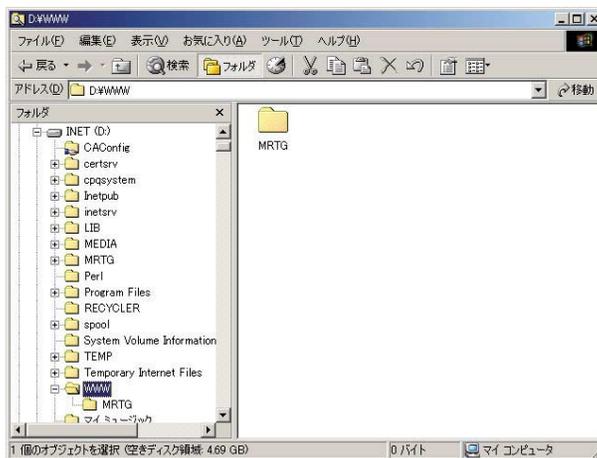
### Multi Router Traffic Grapher

MRTGはインストールが終わっても、すぐには利用できない。まずは実行前に、トラフィック監視の対象にしたいWindowsやネットワーク機器の情報を入力したり、収集したトラフィックの出力先などを決めたりする作業が必要になる。そのため、監視対象の「SNMPコミュニティ名」と「IPアドレス」を調べることから始めよう。Windowsや多くのネットワーク機器では、SNMPコミュニティ名に「public」が標準設定されている。今回は、このデフォルト値である「public」を利用して、MRTGがインストールされ

ているPC (Windows)をネットワーク監視対象として設定を行い、実際にHTMLファイルを出力してみる。

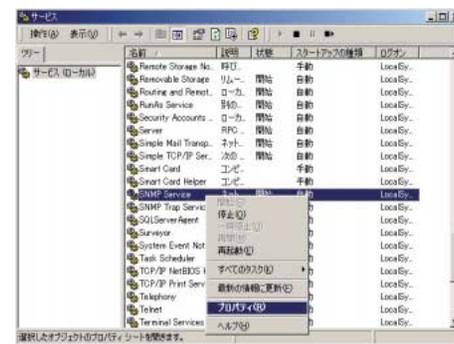
この作業の途中では、MRTGが利用するデフォルトの「設定ファイル」を作成する。「設定ファイル」には、対象機器の「ホスト名」「IPアドレス」や、監視対象はトラフィック負荷なのか、CPU負荷なのかといった監視対象オブジェクトの種類が設定される。MRTGは、この設定ファイルに書かれた内容に従って、HTMLファイルを作っている。

### 1 出力先フォルダを作成する



HTMLファイルの出力先フォルダを作成する。今回の例では、「D:\WWW\MRTG」に作成する

### 2 SNMPコミュニティ名を調べる



「コントロールパネル」の「管理ツール」から「サービス」を起動する。「SNMP Service」を右クリックして、「プロパティ」を開く

「セキュリティ」タブを選択すると、「受け付けるコミュニティ名」として、デフォルトの「public」が設定されていることを確認できる



# Part2 使いこなしの事前設定ポイント

## 3 IPアドレスを調べる

「コマンドプロンプト」を開く。「ipconfig」と入力して、自分自身のIPアドレスを確認する。ここでは「192.168.0.11」と設定されている

```
C:\WINNT40\system32\cmd.exe
D:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter ローカル エリア接続:

    Connection-specific DNS Suffix  . : mrtg.idg.co.jp
    IP Address. . . . .                : 192.168.0.11
    Subnet Mask . . . . .              : 255.255.255.0
    Default Gateway . . . . .          : 192.168.0.250

D:\>
```

## 4 設定ファイルを作成する

```
C:\WINNT40\system32\cmd.exe
D:\>cd %mrtg%bin

D:\MRTG\bin>perl cfgmaker public@192.168.0.11 --global "WorkDir: D:\www\mrtg" --
output mrtg.cfg
--base: Get Device Info on public@192.168.0.11:
--base: Vendor Id:
--base: Populating confcache
--snpo: confcache public@192.168.0.11: Descr 127.0.0.1 ネットワークの内部ループバ
ック インターフェイス --> 1
--snpo: confcache public@192.168.0.11: Descr Compaq NetFlex-3 Driver, Version 5.
0.1.22 (Microsoft's Packet Scheduler) s --> 2
--snpo: confcache public@192.168.0.11: Type 24 --> 1
--snpo: confcache public@192.168.0.11: Type 6 --> 2
--snpo: confcache public@192.168.0.11: Ip 127.0.0.1 --> 1
--snpo: confcache public@192.168.0.11: Ip 192.168.0.11 --> 2
--snpo: confcache public@192.168.0.11: Eth --> 1
--snpo: confcache public@192.168.0.11: Eth 00-50-9b-71-4b-72 --> 2
--base: Get Interface Info
--base: Walking ifIndex
--base: Walking ifType
--base: Walking ifAdminStatus
--base: Walking ifOperStatus
--base: Walking ifSpeed
--base: Writing mrtg.cfg

D:\MRTG\bin>
```

「コマンドプロンプト」を開く。コマンドプロンプト上でカレントフォルダを「D:\MRTG\bin」に移動して、次のコマンドを入力する。

```
perl cfgmaker public@192.168.1.11
--global "WorkDir: D:\www\mrtg" --
output mrtg.cfg
```

