MasterShaper

Easy QoS Traffic Shaping

Documentation

http://shaper.netshadow.at

Version 0.4x

Directory

1 Introduction.... 4
1.1 MasterShaper.... 4
1.2 Requirements.... 4
1.3 IMQ-Devices – What for?.... 5
1.4 Queuing Disciplines – HTB, HFSC or CBQ.... 6
1.5 Support, Ideas & Improvements.... 7
2 Usage.... 8
2.1 Definitions & Terms.... 8
2.1.1 Bandwidth.... 8
2.1.2 Protocols.... 8
2.1.3 Ports.... 8
2.1.4 Targets.... 8
2.1.5 Service Levels.... 9
2.1.6 Filters.... 9
2.1.6.1 layer7 Protocol matching.... 9
2.1.7 Chains.... 10
2.1.8 Pipes.... 10
2.1.9 Bridge or Router.... 10
2.2 Installation.... 12
2.2.1 Package.... 12
2.2.2 Procedure.... 12
2.2.3 Security.... 16
2.3 Statistic collector tc_collector.pl.... 17
2.4 Configuration.... 18
2.4.1 Settings.... 18
2.4.1.1 Options.... 18
2.4.1.2 Users.... 19
2.4.1.3 Targets.... 19
2.4.2 Manage.... 20
2.4.2.1 Service Levels.... 20
2.4.2.2 Filters.... 20
2.4.2.3 Chains.... 21
2.4.2.4 Pipes.... 21
2.4.3 Monitoring.... 22
2.4.3.1 Chains, Pipes and Bandwidth.... 22
2.4.4 Overview.... 23
2.4.5 Rules.... 23
2.4.5.1 Load.... 23
2.4.5.2 Load (debug).... 24
2.4.5.3 Show.... 24
2.4.5.4 Unload.... 24
2.4.6 shaper_loader.sh.... 24
2.5 Tools.... 25
2.5.1 Runlevel-Init-Script.... 25
2.6 Shaping Examples.... 26
2.6.1 Shaping on a Web-Server.... 26
2.6.1.1 Guidelines.... 26
2.6.1.2 Implementation.... 26
2.6.2 Shaping on a gateway.... 27

2.6.2.1 Guidelines....27
2.6.2.2 Implementation....27
2.6.3 Shaping per department....28
2.6.3.1 Guidelines....28
2.6.3.2 Implementation....28
3 License....29

1 Introduction

1.1 MasterShaper

The *MasterShaper* is a web interface for Linux network traffic utilities. It provides an “easy2use” web interface around the *Quality of Service* (QoS) functions available in the Linux 2.4 & 2.6 kernel series. Users can define their own rule sets for handling network flow and also get feedback through graphical statistics about current bandwidth usage and shaping situation.

Mastershaper's goal is to make traffic shaping possible for users who know about networking and the traffic shaping capabilities, but have not much experience with Linux, scripting and other tools needed to do this job.

In the end, the shaping features should be compareable with commercial shaping products like Allot's Netenforcer (http://www.allot.com/html/products_netenforcer.shtm) or Packeteers shaper (http://www.packeteer.com/).

Currently it's only a shaper utility. It's doesn't include a network traffic analyser like the commercial products. It will not display what's going on your network. It will only display the things happens according your rule set.

MasterShaper can be used on a router and also on a transparent bridge.

1.2 Requirements

- Linux kernel version 2.4 or 2.6.x (http://www.kernel.org)
- iproute2 which contains the *tc* command (http://developer.osdl.org/dev/iproute2/)
- IMQ-Devices (If you want to shape inbound traffic, http://www.linuximq.net)
- Web-Server with PHP support (Apache2, mod_php4, http://httpd.apache.org)
- PHP4 with JPEG, libgd and MySQL support (not tested yet with PHP5, http://www.php.net)
- MySQL database (MySQL 4.1, http://www.mysql.com)
- PHP pear modules DB & Net_IPv4
- perl with DBD interface (DBI-MySQL)
- phplayersmenu (sourceforge project, http://phplayersmenu.sourceforge.net)
- jpgraph (http://www.aditus.nu/jpgraph/)
- Web-Browser (with DHTML- and JavaScript-Support, http://www.mozilla.org/products/firefox/)
- sudo

