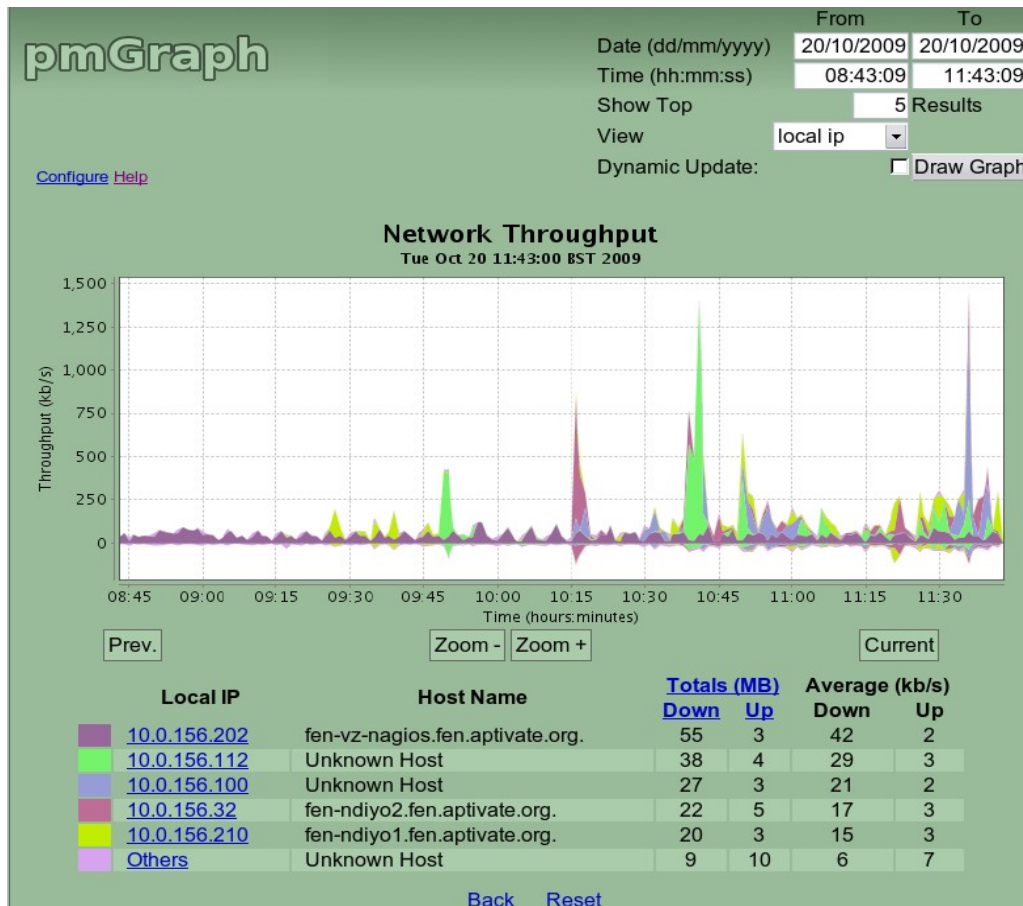


pmGraph Documentation

Date: 26/10/09 Version: 1.3

Please note: This documentation is provided as a reference copy and may not be as up to date as the pmGraph documentation available online. Please see <http://www.aptivate.org/Projects.BMOTools.pmGraph.html> for the most up to date information.



Aptivate

The Humanitarian Centre, Fenner's, Gresham Road, Cambridge CB1 2ES
 Telephone: +44 (0)1223 760887, Fax: +44 (0)1223 331126
 Website: www.aptivate.org Registered No: 4980791

Table of Contents

| | |
|--|----|
| Introduction | 3 |
| What is pmGraph? | 3 |
| PmGraph and pmacct | 3 |
| Main features | 3 |
| Usage and requirements | 4 |
| Installation | 4 |
| Installation from a Package | 4 |
| Third-Party software | 4 |
| Steps to install the software using the debian package | 4 |
| Manual Installation | 6 |
| Software required | 6 |
| Detailed steps to install above software | 6 |
| Common Installation Errors | 8 |
| Errors when you open the pmGraph page | 8 |
| If you can open the pmGraph page but graph is empty | 10 |
| User Guide | 13 |
| Getting started with pmGraph | 13 |
| pmGraph user interface | 13 |
| The Form | 13 |
| The graph | 14 |
| The legend | 14 |
| A real example | 15 |
| The pmGraph Configuration File | 17 |
| Configuration parameters explained | 18 |
| Web-based Configuration | 20 |
| The pmacct configuration file | 21 |
| Working with pmacct network monitoring software | 24 |
| Running pmGraph on a router with OpenWRT software | 24 |
| Setting up pmacct | 24 |
| Setting up MySQL | 25 |
| FAQs | 26 |
| General | 26 |
| Third party software | 27 |
| Intellectual properties and rights | 28 |
| Glossary | 28 |
| Help us to develop pmGraph | 29 |
| Contact Us | 29 |

Introduction

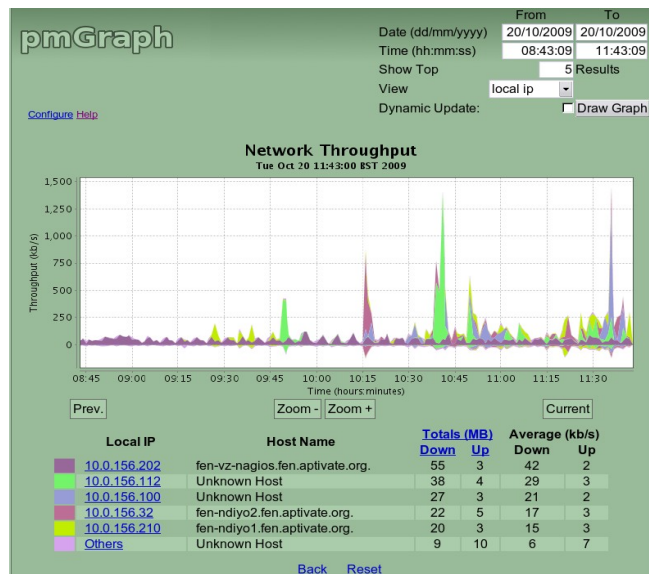
What is pmGraph?

pmGraph is a software application for network monitoring, which is an important part of bandwidth management and optimisation.

(<http://www.aptvate.org/Projects.BMOPositionPaper.html>)

The aim of pmGraph is to enable network administrators, especially at universities, to more easily monitor and manage their networks. pmGraph is the first component of our BMO Tools project

(<http://www.aptvate.org/Projects.BMOTOols.html>) to be released as free, open source software.



PmGraph and pmacct

pmGraph is a visualisation and graphing tool that is designed to complement pmacct (<http://www.pmacct.net/>), a network monitoring and auditing tool. The two tools are supplied together as a debian package, and [instructions for installing pmGraph](#) cover the installation of both tools.

pmacct monitors traffic on a firewall, router or bridge, or collects information from multiple routers, and stores the network data in a database. The database permits powerful analysis, however there is no user-friendly interface to access the data.

pmGraph addresses this issue by providing a graphical overview of the pmacct data, with a user friendly interface.

We have compiled a list of other available network monitoring software (<http://www.aptvate.org/Projects.BMOTOols.pmGraph.OtherNetworkMonitoringSoftware.html>) for reference and comparison.

Main features

- User friendly, simple interface
- Shows info about the connections between remote and local machines, and ports used
- Host name resolution using DNS and DHCP servers
- Shows usage for a specific IP address or port
- Configurable number of results

If you investigate pmGraph and decide not to use it, please let us know why (info@aptvate.org) in order to help us to improve it.

Usage and requirements

pmGraph is platform independent software. It has been developed in Java and is designed to work in a servlet container. However, pmGraph relies upon pmacct, which runs only on **Unix** based systems, so there must be at least one Unix based system in the network.

This graphing application is provided to compliment **pmacct** Network Monitoring Software. More details on using pmacct can be found [here](#), from where you will see that you may need a Cisco or Juniper router that supports [Netflow](#) or [sFlow](#).

pmGraph is quite lightweight, and requires only **8MB** of hard disk to run. However, it relies on other, bulkier programs. If you don't already have Tomcat, Java6, MySQL server and pmacct, you will need a total of 300 MB to get pmGraph running successfully. These components will be installed for you if you use the package installation and you do not need to an understanding of these to install and use pmGraph.