このIPアドレスは、上記「3」で調べたものを入力する。引数「WorkDir」に続くパスは、上記「1」で設定した「MRTGが作成するHTMLファイルの出力先」であるフォルダを入力する。その結果、物理インタフェースだけでなく、自分自身のIPアドレスを含む「Loop-back」インタフェースも含めたネットワークインタフェースを検索する様子が表示される。なお、「WorkDir」は、設定ファイルを作成するプログラムファイル「cfgmaker」の引数である。

また、「mrtg.cfg」の名前を設定対象ごとに変更すると、いくつもの「\*.cfg」ファイルが作成される。次は、「pc001」を対象に設定ファイルを作るコマンド例である。

```
perl cfgmaker public@<IP address>
--global "WorkDir: D:\www\mrtg"
--output pc001.cfg
```

「D:\MRTG\bin」に「mrtg.cfg」という設定ファイルができていることをエクスプローラで確認しておく

```
mrtg.cfg - 大規模
ファイル(E) 編集(E) 書式(O) ヘルプ(H)
### Interface 2 >> Descr: "Compaq-NetFlex-3-Driver,-Version-5.0.1.22-(Microsoft
Target[192.168.0.11_2]: 2:public@192.168.0.11:
SetEnv[192.168.0.11_2]: MRTG_INT_IP="192.168.0.11" MRTG_INT_DESCR="Compaq-NetFle
MaxBytes[192.168.0.11_2]: 12500000
Title[192.168.0.11_2]: Traffic Analysis for 2 -- mrtg
PageTop[192.168.0.11_2]: <H1>Traffic Analysis for 2 -- mrtg</H1>
<TABLE>
<TR><TD>System:</TD> <TD>MRTG_IDG in </TD></TR>
<TR><TD>Maintainer:</TD> <TD></TD></TR>
<TR><TD>Description:</TD><TD>Compaq-NetFlex-3-Driver,-Version-5.0.1.22-(Micro
<TR><TD>ifType:</TD> <TD>ethernetCsmacd (6)</TD></TR>
<TR><TD>ifName:</TD> <TD></TD></TR>
<TR><TD>Max Speed:</TD> <TD>12.5 MBytes/s</TD></TR>
<TR><TD>Ip:</TD> <TD>192.168.0.11 (mrtg.idg.co.jp)</TD></TR>
</TABLE>

WorkDir: D:\www\mrtg
```

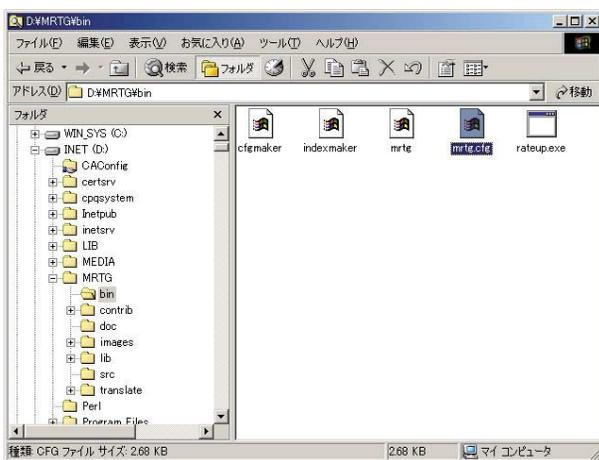
HTMLタグを理解できれば、テキストエディタで「mrtg.cfg」ファイルを開いてみよう。この設定ファイルが出力されるHTMLファイルのひな型になっている。表示テキストの内容を変更することも可能だ

