

Network Management with MRTG

All Dressed Up

MRTG's speciality is monitoring network traffic and displaying the results graphically. The Multi Router Traffic Grapher also retrieves miscellaneous SNMP variables and can be customized to fulfill even the more exotic desires of the administrator.

BY WILHELM BOEDDINGHAUS

Even the dry kind of statistics that SNMP agents continually create can be appreciated at a single glance when visualized. The Multi Router Traffic Grapher (MRTG)[1] by Tobias Oetiker is the classic tool in this area. Released under GPL, MRTG monitors network traffic, queries routers and switches, and creates concise graphs of the collated data embedding them in a website. But MRTG has another couple of tricks up its sleeve, such as querying network interfaces for error messages, or monitoring hard disk loads.

The following examples are based on a Linux router running on SuSE Linux 7.3, fitted with two network interface cards and a single hard disk. It is MRTG's job to monitor the network traffic, while at the same time watching out for NIC errors and hard disk capacity issues on a single partition.

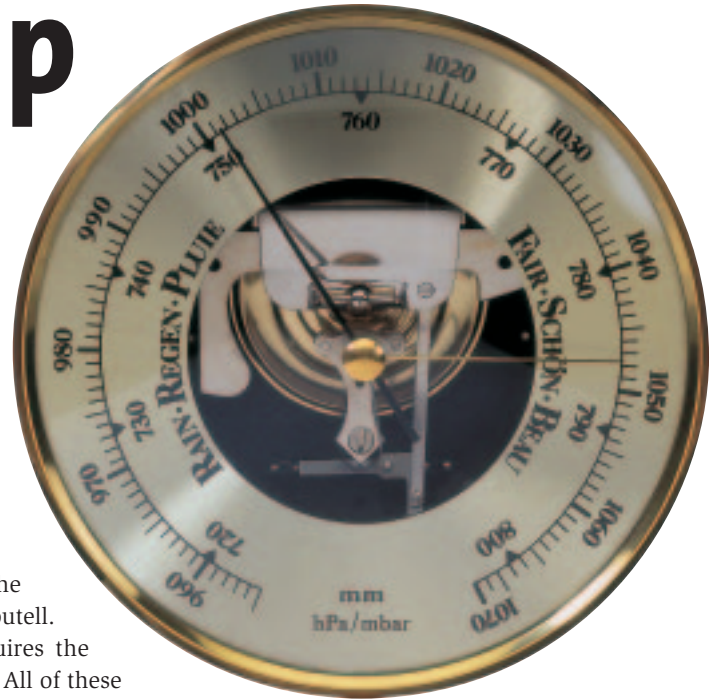
Various Linux distributions include both MRTG and the additional programs and libraries it requires. If this happens not to be true of your distribution, you should not find it too difficult to perform

a manual installation. To do so you will need Perl, a C compiler and the GD library by Thomas Boutell. This library in turn requires the Libpng and Zlib libraries. All of these components will run on Linux, Unix and even Windows.

Auto-Configuration

MRTG needs a configuration file for each device it is to monitor, however the file can be generated automatically. The "cfgmaker" program from the MRTG package writes a configuration file that allows MRTG to monitor network traffic. "indexmaker" creates an HTML index page containing an overview of the devices being monitored.

Using the standard configuration file created by "cfgmaker" MRTG will use SNMP to monitor the network interface of the devices in question. It does not matter whether you are dealing with a server equipped with a single network interface or a router, or switch, with multiple interfaces.



The prerequisites are that the device you are monitoring can produce SNMP data and that MRTG has read privileges for SNMP on that device.

You will need to supply all of the following information in the "cfgmaker" command line:

- IP address or DNS name of the device to be monitored, this is "192.168.33.1" in our example,
- Community String: "secret",
- Name of the config file you want to create: "/usr/local/mrtg/linux.cfg",
- Path for storing the HTML pages: "/usr/local/mrtg/html"
- and the additional option: "growright".

MRTG stores the graphs and the collected data in the same directory as the HTML pages. If you intend to access the pages via a Web server, then the

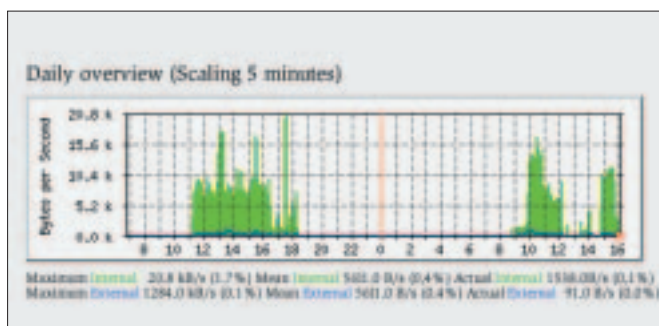


Figure 1: The MRTG graph shows the network has enough capacity. The maximum load is a mere 1.7 percent (lower left value, "Maximum")

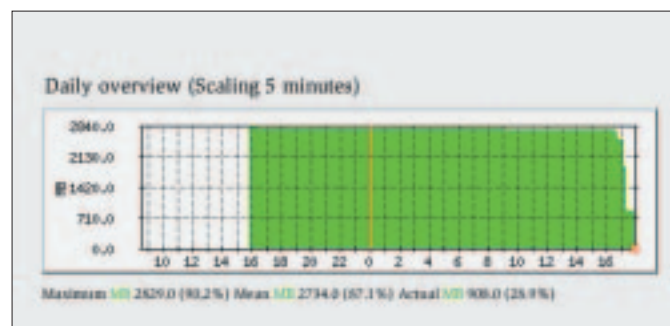


Figure 2: The partition originally contains 2829 Mbytes of data, but this value drops to 908 Mbytes after the user tidies up her hard disk

server will also need read access to the HTML directory.