Installation

In this section we'll cover the steps needed to successfully install pmGraph and get it running. We'll also list the third-party software that is needed to get started. We also include a list of Common installation errors.

There are three options available for installing pmGraph:

- Use a debian package supplied by us which installs all the software needed.
- Use a manual process.
- Use an install script. Note: The installer will need root privileges to run.

pmGraph requires 8 MB of free space.

Installation from a Package

Third-Party software

Using this package the following software will be installed automatically:

- mysql database (<http://www.mysql.com/>)
- pmacct network traffic logger (<http://www.pmacct.net>)
- tomcat web serverWiki (<http://tomcat.apache.org>)
- Java Sun Development Kit
- pmGraph

Steps to install the software using the debian package

Install pmGraph using an APT repository:

1. Add the pmGraph project repository. To do so add the following lines to your `/etc/apt/sources.list` file

```
# app repository for PMGRAPH
deb http://ppa.launchpad.net/pmgraph/ppa/ubuntu hardy main
deb-src http://ppa.launchpad.net/pmgraph/ppa/ubuntu hardy main
```

2. From a terminal session, type this command to update the list of available packages:

```
sudo apt-get update
```

That should show you a message similar to:

```
http://ppa.launchpad.net hardy Release: The following signatures couldn't be
verified because the public key is
not available: NO_PUBKEY XXXXXXXXXXXXXXXX
```

```
W: You may want to run apt-get update to correct these problems
```

3. Import the PGP key of the repository

To import this key you will have to copy the number that appears after the text NO_PUBKEY and use it in the following command.

```
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com XXXXXXXXXXXXXXXX
```

4. Update the list of available packages:

```
sudo apt-get update
```

5. Make a search for "pmgraph"

```
apt-cache search pmgraph
```

you should receive a line like this:

```
pmgraph -pmGraph is a software application for network monitoring to work with
pmacct
```

6. Download and install pmGraph

```
sudo apt-get install pmgraph
```

7. The parameter *LocalSubnet* lets pmGraph know which IPs belong to your local network. You may need to change it to get pmGraph working. Do it by modifying the file database.properties.

```
nano /usr/share/tomcat5.5/webapps/pmgraph/WEB-INF/classes/database.properties
```

And modify the parameter *LocalSubnet* to make it match your local network. For example if your local network is 192.168.1.0/24 the value of the parameter has to be:

```
LocalSubnet = 192.168.1.
```

Then edit pmacct config file which is /etc/pmacct/pmacctd.conf and change the line:

```
pcap_filter: not (src and dst net 10.0.156.0/24)
```

To make it match your local subnet which for the above example will be:

```
pcap_filter: not (src and dst net 192.168.1.0/24)
```

8. Restart Tomcat server and pmacct.

```
sudo /etc/init.d/tomcat5.5 restart
sudo /etc/init.d/pmacct restart
```

9. Now you should be able to access the pmGraph page at.

```
http://{yourbridgemachine}:{yourtomcatport}/pmgraph/
```

For example, if you are running tomcat on the same machine as your web browser and using the default port, try to open

```
http://localhost:8180/pmgraph
```

For security reasons you need to change the default password by editing the configuration file. This also contains other parameters which you may wish to configure. If you have problems during the installation, you can check our list of [common installation errors](#). Otherwise see the [User Guide](#).

Manual Installation

Software required

You will need the following software installed to run pmGraph successfully:

- Mysql server. Package name **mysql-server**. You can use another SQL server, such as PostgreSQL, but this example will use mysql server.
- Pmacct. Package name **pmacct**.
- Tomcat server version 5.5. Package name **tomcat5.5**.
- Java Sun Development Kit version 1.6. Package name **sun-java6-jdk**.

Detailed steps to install above software

1. First of all, if you don't have it installed already, you will have to download mySQL (<http://dev.mysql.com/downloads/mysql/5.0.html#downloads>)

If possible, use a package, for example:

```
sudo apt-get install mysql-server
```

2. Then install the Java Sun Development Kit, for example using:

```
sudo apt-get install sun-java6-jdk
```

3. After that, you will need to download and install pmacct (<http://www.pmacct.net/>)

```
sudo apt-get install pmacct
```

You need to create the database that pmacct will write the traffic data into. You should find a bunch of SQL scripts in either the pmacct source code, if you downloaded it, or in some directory on your system if you installed a pmacct package. For example, if you install the Ubuntu 8.04 (Hardy Heron) pmacct package, you can find the MySQL script in `/usr/share/doc/pmacct/sql/pmacct-create-db_v6.mysql`.

Once you find the script, execute it like this (substituting the correct path within `pmacct-create-db_v6.mysql` on your system):

```
mysql -u root -p < /usr/share/doc/pmacct/sql/pmacct-create-db_v6.mysql
```

In the same directory you should find a file called `pmacct-grant-db.mysql`. Copy this file somewhere and edit the copy, changing the password from `arealsmartpwd` to a password of your choice. This avoids the need to give `pmacct root` access to your database, so it's more secure. Then load it into MySQL in the same way as the previous one:

```
mysql -u root -p < /usr/share/doc/pmacct/sql/pmacct-grant-db.mysql
```

You also have to download the configuration file, `pmacctd.conf`, from the code repository of the project:

<https://pmgraph.svn.sourceforge.net/svnroot/pmgraph/pmgraph/config/>

Overwrite the existing configuration file, which might be in `/etc/pmacct` depending on your distribution, and change the `sql_passwd` to match the new password you created. Also check the other parameters, especially if you are using another database because you may need to make some other changes. You can find out more about these in the [pmGraph Configuration File](#).

4. Now download and install Tomcat (<http://tomcat.apache.org/>). If possible, use a package for your system:

```
apt-get install tomcat
```

In the instructions below, `{tomcatroot}` refers to the directory where you installed Tomcat. If you installed a distribution package, check the package contents. For Ubuntu 8.04, it is `/usr/share/tomcat5.5`.

By default, in Ubuntu distribution the parameter `TOMCAT5_SECURITY` is set, which means that if you want to let any tomcat application access local services or files you need to add specific exceptions in the java policy file. pmGraph needs exceptions to use the mysql server and dns. The simplest way to avoid doing this is to set `TOMCAT5_SECURITY` to `no` in the file `/etc/default/tomcat5.5` by adding the following line:

```
TOMCAT5_SECURITY=no
```

If Tomcat is started without a display (automatically by the system at boot time, or on a remote shell), you will need to edit the Tomcat startup script `{tomcatroot}/bin/catalina.sh` (or `/usr/share/Tomcat5.5/bin/catalina.sh` on Ubuntu 8.04) and add this `JAVA_OPTS` line just above the `#OS specific support` line. This enables the chart library to work properly when Tomcat is started without a display.

```
JAVA_OPTS="-Djava.awt.headless=true"  
#OS specific support. $var _must_ be set to either true or false
```

Restart Tomcat after doing this:

```
sudo /etc/init.d/Tomcat5.5 restart
```

5. Now download the pmGraph WAR file from http://sourceforge.net/project/showfiles.php?group_id=238574. Rename it to pmGraph.war, and copy it into your {tomcatroot}/webapps directory. After a few seconds, Tomcat should automatically unzip it into a directory called pmGraph inside your webapps directory.

Inside this pmGraph directory you should find a file called WEB-INF/classes/database.properties. This file will need editing to customise various parameters:

LocalSubnet variable should match your local subnet. Enter a string prefix for your subnet, for example 192.168.2. matches any address under 192.168.2.0/24.

DatabaseURL should be set to the address for your MySQL database, if it is not installed locally

DatabasePass should be set to match the one that you created earlier

DatabaseTable should be set to match the pmacct database name, currently this is **acct_v6**.

DatabaseLongTable should be set to the name of the second table in which pmacct records data, should you wish it to do so. The default for this is acct_v6_long.

TimespansForLongGraph should be set to the number of hours the graph will cover before it switches to using data taken at a lower sample rate. The default is 24. Using a lower value will disable this feature.

Now start pmacctd to start monitoring your network in promiscuous mode (passive sniffer) with a command like this:

```
sudo pmacctd -f /etc/pmacct/pmacctd.conf
```

You may want to add this command to your startup scripts (e.g. /etc/rc.d/rc.local) to ensure that pmacct is started automatically every time the server is rebooted.