## 5 HTMLファイルを出力する

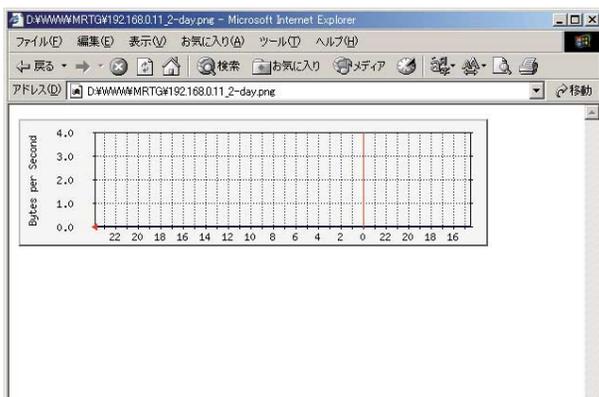


「D:\MRTG\bin」をカレントフォルダとしているコマンドプロンプト上で、次のコマンドを入力すると、MRTGが実行される。

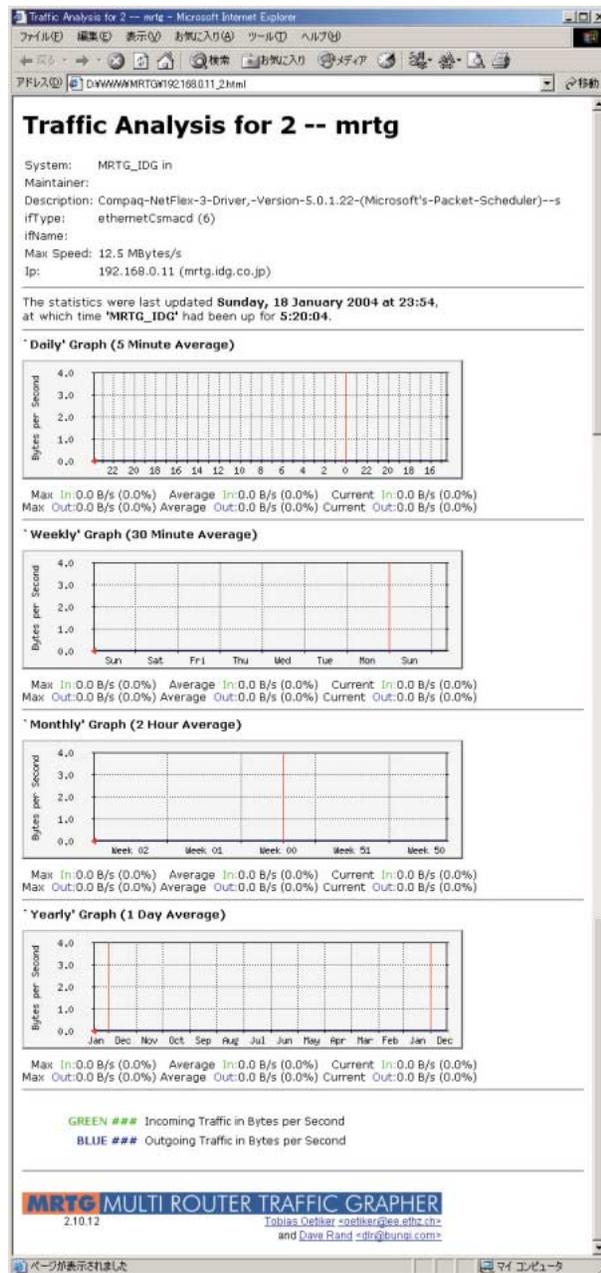
```
perl mrtg mrtg.cfg
```



「D:\WWW\MRTG」内に「\*.html」ファイルや「\*.png」ファイルが作成されている。「\*.png」ファイルは、「Portable Network Graphics」というフォーマットの画像で、MRTGではグラフとして表示される。Webブラウザへのドラッグ&ドロップや、マイクロソフト Officeの「Photo Editor」で確認できる



ファイル名にIPアドレスが付与された「.png」ファイルを開くと、Webブラウザにグラフが表示される



同じくファイル名にIPアドレスが付与された「.html」ファイルを開くと、MRTGが収集した情報を確認できる。監視対象のネットワークトラフィックを表示しているが、この時点では表示可能なトラフィックはない

# Part2 使いこなしの事前設定ポイント

## 設定ポイント2

# タスクスケジューラによる自動化で"常時"監視

## Multi Router Traffic Grapher

設定ポイント1の作業によって、HTMLファイルを作成し、グラフ表示が可能になった。ところがこのままの状態では、「perl mrtg mrtg.cfg」コマンドでMRTGが起動した時点の情報しか収集されない。設定ポイント2では、このコマンドを定期的に起動させて、監視情報を積み上げていく方法を紹介する。Win

dowsでは、「タスクスケジューラ」を使って、とても簡単にアプリケーションを自動スケジュール化できる。この自動化設定を完了すれば、将来にわたって常時監視を続けられる。なお、今回の例では「5分間隔」の監視を設定する。

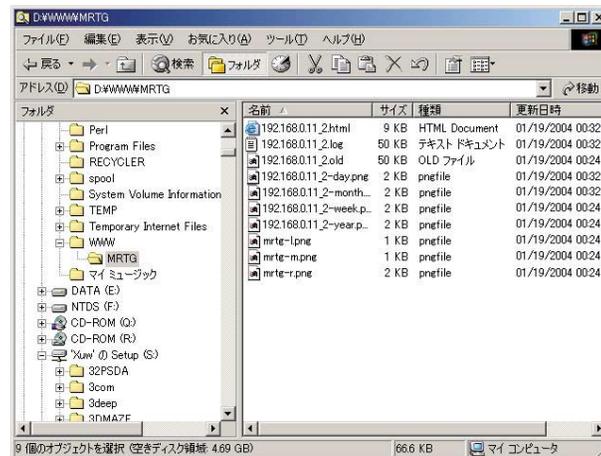
## 1 バッチファイルを作成する



テキストエディタなどで、コマンド行「perl mrtg mrtg.cfg」を反映したバッチファイルを作成する。ファイル名は既存のものと重複しないように注意する。また、複数の監視対象機器があって、機器ごとに「\*.cfg」ファイルを作成した場合には、複数行のコマンドを書いて反映させる



「コマンドプロンプト」から作成したバッチファイルを起動する



バッチファイルが正常に動作すれば、「HTMLファイル」「logファイル」などが更新され、「更新日時」のタイムスタンプが変更されるはずである。エクスプローラで確認しておこう

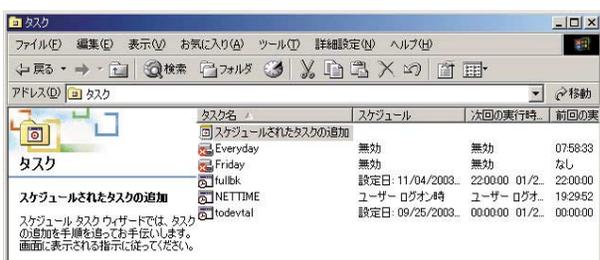
## バッチファイルの作り方

バッチファイルとは、コマンド操作を記述した「プログラムファイル」である。複数のコマンド操作をまとめたり、一行のコマンド操作に長い文字列を入力する手間を省略できたりする。バッチファイルを作るには次のように操作すればよい。「\*.bat」という拡張子を持つテキストファイルを作成する そのファイルを右クリックして、「編集」を選択する 必要なコマンド操作