The default setting displays newer values on the left of the graph, but you can place them on the right by setting the “growright” flag. We want to display the background in light gray and use the interface name, “eth0” for example, instead of a serial number (default) to describe the interface:

```
cfgmaker ➤
--output=➤
/usr/local/mrtg/linux.cfg ➤
--global ➤
"workdir:/usr/local/mrtg/html" ➤
--global "Language:english" ➤
--global ➤
"options[_]: growright" ➤
--global ➤
"Background[_]: #eeeeee" ➤
--ifdesc=descr ➤
geheim@192.168.33.1
```

The results are stored in “/usr/local/mrtg/linux.cfg”. The name of this file is passed to MRTG in the first argument: “mrtg /usr/local/mrtg/linux.cfg”. This syntax creates an HTML page for each interface, logfiles containing the acquired data, and the graphs in the “/usr/local/mrtg/html/” directory. You will need to create this directory before you enter the command. MRTG automatically deletes older data – this results in an error message when you first launch the program, but you can safely ignore the message. Check out Figure 1 for the results.

The MRTG command needs to be launched every five minutes as a cron job. This interval is important to allow

MRTG to calculate mean values correctly. You can set the interval in the configuration file.

Manual Configuration

If you require other SNMP variables than the transferred volume of data, then you must provide a configuration file by hand. The example in listing 1 queries the errors, which arise on a network interface. The global options correspond to those for the data transfer, with other fields explained in table 1. The call, again Cron steered, should take place every five minutes.

External Data Sources

Instead of using SNMP directly, you can also integrate your own scripts and programs that acquire data externally and pass it on to MRTG. This allows you to visualize metrics without using an SNMP agent. It is often easier to use an existing SSH interface to a machine than to set up an SNMP agent on the machine.

The article on OpenSSH in the “Sysadmin” column of this issue shows you how to use cron to call SSH without cron needing to store the password or passphrase. A script can also collate statistics from multiple sources or process the statistics in some other way. The following example uses a Perl script (see Listing 2) to ascertain how much space has been used up on the first hard disk partition. To do so, the script needs to work with two SNMP variables:

- The block size is stored in the “hrStorageAllocationUnits” variable,
- the number of blocks used is stored in “hrStorageUsed”.

The script uses the Perl module “NET::SNMP” to retrieve these two values and then calculates the disk space used in Mbytes. The script expects both the host name and the community string as arguments.

To call the Perl script, the “Target” entry in the MRTG configuration file (see Listing 3) must include the name of the script and the arguments in backticks “`”. The first parameter is the host address “192.168.33.1” followed by the community string “public”.

Differences

The “gauge” option tells MRTG to store the value exactly as it was read, instead

Listing 1: MRTG showing network errors

```
WorkDir: /usr/local/mrtg/html
Language: english
Background[_]: #eeeeee

Target[interfaceerrors_2]:➤
1.3.6.1.2.1.2.2.1.14.2&1.3.6.1.➤
2.1.2.2.1.20.2:secret@192.168.➤
33.1
Options[interfaceerrors_2]:➤
growright,nopercent
Title[interfaceerrors_2]:➤
Error Interface eth0
MaxBytes[interfaceerrors_2]:➤
10000
YLegend[interfaceerrors_2]:➤
Error
ShortLegend[interfaceerrors_2]:➤
Legend1[interfaceerrors_2]:➤
Input Error
Legend2[interfaceerrors_2]:➤
Output Error
LegendI[interfaceerrors_2]:➤
INPUT
LegendO[interfaceerrors_2]:➤
OUTPUT
PageTop[interfaceerrors_2]:➤
<H1>Input / Output Errors</H1>
Error on Interface eth0

Target[interfaceerrors_3]:➤
1.3.6.1.2.1.2.2.1.14.3&1.3.6.1.➤
2.1.2.2.1.20.3:geheim@192.168.➤
33.1
Options[interfaceerrors_3]:➤
growright,nopercent
#... additional entries for eth1
```

Table 1: Configuration Options

Option	Meaning
Target	With the Target keyword you tell mrtg what SNMP variables it should monitor. The Target keyword takes arguments in a wide range of formats
Options	The Options Keyword allows you to set some boolean switches: nopercent is used when you do not want to print usage percentages
Title	Title of the produced HTML site
MaxBytes	The upper limit necessary in order to scale the axes and to compute percentages.
YLegend	The Y-axis label of the graph.
ShortLegend	The units string
LegendI	The strings for the colour legend for Incoming
LegendO	The strings for the colour legend for Outgoing
Legend1, Legend2	The strings for the colour legend
PageTop	Things to add to the top of the generated HTML page. Note that you can have several lines of text as long as the first column is empty.