If you want to monitor your network another way (e.g. NetFlow or sFlow), please check the pmacct documentation at <http://www.pmacct.net/CONFIG-KEYS-0.11.5> and edit the configuration file before running pmacctd, nfacctd or sfacctd.

You should now be able to access the pmGraph page at:

```
http://{yourbridgemachine}:{yourtomcatport}/pmgraph/
```

For example, if running tomcat on the same machine as your web browser and using the default port, try to open

```
http://localhost:8080/pmgraph/
```

If you have problems during the installation, you can check our list of [common installation errors](#). Otherwise see the [User Guide](#).

Common Installation Errors

Errors when you open the pmGraph page

If you get this message while you are trying to access the website:

```
Unable to get a connection to Mysql server due to java security restriction.  
Disable java security or add necessary permissions (check the log file for more  
info)
```

You may be having problems with the security option set by default in Ubuntu. To disable it, you have to go to the file `Tomcat5.5` in `/etc/default` and write `TOMCAT5_SECURITY=no`. After that, restart the server.

```
sudo /etc/init.d/Tomcat5.5 restart
```

If you prefer not to disable it, you may need to add the necessary permissions in the policy file.

Problems with the policy file

We recommend that you disable the security option in Tomcat, as stated in the installation instructions. If you have decided to enable the security option in Tomcat, there are several errors that can be encountered if you have not added specific permissions in the Tomcat WebApps policy file. These are some error messages associated with this problem:

```
Unable to get a connection to Mysql server due to java security restriction.
Disable java security or add necessary permissions (check the log file for more
info).
```

```
A Security Exception has occurred while trying to create graph image.
This error is caused by Java security policy, disable java security or add a
suitable permission in policy file.
```

```
A Security Exception has occurred while trying to access sun.awt.* classes.
This error is caused by Java security policy, disable java security or add a
suitable permission in policy file.
```

```
Unable to access DNS server, check java security policy file.
```

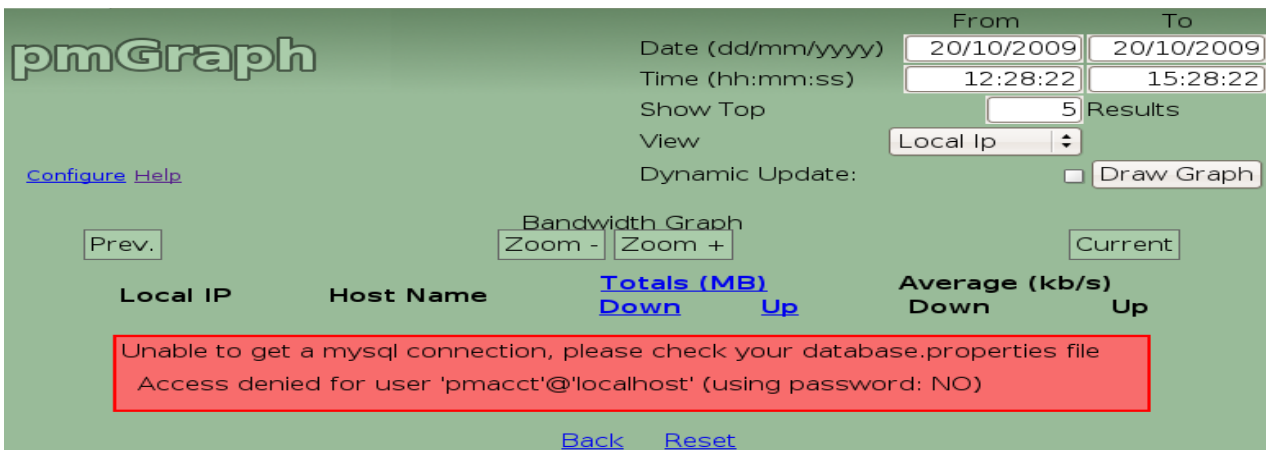
To grant the permissions needed, edit the file `04webapps.policy` which, in Ubuntu 8.04, can be found at `/etc/tomcat5.5/policy.d` and modify the file. Assuming you are using the default SQL server, add the following lines:

```
permission java.net.SocketPermission "localhost:3306-", "accept,connect,listen";
permission java.io.FilePermission
"/var/lib/tomcat5.5/pmgraph/WEB_INF/classes/logging.properties", "read";
permission java.awt.RuntimePermission "sun.awt.*";
permission java.util.PropertyPermission "dns.server", "read";
permission java.util.PropertyPermission "dns.search", "read";
permission java.io.FilePermission "/etc/resolv.conf", "read";
permission java.net.SocketPermission "DHCPserver address:53", "connect,resolve";
```

Problems with the pmGraph configuration file

The pmGraph configuration file has to be customized to match the user and password you have in mysql for pmacct, as well as with the database you created. This file can be found under the Tomcat directory `/usr/share/Tomcat5.5` at the following location: `webapps/pmGraph/WEB_INF/classes`. The name of the file is `database.properties`. The program will show you the same error when any of the parameters in the configuration file (user, password, database name...) are wrong:

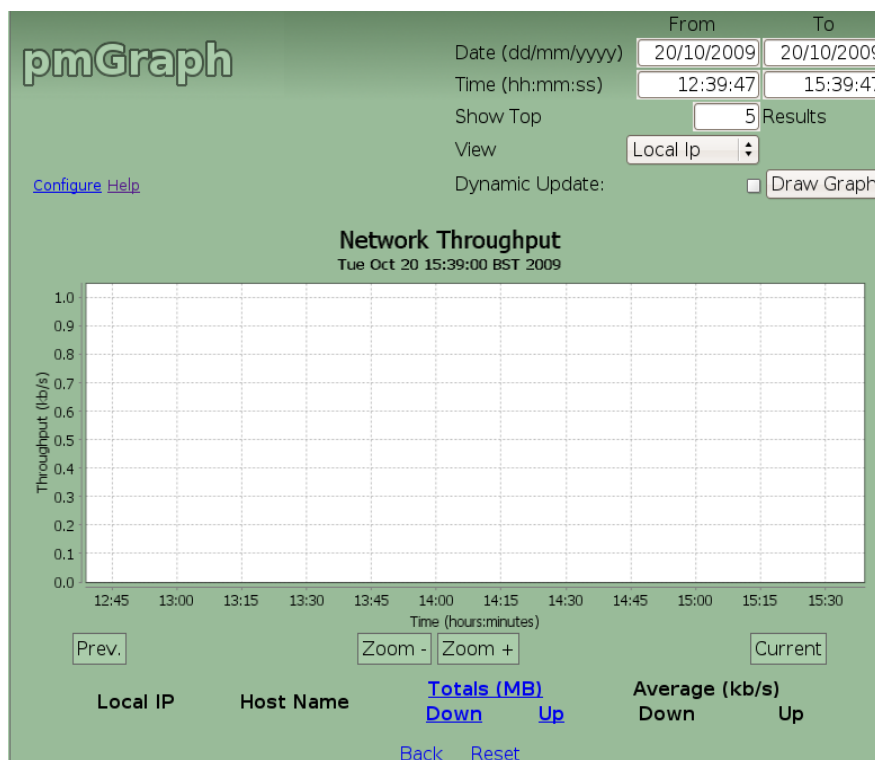
```
Unable to get a mysql connection, please check your database.properties file
```



To solve all these problems you have to open and edit this file, using an editor like *nano* or *vim*, and make the changes that are needed. Remember that you need superuser rights to be able to change the file.

To see a full explanation of the configuration file, go to the [User Guide](#).

If you can open the pmGraph page but graph is empty



you may have one of the following problems:

Problems relating to the database

Check if data is being stored. Go to mysql and try to connect with the user you created:

```
mysql -u "user" -p
```

If you can't do this, you may not have added the correct password in the pmacct configuration file to access to the database. Check it in the file `pmacctd.conf` (in `/etc/pmacct` on Ubuntu) and modify if necessary.

The same will happen if you have set the wrong user in the pmacct configuration file. More info about pmacct configuration parameters can be found at <http://wiki.pmacct.net/OfficialDocumentation>.