を入力する 「上書き保存」を行う ダブルダブルクリックで実行してみる。

一連の操作をバッチファイルで簡単に呼び出すことで、操作は楽になるだろう。断続的な実行が必要なMRTGでは、このバッチファイルを「タスク」に登録することで、5分間隔の自動実行を可能にしている。

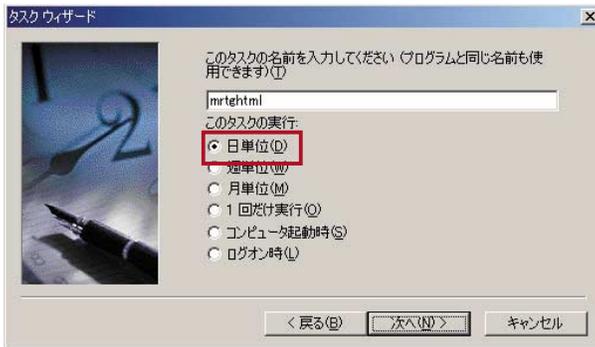
## 2 バッチファイルをタスクに登録する



バッチファイルの起動が確認できたら登録する。「コントロールパネル」の「タスク」を開いて、「スケジュールされたタスクの追加」をダブルクリックする。「タスクウィザード」が起動するので、「次へ」をクリックする



実行するプログラムの選択画面で、「参照」をクリックする。「ファイル名」には、で作成したバッチファイルを選択して、「開く」をクリックする



「タスクの実行」では「日単位」を選択し、「次へ」をクリックする。それに続く「開始する日時/時間」では、そのまま「次へ」をクリックする

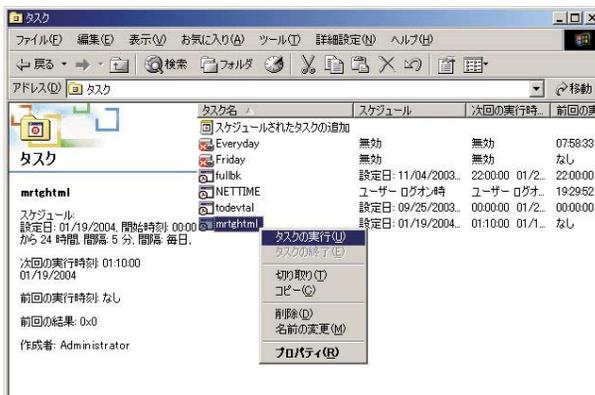


このスケジュールを実行するユーザーとそのパスワードを入力して、「次へ」をクリックする。最後に確認画面が表示されるので、「完了」を押したときに詳細プロパティを開く」にチェックを入れて、「完了」をクリックする

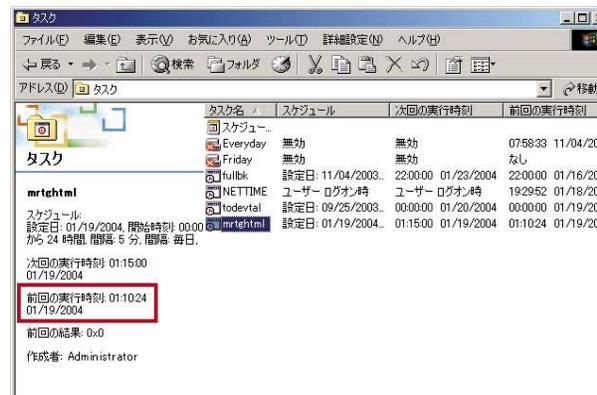


「プロパティ」が開いたら、「スケジュール」タブを選択して、「開始時刻」が「00:00」になっていることを確認する。「詳細設定」をクリックする

「タスクを繰り返し実行」にチェックを入れ、間隔を「5分」「継続期間」を「24時間」に設定して、「OK」をクリックする。「プロパティ」を閉じて、「パスワードの設定」が表示された場合には入力して、「OK」をクリックする



作成したタスクが登録されている。対象ファイルを右クリックして、「タスクの実行」をクリックする



「前回の実行時刻」に日時が表示されていれば、設定したタスクの実行が開始されている。HTMLファイルのタイムスタンプも更新されているはずなので、エクスプローラで確認しよう

### 設定ポイント3

## IISを追加してリモートアクセスを可能にする

### Multi Router Traffic Grapher

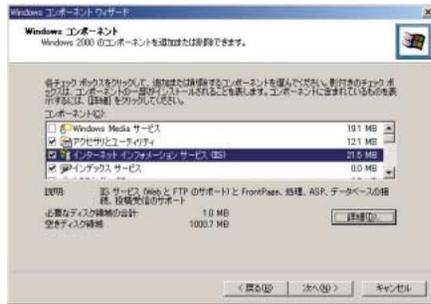
ネットワークの規模がそれほど大きくなければ、MRTGが動作しているPCにリモートからアクセスして、MRTGが作成した

HTMLファイルを確認できるように設定しておく。Windowsどうしの場合、「D:¥WWW¥MRTG」を共有すれば、エクスプロー

# Part2 使いこなしの事前設定ポイント

ラを利用して「マイネットワーク」越しにアクセスできる。しかし、大きなネットワークでは、IISを使うと管理が煩雑になるので、ネットワーク管理者だけがアクセスできる安全なWebサーバを構