Listing 2: External Script

```
#!/usr/bin/perl -w
# MRTG Script calculates ↗
amount of disk space used

use Net::SNMP;

# Object IDs of SNMP Variables
my $uptimeOID = ↗
'.1.3.6.1.2.1.1.3.0';
my $nameOID = ↗
'.1.3.6.1.2.1.1.5.0';
# hrStorageAllocationUnits
my $unitsOID = ↗
'.1.3.6.1.2.1.25.2.3.1.4.1';
# hrStorageUsed
my $usedOID = ↗
'.1.3.6.1.2.1.25.2.3.1.6.1';

# Retrieve values
($session, $error) = ↗
Net::SNMP->session(
    Hostname => $ARGV[0],
    Community => $ARGV[1]);
the "Session-Error: $error" ↗
unless ($session);
# Uptime and name
$result = $session->
get_request($uptimeOID);
$uptime = $result->{$uptimeOID};
$result = ↗
$session->get_request($nameOID);
$name = ↗
$result->{$nameOID};
# Block size and number of ↗
blocks in use
$result = ↗
$session->get_request($unitsOID);
$units = $result->{$unitsOID};
$result = ↗
$session->get_request($usedOID);
$used = $result->{$usedOID};
# Convert space used to Mbytes ↗
$usedMB = int (($units * ↗
$used) / (1024 * 1024));

# Pass values to MRTG
print "$usedMB\n";
print "0\n"; ↗
# second value is not displayed
print "$uptime\n";
print "$name\n";
```

of storing the difference to the last value. If you are attempting to obtain a data transfer statistic, you would need the difference between the two values as SNMP agents accumulate the total number of packets transferred, i.e. the value increases continually. An absolute value is appropriate for a hard disk statistic. The “noo” (no output) entry stops MRTG from outputting the second value; “noi” would prevent the first value from being displayed. However, the script will need to pass both values. MRTG expects four lines of output:

```
first value,
second value,
system uptime, and
name of system.
```

Listing 3: Configuration for an external script

```
WorkDir: /usr/local/mrtg/html
Language: english
Background[_]: #eeeeee

Target[harddisk]: ↗
`/usr/local/mrtg/perl/ ↗
mrtg-get-linux.pl 192.168.33.1 ↗
public`
Options[harddisk]: ↗
growright,noo,gauge
Title[harddisk]: ↗
hard disk usage
MaxBytes[harddisk]: 3138
Ylegend[harddisk]: MB
ShortLegend[harddisk]:
Legend1[harddisk]: ↗
hard disk usage in MB
Legend2[harddisk]: not used
LegendI[harddisk]: MB
LegendO[harddisk]: not used
PageTop[harddisk]: ↗
<H1>Hard disk usage</H1> ↗
harddisk "/"
```

INFO

- [1] Tobias Oetiker, MRTG Homepage: <http://www.mrtg.org>
- [2] RRD Tool: <http://people.ee.ethz.ch/↗~oetiker/webtools/rrdtool/>
- [3] MIB Central: <http://www.mibcentral.com/>
- [4] SNMP for the Public: <http://www.wtcs.org/snmp4tpc/>

MRTG writes the contents of the third and fourth lines at the top of the HTML pages, prepending the statistics. You can see the results in Figure 2.

Those who want to provide many statistics with MRTG, will be pleased with the “more indexmaker” feature. This program from the MRTG package produces a summary page of the MRTG graphs. This usually requires filters of several MRTG configuration files.

Keeping track of things with Indexmaker

The following example defines the output file and a single column overview. The page title is additionally supplied. The title stored in the “Title” variable in the configuration file will be used for each graph; this shows up as “hard disk usage” in our example.

```
/usr/bin/indexmaker ↗
--output=/usr/local/mrtg/html/↗
index.html ↗
--columns=1 ↗
--title "Status Linux Router" ↗
--section title ↗
/usr/local/mrtg/linux.cfg ↗
/usr/local/mrtg/↗
linux-errors.cfg ↗
/usr/local/mrtg/↗
linux-harddisk.cfg
```

The result is a HTML page with five daily graphs (Data Transfer IN, Data Transfer OUT, Errors IN, Errors OUT, harddisk). The sixth value has been masked and will be ignored by the Indexmaker. Each graph is a link: Just click on it to display the page with detailed graphs and statistics in your browser.

Prospects

MRTG is a flexible and powerful network and host monitoring system. However, the format used to store the statistical data prevents any further processing, as MRTG only logs mean values required for producing weekly, monthly and annual statistics. If you look close enough, you might be able to estimate the future load on your components, but you will be unable to make an entirely accurate prediction.

RRD[2] may prove a better tool for this task: The Round Robin Database was also written by MRTG’s author. ■