Be aware that the functionality MasterShaper represents is dependent on the availability of features on your system. Some features of the 2.6.x kernels are not back-ported to 2.4.x kernels and so can't be used under 2.4. Also if you have outdated versions of *iptables* or *iproute2* you will discover problems with some features. **Before report problems check if newer versions are available and upgrade first.**

1.3 IMQ-Devices – What for?

Primarily *tc*-utilities are arranged for outbound traffic (*egress*). With outbound traffic you have a lot of options to shape your network traffic. The built-in ingress functions are not so powerful and rudimentary – so you have much less possibilities to control your incoming traffic.

The problem on ingress shaping is, that traffic which arrives on your network card interface, is already on the line and consumes bandwidth. You only have the possibility to drop fast incoming packets or delaying sending acknowledge (ACK) packets and hope that the sender will slow down sending rate (most IP stacks act in this way).

However - in this case you are sitting on the wrong side of the network flow and ingress shaping is less effective then egress shaping. Whatever a supplier of commercial shaping products is telling you - they have all the same problem like the solutions with Linux QoS utilities. The ideal solution is to shape on both sides.

Meanwhile IMQ-Devices – *Intermediate Queueing Device* http://www.linuximq.net - have been arranged to fit this needing. With iptables rules incoming & outgoing traffic will be forward to the queueing devices. The advantage of this - you can use “outgoing traffic rules” on ingress traffic.

To use the IMQ devices you have to patch your kernel and iptables. You will find enough howto's for this in the Internet (use google) – so it will not be explained here.

http://www.linuximq.net/faq.html
http://wiki.nix.hu/cgi-bin/twiki/view/IMQ/ImqFaq

If you want to use IMQ for your external interface (lets say it *ppp0*), then the IMQ usage can be enabled with:

```
ip link set imq0 up
ip link set imq1 up

iptables -t mangle -I PREROUTING -i ppp0 -j IMQ --to-dev 0
iptables -t mangle -I POSTROUTING -o ppp0 -j IMQ --to-dev 1
```

Don't forget – For IMQ usage you have to patch kernel AND iptables! For your kernel you need the options CONFIG_IMQ **AND** CONFIG_IP_NF_TARGET_IMQ**.**

Be aware that you can NOT **use Mastershaper's iptables-matching on IMQ devices! With IMQ device you can only use tc-filter. iptables isn't capable to match packets appearing on a IMQ device.**

1.4 Queuing Disciplines – HTB, HFSC or CBQ

Since V0.30 MasterShaper supports three Queuing Disciplines:

- HTB (*Hierarchical Token Bucket*) http://luxik.cdi.cz/~devik/qos/htb/
- HFSC (*Hierarchical Fair Service Curve*), http://www.cs.cmu.edu/~hzhang/HFSC/main.html
- CBQ (*Class Based Queueing*), http://www.icir.org/floyd/cbq.html

These queuing disciplines have different appendages to do their job. If you want to know more about the theory of these schedulers, please refer to their documentations or home pages in the web. This would blow up this documentation, so only some features will be highlighted here.

HTB is capable of supporting a *guaranteed minimum bandwidth* for a traffic class. Furthermore it let you define the *maximum bandwidth*, which a class can lend from other classes, if the bandwidth is unused. You can define *burst levels* and *priorization* of HTB class. Priorization only affects how much unused bandwidth a class can lend from other classes – higher priorities will gain more bandwidth. If you choose HTB in Mastershaper's settings, it will use HTB for classifying and SFQ (*Stochastic Fairness Queueing*) as final queuing mechanisms when sending packets out to lower layers.

HFSC is capable of supporting a *guaranteed maximum delay* of network packets. This is important for real time applications like "Voice over IP" (VoIP), where delays & jitter have a bad impact on speech quality. Also with HFSC you can define a *minimum guaranteed bandwidth* for each class, and a *maximum bandwidth* which can be used by this class.