築する必要があるかもしれない。ここでは、Windowsに標準で用意されているIISを利用して、Webアクセスを可能にする方法を紹介する。



「コントロールパネル」の「アプリケーション(プログラム)の追加と削除」を開いて、「Windowsコンポーネントの追加と削除」をクリックする。ウィザードが表示されたら、「コンポーネント」の一覧から「インターネットインフォメーションサービス(IIS)」を選択して、「詳細」をクリックする



「コントロールパネル」の「管理ツール」から「インターネットサービスマネージャ」を起動する。このコンピュータ名が表示されるはずである。コンピュータ名をクリックすると、右ペインに各サイトが表示される。「既定のWebサイト」を右クリックして、「プロパティ」を選択する

「WWW(World Wide Web)サーバ」を選択後、「OK」をクリックして、「次へ」をクリックする。ファイルのコピーが始まる



## MRTG上級テクニク

### トラフィックだけじゃない! CPU使用率やメモリ使用量を測る

MRTGは、SNMPが管理するオブジェクトデータベース「MIB」に用意されているデータも収集できる。ただし、MIBの「Object ID(OID)」の知識が必要なので、トラフィック測定に慣れてから使ったほうがよい。今回は、サーバのリモート監視項目として、「CPU使用率」「メモリ使用量」を計測するため、Windows ServerのMIBを拡張する「SNMP4Win2k(無償)」を利用する。

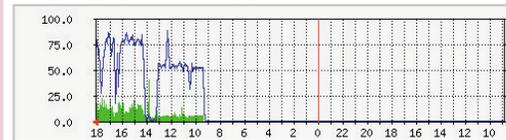
「SNMP for the Public Community」のWebサイト(<http://www.wtcs.org/snmp4tpc/default.htm>)から、「SNMP4Win2k」をダウンロードしてインストールする。続いて、「\*.cfgファイル」に次の内容を反映させると、MRTGで取得したい情報を追加できる。OIDはSNMP4WIN2Kの「Readme.txt」ファイルにの記載されているので参照してもらいたい。

Target[統計情報名]: OID1&OID2:コミュニティ名@SNMPエージェントのIPアドレス  
(CPU使用率を測定する例)  
Target[Processor]:1.3.6.1.4.1.311.1.1.3.1.1.2.1.5.1.48&1.3.6.1.4.1.311.1.1.3.1.1.2.1.4.1.48:  
public@IPアドレス:

#### Processor Time

The statistics were last updated Sunday, 8 February 2004 at 18:14

^Daily' Graph (5 Minute Average)



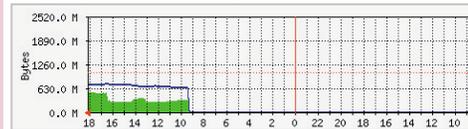
Max Priv: 41.0 % Average Priv: 10.0 % Current Priv: 20.0 %  
Max User: 88.0 % Average User: 55.0 % Current User: 72.0 %

MRTGが生成したCPU使用率のグラフ

#### Available Bytes Committed Bytes

The statistics were last updated Sunday, 8 February 2004 at 18:14

^Daily' Graph (5 Minute Average)



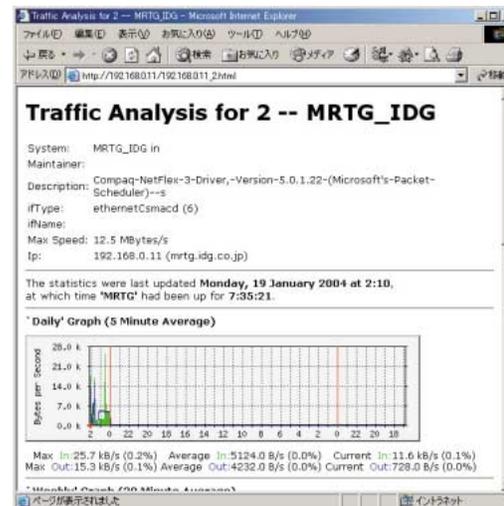
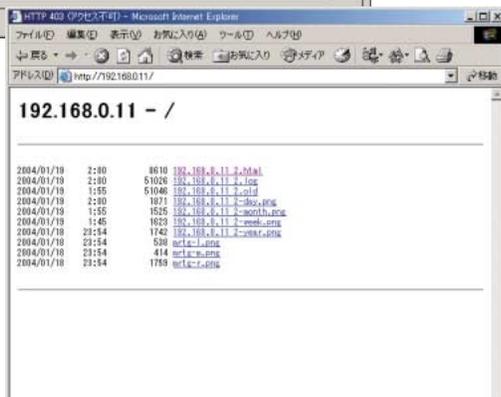
Max Avail : 548.8 Mbytes (52.4%) Average Avail : 356.5 Mbytes (34.0%) Current Avail  
Max Commit: 762.9 Mbytes (30.3%) Average Commit: 707.8 Mbytes (28.1%) Current Comm

MRTGが生成したメモリ使用量のグラフ

## < 特別企画 > MRTG で Windows ネットワーク監視



「ホームディレクトリ」タブを選択する。「ローカルパス」の「参照」をクリックして、「D:\WWW\MRTG」を反映させる。また、「ディレクトリの参照」にチェックを入れて、「OK」をクリックする。「読み取り」「ログアクセス」にはデフォルトでチェックが入っている