Wrong subnet

Another reason why your graph can appear empty is that the local subnet, configured in pmacctd.conf and in database.properties, doesn't match your actual local subnet. To solve it, just open both of them with an editor like vim or nano and modify them to match. Remember that you need superuser rights to modify these files.

If none of these problems match yours, you can try several things

Check java version

Type

```
java -version
```

at the command line. We recommend using Sun java sun-java6-jdk. GNU java can cause errors.

Check the configuration file for Tomcat

usually located at /etc/default/, modify it to disable the java_security option

```
TOMCAT5_SECURITY=no
```

If you don't do this you will have to add some permissions to the policy file, as above.

PGP key problem (NO_PUBKEY)

If after updating the list of available packages, you get this message:

```
http://ppa.launchpad.net/hardy/Release: The following signatures couldn't be
verified because the public key is
not available: NO_PUBKEY XXXXXXXXXXXXXXXX
```

W: You may want to run apt-get update to correct these problems

There is a problem with the key that was provided in the instructions. To check whether the key matches the one in the instructions, read the key at the end of the message (after NO_PUBKEY), if it doesn't match the default key you have to import it.

To import this key you have to copy the number that appears after the text NO_PUBKEY and use it in the following command (in place of XXXX).

```
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com XXXXXXXXXXXXXXXX
```

Now you need to update the repository.

```
sudo apt-get update
```

Trying on your own

If you haven't found your error in the list above and want to investigate further, modify the log file:

Enable "debug" level log in the file of log4j.properties located at
\$TOMCAT_HOME/webapps/pmgraph/WEB-INF/classes/

change this line:

```
log4j.rootCategory=info, stdout
```

to

```
log4j.rootCategory=debug, stdout
```

Now in the log directory for Tomcat in a file called catalina you should receive verbose output of what pmGraph is doing. If this file doesn't exist then try the `/var/log/syslog` file.

Contact us

If you can't find the solution for your problem, please send an email to info@aptivate.org telling us your specific problem. Please try to be as clear as possible and send us some screenshots with the error you encountered.

User Guide

Getting started with pmGraph

To start using pmGraph once it is installed, just use a web browser to access the URL:
`http://{yourtomcatservermachine}:{yourtomcatport}/pmgraph/`

For example if your tomcat server is called "tomcatserver" and uses the default port 8080, the URL would be:
`http://tomcatserver:8080/pmgraph/`

Or `http://tomcatserver:8180/pmgraph/` on Ubuntu where the default tomcat port is 8180.

pmGraph user interface

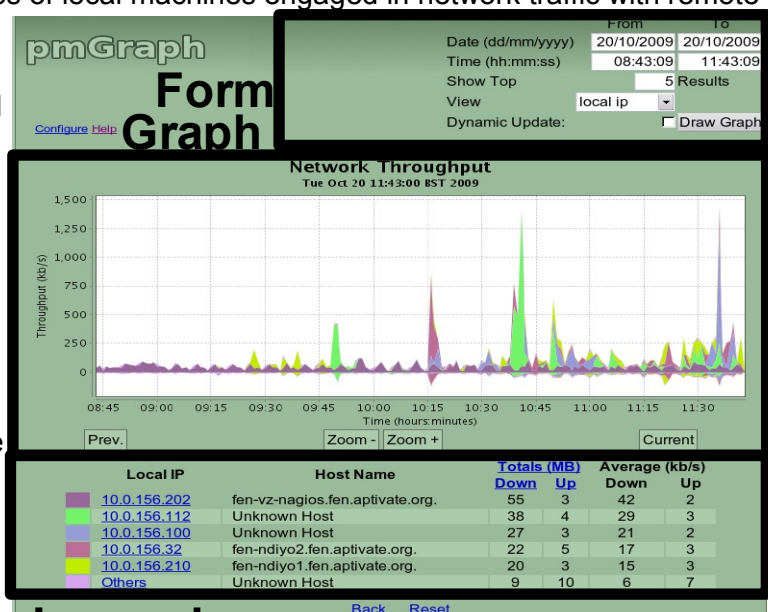
The pmGraph user interface provides an overview of network traffic at a glance. There are three main parts (see picture): a **Form** in the top of the page to select the information required; a **Graph** (middle of the page) showing data relating to the IPs or ports selected, which has a related navigation bar below; and a **legend** (in the bottom of the page) which shows the key for the graph drawn.

The Form

The Form allows various parameters to be set:

- **From date / From time** sets the graph start point
- **To date / To time** sets the graph end point
- **Show Top** Changes the number of results shown on the graph, ordered by significance. Other less significant results are aggregated in "Other"
- **View** Selects from the 4 different possible views:
- **Local port** Shows the ports being used by local machines for network traffic with remote machines
- **Local IP** Shows the IP addresses of local machines engaged in network traffic with remote machines
- **Remote port** Shows the ports on remote machines being used for network traffic with local machines
- **Remote IP** Shows the IP addresses for remote machines engaged in network traffic with local machines
- **Dynamic update** If this box is checked the graph will be automatically refreshed from the database every three minutes.

When a specific IP or port is selected by clicking on it in the legend,



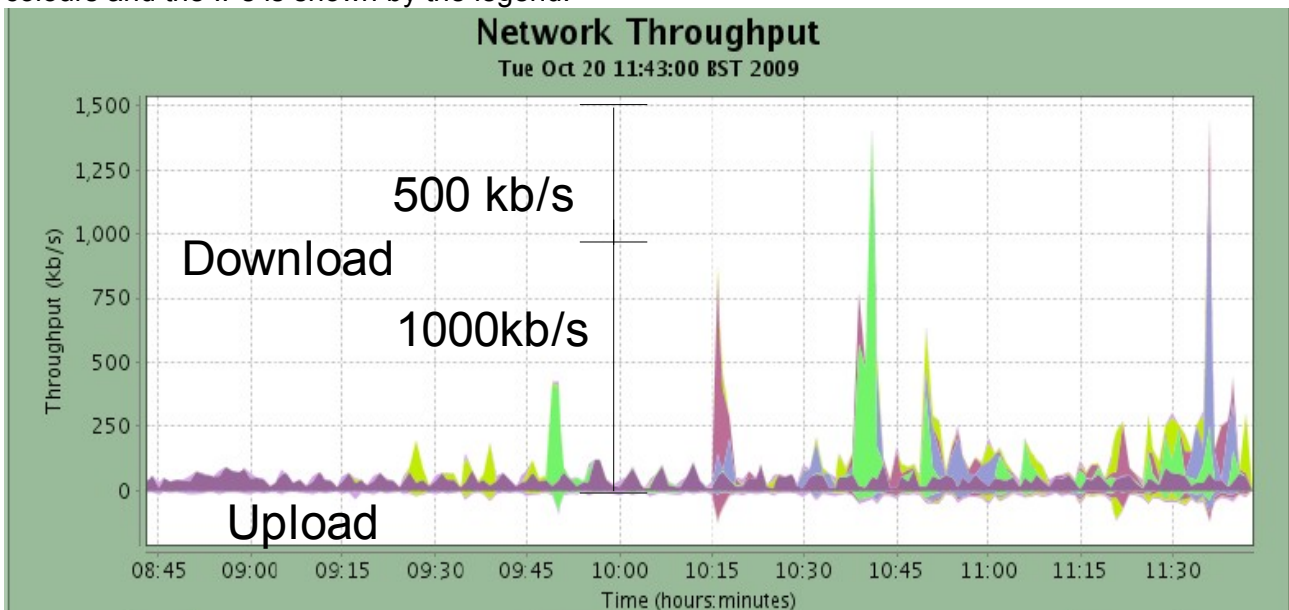
Legend

the form shows the specific port or IP address selected. To show a less specific view either clear this field, or use the Reset link to draw the default graph at the current time. After changing the settings to the desired ones, click "Draw Graph" to redraw the graph.

The graph

The graph generated shows the network traffic rate in kb/s (Y axis) over time (X axis), for the view and time period specified. The X axis (time) increments in units of 1 minute. Downloaded traffic (i.e. traffic from a remote machine to a local machine) is shown in the positive part of the Y axis, and uploaded traffic (traffic from a local machine to a remote machine) is shown in the negative one.

Each colour in the graph represents a different IP or port. The relationship between the colours and the IPs is shown by the legend.