CBQ exists much longer then HTB but has less powerful options for traffic control. MasterShaper support CBQ in case HTB is not available.

Vincent Perrier has made some tests HTB versus HFSC. Take a look on his homepage:

http://www.rawsoft.org/example_of_use.html

Mastershaper's default behaviour is HTB. You can change this in the MasterShaper web interface:

Settings → *Options* → *Queuing Discipline*

1.5 Support, Ideas & Improvements

In case you have problems setting up MasterShaper, take a look into the support forum on Mastershaper's homepage if your problems are already known – so please – **use the SEARCH functionalities first before make a new post**:

http://shaper.netshadow.at

If you have ideas or other improvements proposals don't hesitate to post them into the “Feature Request & Inspirations” thread in the support forum.

2 Usage

2.1 Definitions & Terms

The MasterShaper uses some terms to define the shaping rules.

2.1.1 Bandwidth

Bandwidth mean the network speed of your link. MasterShaper always uses speed definitions in kbit per second (kbit/s).

2.1.2 Protocols

You often meet protocols in network environments. In our current time you will often meet IP-Traffic (TCP/UDP) or ICMP-Traffic (ping) – but there are many other protocols like ESP & AH for IPSec, GRE for GRE-Packet-Tunnelling or Router-Protocols like IGMP available.

Each protocol has a unique number which is assigned by IANA:

http://www.iana.org/assignments/protocol-numbers

2.1.3 Ports

Ports reflect the common port numbers for TCP- & UDP-traffic (HTTP/80, IMAP/143,...).

During installation you can instruct MasterShaper Installer to fill the ports table with all ports assigned by IANA:

http://www.iana.org/assignments/port-numbers

2.1.4 Targets

Targets means *IP-Addresses* or *MAC-Addresses*.

IP-Addresses can be specified as single host (1.1.1.1), network address (10.0.0.0/8) or ip-range (1.1.1.1-1.1.1.9).

MAC-Addresses can only be used in iptables mode. Multiple targets can be grouped together as *target groups*.

Think about when you are able to match on MAC-Addresses! You only see MAC-Addresses in your local attached networked. You can't match on MAC-Addresses from machines which are behind routers or in other subnets. This is Ethernet design and has nothing to do with Linux or other system utilities.

2.1.5 Service Levels

Service Levels mean predefined bandwidth limits.

Here you define detailed parameters for *HTB, HFSC* or *CBQ* queuing discipline. For CBQ you can specify rate and priority. In addition you can define ceil and burst with HTB for incoming and outgoing traffic (asymmetric lines). For HFSC it's possible to specify the maximum delay of network packets.

2.1.6 Filters

Filters represent methods to match your traffic against defined rules. For example you can define, that a filter “Web-Traffic” match the HTTP- & HTTPS-ports 80/tcp & 443/tcp. Furthermore you can match on TOS-Flags, TCP-Flags, IPP2P, layer7, Time, packet length, ...

The availability of filter-functions depends on which matching-system you use. MasterShaper supports *tc-filter* and *iptables*. While tc-filter is fast and already integrated in the iproute2 package, iptables is a additional subsystem which supports many fancy match-methods. If you don't need the features iptables offers simply rely on tc-filter.

To find out, if your iptables installation supports all MasterShaper features, check out if the necessary match-modules are available in the iptables modules directory (usually */lib/iptables*)

*) *libipt_TOS.so*	for TOS matching
*) *libipt_tcp.so*	for TCP-Flag matching
*) *libipt_ipp2p.so*	for IP-P2P matching (http://www.ipp2p.org)
*) *libipt_time.so*	for time matching
*) *libipt_length.so*	for packet length matching
*) *libipt_layer7.so*	for layer7 protocol matching (http://l7-filter.sf.net)
*) *libipt_helper.so*	for ftp data channel matching
*) *libipt_conntrack.so*	for ftp data channel matching

MasterShaper isn't currently checking if all these modules are available. If you get some errors when loading the iptables-matching ruleset, check if all modules are in place first!