トラフィック量を反映したグラフが表示された。これは「タスク」で設定したバッチファイルがMRTGを自動起動して、トラフィックを収集しているためである

Webブラウザで「http://<IP address>/」を入力すると、MRTGが作成したファイルのリンクが表示されるので、知りたい情報のファイルをクリックすればよい。なお、このIPアドレスは、設定ポイント1で調べた自分のIPアドレスである

### 設定ポイント4

## 周辺ネットワーク機器の情報を収集する

### Multi Router Traffic Grapher

「Ciscoルータ」や「Catalystスイッチ」など、トラフィックを測定したいネットワーク機器(SNMPエージェント)がある場合、MRTGからアクセスできるように設定にする。その方法は、設定ポイント1のPCと同じく、SNMPコミュニティ名を「public」に設定すればよい。

### デフォルトコミュニティ名 「public」はセキュリティホール

今回の設定例では、PCでもCiscoルータでも、「SNMPコミュニティ名」として「public」を使用している。ところが、「public」という文字列は、SNMPの世界で周知されたデフォルト値として設定されていることが多い。つまり、だれでも知っているので、「public」でアクセスしてきたユーザーに対して、データを開示してしまうことになる。これはセキュリティホールなりえるだろう。そこで、管理者だけが知り得る、一般には想像もつかない文字列を設定することが望ましい。



「コマンドプロンプト」を開く。「telnet」コマンドでIPアドレスを入力して、Ciscoルータにアクセスする



「パスワード」を入力 「enable」コマンドで特権モードに移行 「configure terminal」コマンドでグローバルコンフィギュレーションモードに移行 「snmp-server」コマンドで、「community」名を「public」に設定する

## Part3 | ボトルネックや最高実効速度を探る!

# グラフ分析の実践ケーススタディ

インストールと設定作業が無事に終了すれば、早速MRTGを使ってグラフに出力してみる。ここでは、一般的なオフィスやブロードバンド環境での事例を3つ紹介しながら、グラフの「読み方」を練習する。グラフに隠された情報を発見できれば、障害予測につなげられる。

### Case1

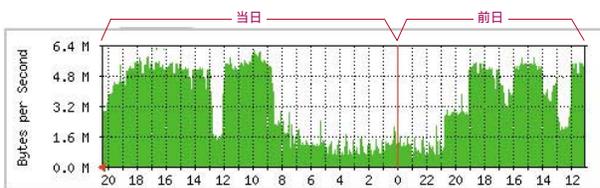
## 帯域飽和の前兆を読み取る

### Multi Router Traffic Grapher

まずは、ある職場のネットワークトラフィックを例に、グラフの基本的な読み方を習得しよう。MRTGが作る「Dailyグラフ」では、時間軸(横軸)の「0」=深夜0時を基点にして、その右側が前日、左側が当日となっている(画面3)。

この職場では、「9時始業」「12時~13時昼休み」「18時終業」という勤務時間体系を採用している。画面3のグラフからは、その勤務時間と合致するように、お昼休みの時間帯のトラフィックが減少していることを読み取れる。また、18時以降は就業時間帯より少なくなるが、それでも一定のトラフィックが観測できる。残業している社員がいるからだろうか。

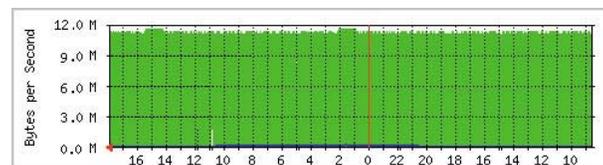
具体的な帯域速度に目を向けると、就業時間帯は定期的に「約5Mバイト/秒」のトラフィックがある。そこで、「前日」を詳細



画面3 一般的な職場のトラフィック。就業時間は、お昼休みを除く9時~18時であるが、それ以外の時間にもトラフィックを観測できるのは、どうだろうか？

に観察すると、19時にトラフィックが半減し、21時近くに深夜と同じトラフィックに落ち込む。このことから、就業時間後も半数近くの社員がネットワークを利用しており、そのほとんどが21時近くまで仕事をしているのだろう。一方、「当日」のトラフィックを観察すると、18時以降はなだらかにトラフィックが減少している。社員がまんべんなく仕事を切り上げていることがわかる。

ネットワーク管理者の仕事としては、9時から18時の間に観測されるトラフィックの“増加傾向”を読み取ったら、「Monthly(月間)」や「Yearly(年間)」のグラフで帯域が飽和する時期を予測し、それまでにネットワーク帯域をアップグレードすることが望まれる。すでに定期的に飽和していれば、早急なアップグレードを要するだろう(画面4)。



画面4 定期的に帯域が飽和しているネットワークでは、その早急なグレードアップが求められる

### Case2

## ボトルネックはどこにあるのか？

### Multi Router Traffic Grapher

筆者は社内で複数のサーバを運用している。そのデータのバックアップは、ネットワーク越しに行っていて、バックアップ先のサーバには、テープドライブ装置を実装している。また、バックアップ先とバックアップ元のサーバは、スイッチングハブに接続されていて、スイッチングハブの各ポートでは、そのサーバに宛

てたトラフィックと、サーバから発信したトラフィック以外は通過することがない。

バックアップは、休業日の毎週日曜日に行っている。休業日に行うのは、ユーザーによるデータ変更がないからだ。ところが困ったことに、運用当初はフルバックアップでも3時間あれば