The data is drawn as a stack graph where the values for each data series are added to the data series below it. The colour nearest the X axis is the colour with the highest peak in the time period selected. The graph shows both the traffic rate of various IP addresses or ports over time, and also the total bandwidth usage at any moment in time.

There is also a navigation bar below the graph with four buttons. **Zoom -** and **Zoom +** allow the user to respectively reduce or increase the time period, and **Prev.** and **Next** move the graph backward and forward by 90 minutes. When the graph shows a time period close to the current time, the **Next** button is replaced by **Current**, and clicking on it will give the most recent data.

The legend

The legend shows the relationship between the colours in the graph and the IPs or ports, depending on the selected view. It also shows the total amount of data uploaded and downloaded for each IP or port during the time period selected and the average upload and download throughput for each IP or port in this period. The units (GB, MB, etc) used for the totals amounts are customized to match the time period selected.

You can order the items in the legend, or reverse the current ordering, by clicking on the links of the titles **Uploaded**, **Downloaded** or **Total MB**.

You can obtain a graph of traffic for an individual IP or port by clicking on it in the legend. Once this has been done, an even more specific view can be achieved by clicking again on an individual local or remote port, or remote IP. The graph will then show only traffic involving the selected local IP and that port or remote IP. The title of the graph will always reflect the graph being drawn in each case. To reset to viewing the whole graph, delete the port and IP settings from the Form and click "Draw Graph".

pmGraph will resolve as many IPs as possible to host names in the legend, using the DNS server or a DHCP when it is configured (see [pmGraph configuration file](#)). If pmGraph cannot resolve the hostname, "Unknown Host" is displayed. In the port view you will see the protocol (udp/tcp/icmp) and the services associated with each port. Note: "n/a" is shown in the port column for the icmp protocol since this protocol does not have any port associated.

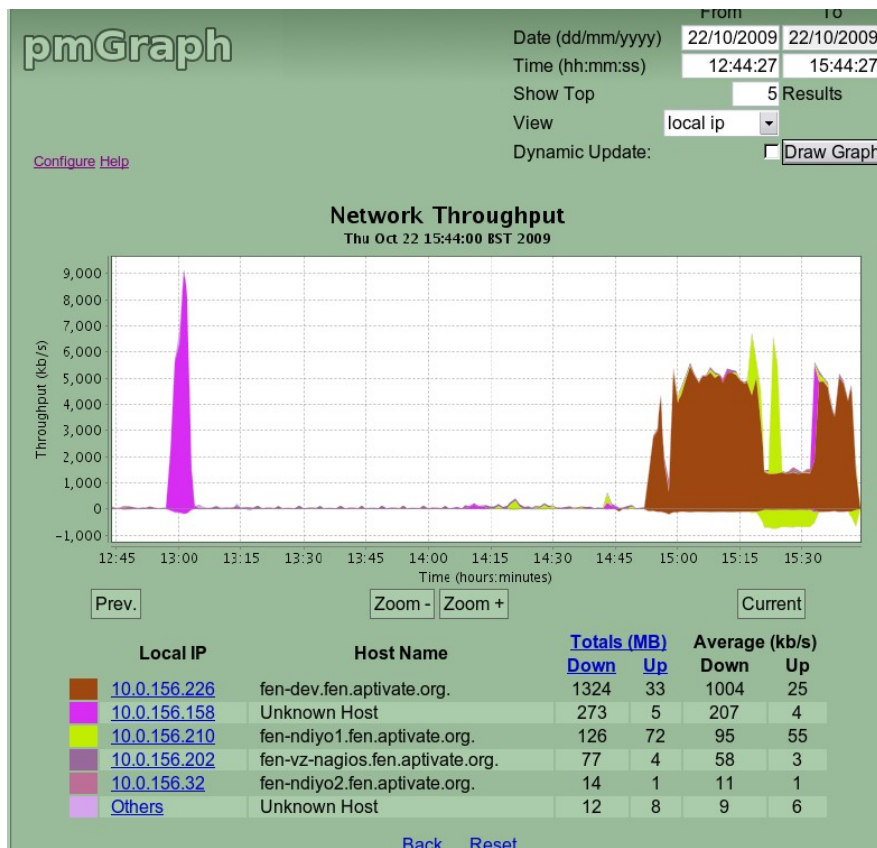
A real example

Let's look at real graph of a network and see how pmGraph can help us to investigate what is happening.

We have installed and configured pmGraph to work in a local network 10.0.156.0/24. Pmacct is running on the router that connects our network to internet and is logging all the traffic between our network (10.0.156.0/24) and any other network.

Step 1

The graph of the current network traffic using the local IP view is shown in the picture on the right.



As you can see in the graph, the IPs which are consuming most bandwidth are:

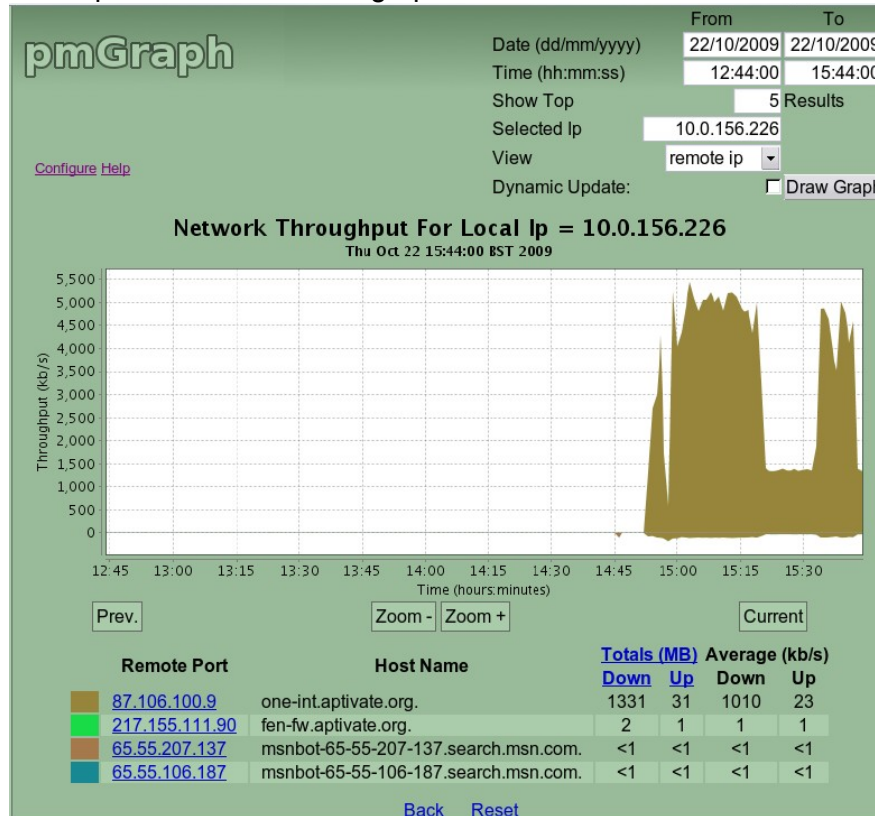
- 10.0.156.226 (fen-dev.fen.aptvate.org): 1324MB downloaded and 33MB uploaded.

- 10.0.156.158 (Unknown host): 273MB downloaded and 5MB uploaded.

We can also see that over the past half hour, both the IP that has had the highest upload and download rates at any one time is 10.0.156.210 (fen-ndiyo1.fen.aptivate.org). The traffic for that IP has gone close to 6,500 kb/s which is quite a bit faster than either of the IP's ranked above it. The first IP is still the highest bandwidth user by quite a long way.