完了していたのが、データ容量が増えた現在では、5時間以上もかかるようになってしまった。そこで、バックアップ時間を短縮する方法をいくつか考えた。1つは、バックアップ装置をより高速なテープドライブ装置に移行する方法である。もう1つは、バックアップ先のサーバにつながる回線速度を100Mbpsから1Gbps(ギガビットEthernet)に移行する方法である。

この2つの方法うち、どちらがバックアップ時間を短縮するのに適しているだろうか。これを判断するのに役立つのが、MRTGが出力するトラフィック情報のグラフである。まずは、現時点でのバックアップトラフィックをMRTGでグラフにして、観察しよう。最初に、ユーザーからのアクセスがない休業日に、バックアップ先とバックアップ元でトラフィックを収集する。同様に、バックアップ先とバックアップ元のサーバがそれぞれ接続されているスイッチングハブのポートのトラフィックを収集する。また、参考情報として、どちらのサーバにも接続されていないスイッチングハブのほかのポートの情報も収集した(画面5)。

これらの5つのグラフから、このネットワークのトラフィックの特徴として、次の3点のことが判断できる。筆者がこう判断した理由もいっしょに考えてもらいたい。

バックアップ先のサーバには、バックアップしているデータ以外のアクセスはない。

バックアップ元のサーバには、バックアップ先のサーバ以外からのアクセスはない。

バックアップ先とバックアップ元のサーバへのトラフィック以外に、ネットワーク上にトラフィックはない。

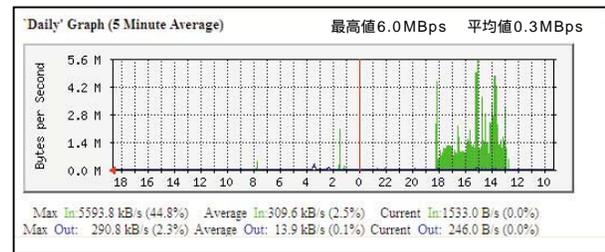
このように判断できたのは、バックアップ先とバックアップ元で計測されたのグラフを比較すると、トラフィックの方向が「In」と「Out」の逆になっているだけで、大きさや波形が一致しているからである。バックアップ先からほかの方向へのアクセスや、バックアップ元以外からのアクセスがバックアップ先に存在すると、グラフの波形は一致しない。

では実際に、グラフの数値を確認してみよう。MRTGが作るグラフの縦軸はバイト/秒(Bps)である。ところが、一般的なNIC(Network Interface Card)やルータ、ハブのポート速度は「ビット/秒(bps)」で表記されるので、周辺機器も考慮しながら、その速度を比較する場合は、1バイト=8ビットで換算してから行う必要がある。

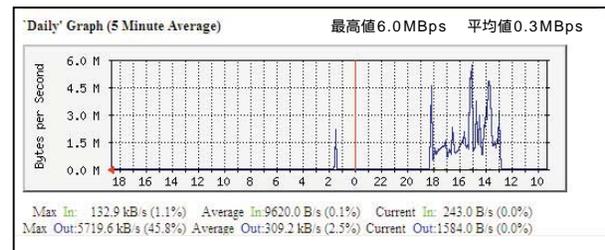
このグラフでは、最高値が約6.0MBpsなので、これを換算すると「48Mbps」の最大速度を実現していることになる。同じく平均値0.3MBpsは、「2.4Mbps」になる。一方、実装しているNICとスイッチングハブは、100Mbpsのスペックを持つ。

ここで、グラフから読み取らなければならない情報は、ピークの最高値が「一瞬」であるということ。つまり、定期的に48Mbps

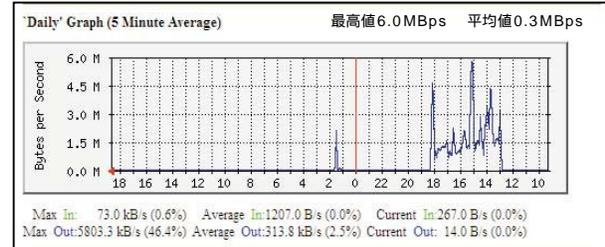
バックアップ先のサーバ(方向はIn)



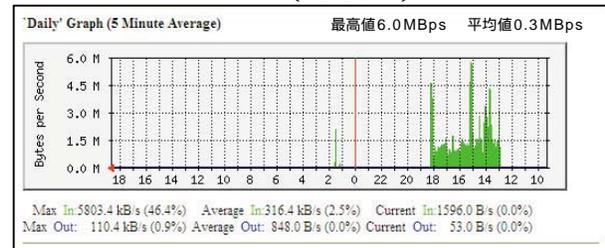
バックアップ元のサーバ(方向はOut)



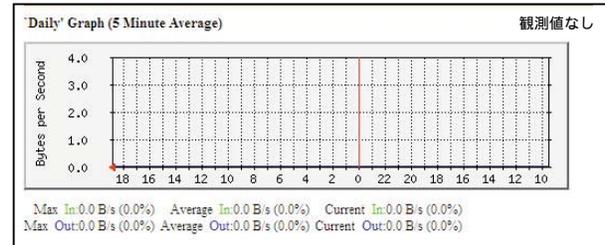
バックアップ先のサーバが接続された  
スイッチングハブのポート(方向はOut)



バックアップ元のサーバが接続された  
スイッチングハブのポート(方向はIn)



サーバに接続されていない  
スイッチングハブのポート



画面5 サーバとスイッチングハブのポートで取得されたトラフィックデータのグラフ(画面内の「最高値」「平均値」は筆者による書き込み)

# Part3 グラフ分析の実践 ケーススタディ

でトラフィックが流れているわけではないので、ネットワークの実効速度が仮に「50Mbps」だとしても、バックアップを行っている全時間帯でみれば、まだまだ帯域は飽和していないと考えられる。よって、バックアップ時間を短縮化するのに、ギガビットEthernetに移行する方法は適切な対策にならないと判断できる。

もし、バックアップを行っている時間帯に、定期的に帯域が飽和していたら、帯域を太くすればよいが、今回のケースでは、ネットワーク以外のところにボトルネックがあるのだろう。筆者は、より高速なテープドライブ装置を導入するほうが、バックアップ時間の短縮につながると判断した。

Case3

## 実効スループットの限界値を知る

Multi Router Traffic Grapher

NTT東西「Bフレッツ」や東京電力「TEPCOひかり」などを使い、100Mbps級の帯域でインターネットに接続する個人ユーザーが増えている。これらのネットワークでは、その帯域を最大活用してもらうため、「3～6Mbps」のストリーミングコンテンツが提供されている。

ここでMRTGを使った実験をしてみよう。複数のPCとブロードバンドルータを使って、これらのストリーミングコンテンツを閲覧すれば、どの程度のトラフィックで帯域が飽和するかを把握できる。もちろん100Mbps級の帯域を導入していないユーザーでも計測可能だ。

この実験は、FTPを使ったファイルのダウンロード時間を計測する「スピードテスト」などとは違い、「一定時間に、これ以上多くのデータは流れない」という計測値を出力できる。利用している回線やネットワークの帯域幅、ブロードバンドルータの実効スループットを求められるだろう。

実験環境には、「Cisco 2600」「Catalyst 2900」など、SNMPに対応しているブロードバンドルータやスイッチングハブを用意する必要がある。一方、SNMPに対応していない家庭用ブロードバンドルータしか手元にない場合でも、Windows ServerやLinux/FreeBSDなどでPCルータ(SNMPエージェント)を構成して、ブロードバンドルータとLANの間に設置する。その結果、全PCのストリーミングデータのダウンロードトラフィックを計測できるようになる(図2)。

具体的な実験方法としては、複数のPCでストリーミングコンテンツを「Windows Media Player」の連続再生モードで視聴して、グラフで帯域の飽和点を観測すればよい。筆者は、8台のPCを用意して、6Mbpsのストリーミングデータを連続再生することにした。どのPCでも6Mbpsで再生できるなら、次のような数値(平均値)が続くグラフが出現するだろう。

$$6\text{Mbps} \times 8\text{台} = 48\text{Mbps}(\text{ビット/秒}) = \underline{6\text{MBps}(\text{バイト/秒})}$$

計測は17時30分から翌朝11時30分の時間帯で行った。その結果、計測開始から深夜1時ごろまでは、いくつかのPCでコマ

落ち的な画像の乱れがランダムに観測された。MRTGのグラフでも、平均5MBpsの帯域を確認できた(画面6)。この時点では、ブロードバンドルータのスループットが「5MBps = 40Mbps」程度で限界なのか、WAN回線が飽和しているのか不明であった。ところが、深夜1時を過ぎてから翌朝11時30分の計測終了までには、コンスタントに「6MBps = 48Mbps」のグラフを取得できた。この差はブロードバンドルータのWAN側の回線がベストエフォートで提供されており、17時30分から深夜1時までは、ほかの利用者と同時に帯域を利用していたことが原因だと思われる。深夜1時を過ぎると、ほかの利用者が少なくなったのだろう。最終的に、使用したブロードバンドルータは少なくとも48Mbpsの実効スループットを持っていることがわかった。

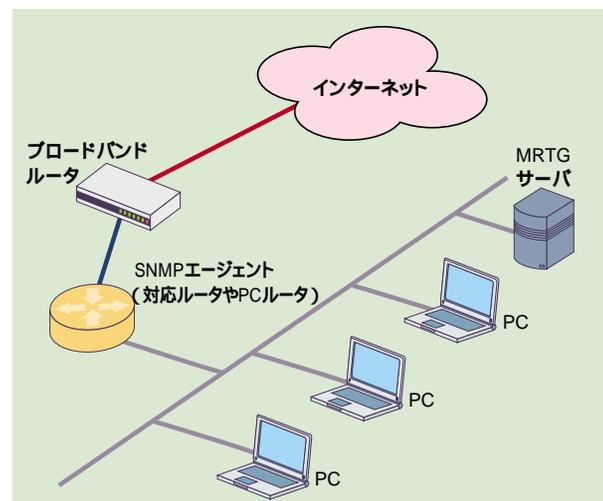
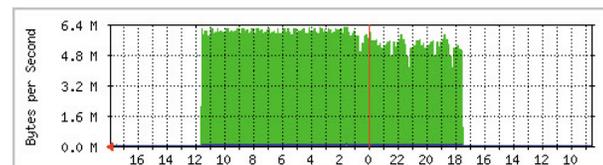


図2 Case3で使用したネットワーク構成例。LAN側のPCでダウンロードしている全トラフィックがSNMPエージェントを通過する



画面6 PC8台を用意して、6Mbpsのストリーミングコンテンツを視聴した際のグラフ。ある時間を境に、グラフの変化が読み取れる