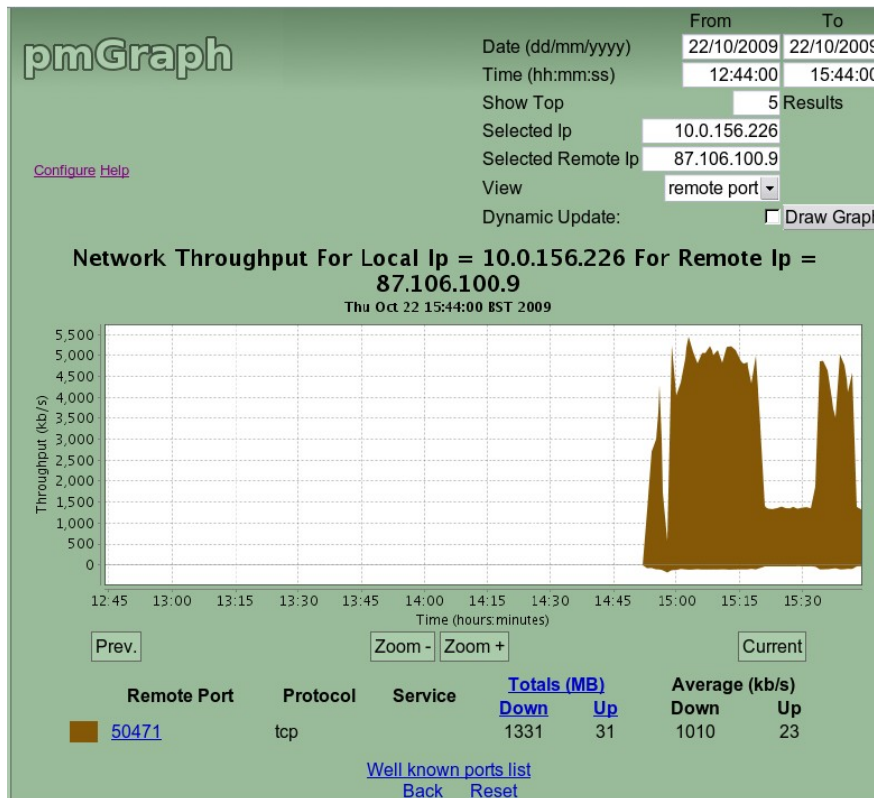
Step 2

We decide to investigate the IP 10.0.156.226, so we click on its link in the legend. By default that shows a graph with the local port view selected but we want to know where this local machine is connecting to, therefore we first select the "Remote IP" view, and then click on "Draw Graph". The result is the graph shown below:



Step 3

This graph shows that most of the traffic for this IP has been with the remote machine "one-int.aptivate.org" which corresponds to the IP 87.106.100.9. Clicking on the IP and selecting the remote port view we see that the connection to the remote machine is on port 50471, using TCP. We conclude that this IP has been downloading from a server called one-int.aptivate.org.



The pmGraph Configuration File

The pmGraph configuration file can be used to set various parameters, e.g. the database connection, local network information and the number of results shown by default. Some parameters may need changing, depending on your circumstances. **It is important to change the database password from the default, for security reasons.**

The pmGraph configuration file is called "database.properties" and can be found in the directory where the war file of pmgraph has been uncompressed by Tomcat. If you have installed the software following our instructions it should be in {CATALINA_HOME}/webapps/pmgraph/WEB-INF/classes/.

The default content of this file is shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment/>
<!-- Default number of IP's shown in a graph -->
<entry key="ResultLimit">5</entry>
<!-- GNU Java DatabaseURL = jdbc:mysql://localhost:3306/pmacct?useJvmCharsetConverters=true -->
<entry key="DatabaseURL">jdbc:mysql://localhost:3306/pmacct</entry>
<entry key="DatabaseUser">pmacct</entry>
<entry key="DatabasePass">secret</entry>
<entry key="JdbcDriver">com.mysql.jdbc.Driver</entry>
<entry key="LocalSubnet">10.0.156.</entry>
<entry key="DatabaseTable">acct_v6</entry>
<entry key="DatabaseLongTable">acct_v6_long</entry>
<!-- This value is the number of hours the graph must cover before pmGraph switches to displaying over a -->
<!-- longer time period. If it is < 24, this feature is disabled. -->
<entry key="TimespansForLongGraph">24</entry>
```

```

<!-- DHCP data in order to connect to it and get hostNames. Info key in base64 -->
<!-- If you haven't got a DHCP server just leave parameters empty to disable -->
<!-- DHCP hostnames resolution -->
<entry key="DHCPPass">yourkey</entry>
<entry key="DHCPPort">7911</entry>
<!-- Name of the secret key as in dhcpd.conf -->
<entry key="DHCPName">omapi_key</entry>
<entry key="DHCPAddress"/>

<!-- Test PostGresql example -->
<!-- entry key="DatabaseURL"--> <!-- jdbc:postgresql://fen-apps/test --><!-- /entry -->
<!-- entry key="DatabaseUser"--> <!-- pmacct2 --><!-- /entry -->
<!-- entry key="DatabasePass"--> <!-- mypassword --><!-- /entry -->
<!-- entry key="JdbcDriver"--> <!-- org.postgresql.Driver --><!-- /entry -->

</properties>

```

Configuration parameters explained

Database connection parameters

- *DatabaseURL*: sql connection string. Specifies the address of your SQL server, the port and the name of the database used by pmacct. The syntax for this parameter is the same as that for JDBC connection strings.

```
jdbc:{databaseType[mysql/psql]}://{serverAddress:serverPort}/{DatabaseName}
```

- *DatabaseUser*: username for connecting to the database.
- *DatabasePass*: password used to connect to the database.
- *JdbcDriver*: the driver used by JDBC to connect to the database. This parameter should be changed if you are using a different database like postgresQL.
- *DatabaseLongTable*: pmacct can be configured to store data in two tables with different sample rates. The DatabaseLongTable parameter contains the name of the table with the lower sample rate.
- *TimespansForLongGraph*: The amount of time, in hours, that the graph will cover before it switches to using data gathered at the lower sample rate. If this is less than 24, the feature is disabled.

Example values to connect to a local mysql server and a database called pmacct:

```

<entry key="DatabaseURL">jdbc:mysql://localhost:3306/pmacct</entry>
<entry key="DatabaseUser">someUser</entry>
<entry key="DatabasePass">somePassword</entry>
<entry key="JdbcDriver">com.mysql.jdbc.Driver</entry>
<entry key="DatabaseTable">acct_v6</entry>
<entry key="DatabaseLongTable">acct_v6_long</entry>
<entry key="TimespansForLongGraph">24</entry>

```

Note

If you have changed these parameters you may want to change the pmacct connection parameters located in the pmacct configuration file `/etc/pmacct/pmacctd.conf`. Simply edit it and modify the parameters *sql_passwd*, *sql_host*, *sql_user*.

Note

Change SQL user password: By the default the installer has set "secret" as the user

password that pmGraph will use to make queries to the database and pmacct will use to put data in the database. **You should change this password for security reasons.** To change the password of the user pmacct in your server you can use the following command:

```
echo "GRANT SELECT,INSERT, LOCK TABLES ON pmacct.* TO 'pmacct'@'localhost'
identified by 'newPassword';" | mysql -u
root -p
```

The command will ask you for the mysql password of the root user.

Using postgresSQL.

You can specify a different Sql server such as PostgreSQL. The configuration below is a example of how to use a PostgreSQL server.

```
<!-- Test PostGresql example -->
<entry key="DatabaseURL">jdbc:postgresql://fen-apps/test</entry>
<entry key="DatabaseUser">pmacct2</entry>
<entry key="DatabasePass"> <mypassword></entry>
<entry key="JdbcDriver">org.postgresql.Driver</entry>
<entry key="DatabaseTable">DatabaseTableName</entry>
<entry key="DatabaseLongTable">DatabaseLongTableName</entry>
```

We have changed the sql connection string, which is the parameter *DatabaseURL*, to connect to a PostgreSQL server and changed the driver class setting it to the class used by JDBC for a PostgreSQL server.

The Local subnetwork parameter

pmGraph is designed to show graphs of the traffic between your local network and other networks. This means that all the internal traffic on your local network will be ignored. This parameter tells pmGraph which IP addresses should be ignored as local. LocalSubnet parameter should be set to match your local network. For example if your local network is 192.168.1.0/24 the value of the parameter should be:

```
<entry key="LocalSubnet">192.168.1.</entry>
```

Note

If you change this parameter, remember to change the pmacct configuration found at `/etc/pmacct/pmacctd.conf`, and change the line:

```
pcap_filter: not (src and dst net 10.0.156.0/24)
```

Make it match your local subnet which for the above example will be:

```
pcap_filter: not (src and dst net 192.168.1.0/24 )
```

The DHCP name resolution parameters

pmGraph tries to resolve the names of the IPs listed in the legend using the DNS but if it doesn't get a name for an IP using DNS it can try lookup on a DHCP server. The

configuration file contains four parameters to configure the connection to a DHCP server:

- *DHCPAddress*: address of the DHCP server in your network.
- *DHCPPort*: port configured to be used by the DHCP server for example 7911.
- *DHCPName*: username for the DHCP server (in a linux DHCP server this is the username in dhcpd.conf).
- *DHCPPass*: user password

Example values for this parameters are:

```
<entry key="DHCPAddress">192.168.1.245 </entry>
<entry key="DHCPPort">7911</entry>
<entry key="DHCPName">omapi_key</entry>
<entry key="DHCPPass">your password</entry>
```

Note

It is possible to disable DHCP hostname resolution if you do not want to use it or if you don't have a DHCP server in your network. To do so simply do not set the value of the parameter *DHCPAddress*, as shown in the default configuration.

The ResultLimit parameter

This is used to set the default number of results to display. It will be used if no value is entered in the *Show Top* field.

The default value of this parameter is:

```
<entry key="ResultLimit">5</entry>
```

Web-based Configuration

The local subnet property in the pmGraph configuration file can be modified via a web interface, shown below. It can be accessed by clicking “Configure” on the page displaying the graph.



From here, edit the Local Subnet value (10.0.156. in the example above) and click “Save

configuration” to update the subnet value or “Back” to cancel.

The pmacct configuration file

The pmacct configuration file is “pmacctd.conf” and is placed in the `/etc/pmacct` directory.

The pmacct configuration file used for only displaying graphs over a short time period is shown below:

```
daemonize: true
pidfile: /var/run/pmacctd.pid
syslog: daemon

plugins: mysql

aggregate: src_host, src_port, dst_host, dst_port, proto
! Just log not local traffic
pcap_filter: not (src and dst net 10.0.156.0/24)

sql_db: pmacct
sql_table: acct_v6
sql_history: 1m
sql_history_roundoff: m
sql_table_version: 6
sql_host: localhost
sql_user: pmacct
sql_passwd: secret
sql_refresh_time: 60
sql_dont_try_update: true
sql_optimize_clauses: true

! Just log traffic higher than 1kb/min. Set this value to a 1% of your connection bandwidth or less.
sql_preprocess: minb = 1000
```

The default one provided displays data for both short and longer time periods, but it is more complicated and will be more fully explained later on.

A few of the pmacct configuration file parameters are explained below. More information about this can be found on the relevant pmacct page at

<http://wiki.pmacct.net/OfficialDocumentation>.

The SQL database connection configuration

The *plugin* parameter defines where pmacct keeps its data. In our example it uses MySQL server and consequently the *plugins* parameter is set to mysql:

```
plugins: mysql
```

It is also possible to use postgresSQL server.

Other parameters used to configure the SQL connection are:

- *sql_db*: name of the database where the data will be stored.
- *sql_table*: name of the table where the data will be stored.
- *sql_host*: machine where the mySQL server is installed.
- *sql_user*: username used to connect to the database.
- *sql_passwd*: password used to connect to the SQL server, associated with the user indicated above.

In the default file which is provided, the parameters are set to:

```
sql_db: pmacct
sql_table[inbound1]: acct_v6
sql_table[outbound1]: acct_v6
sql_table[inbound2]: acct_v6_long
sql_table[outbound2]: acct_v6_long
sql_host: localhost
sql_user: pmacct
sql_passwd: secret
```

Reducing the amount of logged data

There are two parameters that can be used to modify the amount of data logged:

- *pcap_filter*: Establishes which data will be logged. We use this to exclude local traffic. There are other conditions that you can set, which are explained in the pmacct page at <http://wiki.pmacct.net/OfficialDocumentation>
- *sql_preprocess*: Using this parameter you can specify the minimum traffic per minute you want to log.

In the default file, these parameter are set to the following values:

```
pcap_filter: not (src and dst net 10.0.156.0/24) sql_preprocess: minb = 1000
```

Logging data to more than one table.

When viewing graphs covering longer time periods, it can be useful to have data sampled over longer time periods, in order to speed up the creation of the graph and make the graph clearer. To do this, some changes have to be made to the pmacct configuration file. The plugins declaration requires one plugin for each table. For example, for two tables:

```
plugins: mysql
```

becomes

```
plugins: mysql[short], mysql[long]
```

You will need one *aggregate*, *sql_table*, *sql_history*, *sql_history_roundoff* and *sql_refresh_time* parameter for each plugin, which should be labelled in the same way as the plugins. So

```
aggregate: src_host, src_port, dst_host, dst_port, proto
```

would become two parameters like this one:

```
aggregate[short]: src_host, src_port, dst_host, dst_port, proto
```

The parameters starting with “sql” would need to have different values for different pairs. For

```
{sql_table[short]: acct_v6
sql_table[long]: acct_v6_long
```

The table for long data will need to be created, and the query for doing this will look something like the one below (in MySQL), although you might want to change the table name.

```
create table if not exists acct_v6_long ( ip_src CHAR(15) NOT NULL, ip_dst CHAR(15) NOT NULL, src_port INT(2) UNSIGNED NOT NULL, dst_port INT(2) UNSIGNED NOT NULL, ip_proto CHAR(6) NOT NULL, packets INT UNSIGNED NOT NULL, bytes BIGINT UNSIGNED NOT NULL, flows INT UNSIGNED NOT NULL, stamp_inserted DATETIME NOT NULL, stamp_updated DATETIME, PRIMARY KEY (ip_src, ip_dst, src_port, dst_port, ip_proto, stamp_inserted));
```

Where the sql_history times are concerned, the minimum time for long data is one hour so the parameter might look like this:

```
sql_history[long]: 1h
```

The h above means hours, meaning that data will be collected into one hour periods. Sql_history_roundoff takes a time period to round off the data to. When recording for every hour, the data is, by default rounded off to every hour, which is declared as:

```
sql_history_roundoff[long]: h
```

The sql_refresh_time is the number of seconds after which data should be written to a table. This is given in seconds and is, by default the same as the sql_history parameter, so if the sql_history parameter for a plugin is 1h, the sql_refresh_time would be:

```
sql_refresh_time[long]: 3600
```

A modified pmacct configuration file looks something like this:

```
daemonize: true
pidfile: /var/run/pmacctd.pid
syslog: daemon

plugins: mysql[short], mysql[long]

aggregate[short]: src_host, src_port, dst_host, dst_port, proto
aggregate[long]: src_host, src_port, dst_host, dst_port, proto
! Just log not local traffic
! May need to change for local network
pcap_filter: not (src and dst net 10.0.156.0/24)

sql_db: pmacct
sql_table[short]: acct_v6
sql_table[long]: acct_v6_long
sql_history[short]: 1m
sql_history[long]: 1h
sql_history_roundoff[short]: m
sql_history_roundoff[long]: h
sql_table_version: 6
sql_host: localhost
sql_user: pmacct
sql_passwd: apass
sql_refresh_time[short]: 60
sql_refresh_time[long]: 3600
sql_dont_try_update: true
sql_optimize_clauses: true

! Just log traffic higher than 1kb/min Set this value to a 1% of your connection speed
sql_preprocess: minb = 1000
```

Working with pmacct network monitoring software

Usually, it is not possible or desirable to monitor your entire network, as the volume of local traffic can be enormous. Normally it is sufficient to monitor the point where your network connects to your Internet Service Provider (ISP), and record all traffic crossing this point, which is therefore entering or leaving your domain of control (your own network). The interconnection point may be a router, modem, bridge, firewall, switch or access point. Chapter 3 of the Bandwidth Management Book (<http://www.bwmo.net/>) explains network monitoring in more detail.

pmacct, the software which does the actual network logging, provides four options for monitoring the interconnection point between the networks:

Passive Sniffer

You can use your server to monitor passing network traffic in [promiscuous mode](#). Most Unix servers can do this. However, unless you connect the server to a [hub](#) or a monitoring port on a [switch](#), your server will not see packets generated by other computers, and this will limit your monitoring ability.

Unix Router

You can use your server as a [router](#) between two networks. The server will need two network cards, one connected to each network. The server's operating system must support routing, but almost all do. You will also need to know how to configure routing and packet forwarding on your server. You may want or need to use [NAT](#) between the networks (for example, if you replace an existing router that already does NAT for you). If you already have a Unix server or firewall performing this role, then you might find this setup easiest, as you may not need to change your network structure at all.

Unix Bridgenetwork switch

You can use your server as a [bridge](#) between two networks. The server will need two network cards, one connected to each network. The server then acts like a [network switch](#) with two ports, copying packets between the networks without modifying them, and recording them at the same time. The server's operating system must support bridging (e.g. Linux and FreeBSD). You will also need to know how to configure bridging on your server.

Netflow or sFlow

You can use a Cisco or Juniper router to send Netflow or sFlow network accounting records to your Unix server. This removes the need to connect the Unix server directly to the point where the networks meet, as long as the router is already there.

Running pmGraph on a router with OpenWRT software

Setting up pmacct

Once pmacct is installed on the router, type in the following in order to get it to use the pmacctd.conf file in /etc:

```
pmacctd -f/etc/pmacctd.conf
```

Replace the contents of the pmacctd.conf file with one of the examples on the Configuration file section of our website and make the following changes:

```
daemonize: true
```

should be

```
daemonize: false
```

Add the line

```
debug: true
```

Add a line that starts

```
interface:
```

The value given to the above parameter should be the interface that matches the platform that you wish to graph. E.g. eth0.0

If you have 8MB of memory or less you will want to ensure that pmacct only logs to one table and that the TimespansForLongGraph parameter in the pmGraph database.properties file is less than 24. Please see the Configuration file section of for more details. An example configuration file for an OpenWRT router is given below:

```
debug: true
interface: eth0.0
daemonize: false
aggregate: src_host, src_port, dst_host, dst_port, proto
plugins: mysql
pcap_filter: not (src and dst net your.local.subnet.0/24)
imt_buckets: 65537
imt_mem_pools_size: 65536
sql_host: ip.of.graphing.machine.goes.here
sql_db: pmacct
sql_table_version: 6
sql_passwd: mysqlpasswdforpmacctusgoeshere
sql_user: pmacct
sql_table: acct_v6
sql_history: 1m
sql_history_roundoff: m
sql_refresh_time: 60
sql_dont_try_update: true
sql_optimize_clauses: true
sql_preprocess: minb = 1000
```

Setting up MySQL

Since MySQL is probably not going to be running on your router, it will need to be accessible from your network. In order to do this, edit /etc/mysql/my.cnf and comment out

```
bind-address 127.0.0.1
```

by putting a hash in front, and restart MySQL.

```
/etc/init.d/mysql restart
```

Make sure the time is being set correctly on the router where pmacct is running or you may not see your data displayed in pmGraph.

You may also want to ensure that pmacct isn't running on the machine where pmGraph is installed. One way to do this is given below:

```
killall pmacctd  
chmod -x /usr/sbin/pmacctd
```

Disclaimer: We have not tested the instructions on running pmGraph on an OpenWRT router. Thanks to Amish Crusader for providing them. If you think any amendments or corrections are necessary, please [contact us](#).

FAQs

General

Why might I want to use pmGraph?

It can be really hard to understand and manage the network data stored in a database because the data can grow to hundreds of thousands of rows within a few days. To simplify this process we have created pmgraph which allows the user to view the data in a visual form. Additionally you can select a particular IP or port by clicking on it. This allows you to investigate the traffic for a particular IP or port and so better understand your network usage.

On what systems does pmGraph run?

pmGraph is platform independent software. It has been developed in Java and is designed to work in a servlet container. However, pmGraph relies upon pmacct, which runs only on Linux systems, so there must be at least one Linux system in the network.

How can I contribute to pmGraph development?

Contribute as an user: You don't have to be a programmer to contribute to pmGraph. In fact, the most common and valuable way of contributing is through other means:

- 1) Be sure that you report all the bugs you find using the pmGraph sourceforge page at http://sourceforge.net/tracker/?group_id=238574&atid=1106562 .
- 2) If you think pmGraph is missing some features or it can be improved in some way, let us know by sending a email to info@aptivate.org.

3) Help improve pmGraph documentation, for example by making suggestions, or telling us what you found lacking.

4) Contribute as a developer

(<http://www.aptivate.org/Projects.BMOTools.pmGraph.Development.html>):

we are eager to have new people collaborating with us. Please [contact us](#) if you are interested.

What are the system requirements?

pmGraph is quite lightweight, and requires only 8MB of hard disk to run. However, it

relies on other, bulkier programs. If you don't already have Tomcat, Java6, MySQL server, and pmacct, you will need a total of 300 MB to get pmGraph running successfully.

I tried to install pmGraph but it didn't work, what can I do?

Check the [installation steps](#) to make sure you have done everything. If you still have no joy, try the [common installation errors](#).

How can I limit access to pmgraph on my network?

One way is by specifying the IPs allowed to access it, by adding the following lines to the Tomcat file context.xml with your choice of IP address.

```
<Context path="/pmgraph"...>
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
  allow="127.0.0.1" deny=""/>
</Context>
```

I found a bug, where can I report it?

Please tell us about any bug you find. To report them you can use the sourceforge pmgraph project bug tracker at

http://sourceforge.net/tracker/?atid=1106562&group_id=238574&func=browse

My database is growing too fast, can I change the amount of data I am storing?

Yes, using the pmacct config file.

Is there some way to make pmGraph run faster?

Reducing the amount of data you are collecting will change the speed of graph generation.

Third party software

Can I use other databases?

Besides MySQL, you can you use a PostgreSQL database.

Can I use other web servers?

pmGraph is tested with Tomcat 5.5. We are investigating running the program on Jetty which should to reduce the amount of memory needed by the server. If you want to try another server, please do it and let us know how you got on so we can update these instructions [contact us](#).

Can I use other program instead of pmacct?

No, pmGraph is dependent on pmacct.

What other programs can I use that have a smaller memory footprint?

We are currently working on it. The main idea is using another web server (perhaps Jetty) and trying not to use the last version of Java. If you have some ideas, please [contact us](#).

Intellectual properties and rights

What kind of license does pmGraph have?

PmGraph is registered under the GNU General Public License.

<http://www.gnu.org/copyleft/gpl.html>.

Glossary

| Name | Description |
|---|--|
| <u>bridge</u> | Bridging is a forwarding technique used in packet-switched computer networks. Unlike routing, bridging makes no assumptions about where in a network a particular address is located. Instead, it depends on flooding and examination of source addresses in received packet headers to locate unknown devices. Once a device has been located, its location is recorded in a table where the MAC address is stored so as to preclude the need for further broadcasting. |
| <u>hub</u> | A network hub or repeater hub is a device for connecting multiple twisted pair or fiber optic Ethernet devices together and thus making them act as a single network segment. Hubs work at the physical layer (layer 1) of the OSI model. |
| <u>NAT</u> | network address translation (NAT) is the process of modifying network address information in datagram packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another. |
| <u>Netflow</u> | NetFlow is a network protocol developed by Cisco Systems to run on Cisco IOS-enabled equipment for collecting IP traffic information. It's proprietary and supported by platforms other than IOS, such as Juniper routers or FreeBSD and OpenBSD. |
| <u>Network switch</u> | A network switch is a computer networking device that connects network segments. The term commonly refers to a Network bridge that processes and routes data at the Data link layer (layer 2) of the OSI model. Switches that additionally process data at the Network layer (layer 3 and above) are often referred to as Layer 3 switches or Multilayer switches. |
| <u>Promiscuous mode</u> | In computing, promiscuous mode or promisc mode is a configuration of a network card that makes the card pass all traffic it receives to the central processing unit rather than just packets addressed to it. |
| <u>router</u> | Routing is the process of selecting paths in a network along which to send network traffic. Routing is performed for many kinds of networks, including the telephone network, electronic data networks (such as the Internet), and transportation networks. |
| <u>sFlow</u> | The sFlow standard describes a mechanism to capture traffic data in |

switched or routed networks. It uses a sampling technology to collect statistics from the device and is, for this reason, applicable to high speed networks (at gigabit speeds or higher).

Help us to develop pmGraph

pmGraph is under active development by Aptivate staff and volunteers, to make it a more flexible and powerful tool, with advanced user-friendly graphing capabilities. Collaborators on the project are both welcome and important for further development. Please [contact us](#) if you'd like to get involved, or see our more detailed developers' guide (<http://www.aptivate.org/Projects.BMOTools.pmGraph.Development.html>).

Contact Us

If you need more help on any of these instructions, please contact us at info@aptivate.org.