2.1.6.1 layer7 Protocol matching

With V0.32 layer7 protocol matching support (http://l7-filter.sf.net) has been integrated into MasterShaper.

With *Option* → *Update L7 Protocols* MasterShaper will get the available protocol match names (*.pat files* in */etc/l7-protocols*) and save them in the database. If you update your l7-filter installation you have to run this update process in MasterShaper again to get new supported protocols into MasterShaper configuration.

2.1.7 Chains

Chains are building traffic channels. Each chain has an assigned service level – the maximum available bandwidth within this channel. If you have only one chain, this service level is normally equal to your line speed (*2048/1024kbit/s* for example).

In addition each chain has a fall-back service level - any traffic, which is not matched through a pipe definition can only use the bandwidth of the fall-back service level. So MasterShaper makes sure, that no *unknown* traffic can't eat up your whole bandwidth.

To get the traffic into the chains, the network traffic will be matched by target definitions. The order of the chain rules are important – **the first match win, not that one which is the exactest.**

So if you have two chains with the following targets (in this order):

- *192.168.1.0/24*
- *192.168.1.1*

traffic to/from *192.168.1.1* will be matched by the chain with the *192.168.1.0/24* target and not by the chain with the *192.168.1.1* target.

If you won't specify IP addresses for targets, you can also use the “any” entry in the chain setup.

It's also possible to define a chain which completely ignores the QoS settings. This is sometimes useful if you have traffic which should not be touched by any shaper settings (LAN ⇔ DMZ). Chains which are ignoring QoS setting are not recorded through tc_collector.pl and aren't shown in monitoring graphs.

2.1.8 Pipes

Pipes bring chains, filters and service levels together. In addition you can specify the direction of the pipes (incoming, outgoing). Here you also assign a service level, which regulate the bandwidth usage of this pipe.

The current bandwidth distribution between pipes can be displayed over *Monitoring* → *Pipes*.

2.1.9 Bridge or Router

A *bridge* is a transparent network device. For example – normally you have connected your main router (Cisco, Nortel, ...) directly to your network switch. Now you connect the router on the first interface of the bridge. The second interface of the bridge is connected to your network switch. The bridge acts totally invisible for any connections between the router and your network. But you are capable to affect the network flow on both interfaces of the bridge. More informations about setting up a Linux bridge can be found here:

http://bridge.sourceforge.net/

A *router* connects two different networks together (like 192.168.191.0/24 and 172.16.2.0/24). None of the clients in different subnets know about any other clients on the other networks. They only know how to send packets to other networks (via default gateway, route, ...). The router knows - according his routing table - where to send these packets.

Packet handling – exactly matching the network interfaces – is a bit different between routers & bridges so you have to tell MasterShaper in which mode it has to act.

2.2 Installation

2.2.1 Package

The MasterShaper Installer consists the following files & directories:

INSTALL	...	*install notes*
README	...	*non relevant ReadMe*
LICENSE	...	*relevant GPL2 license*
UPGRADE	...	*some upgrade informations*
docs	...	*documentation in OpenDocument format*
htdocs	...	*document root, php files, perl files,...*
tools	...	*runlevel init script,...*

2.2.2 Procedure

Some steps in the installation procedure need knowledge of some basic MySQL commands & actions. If you are not familiar with MySQL, consider some helpful tools like phpMyAdmin (http://www.phpmyadmin.net/) to get the database ready for MasterShaper.

- **Create a new MySQL database**

This command will create the example database *db_shaper:*

```
create database db_shaper;
```

It's a good idea to create a new database user to access this database and don't make the database connect with the MySQL-root user – checkout the MySQL documentation how to add additional MySQL users (GRANT).

- **Extract the MasterShaper package**

Extract the *mastershaper_x.xx.tar.gz* file which you had downloaded from MasterShaper website to some temporary directory:

```
mkdir /tmp/shaper
cd /tmp/shaper
tar zxfv (PATH_WHERE_FILE_IS_LOCATED)/mastershaper_x.xx.tar.gz
```

- **Move MasterShaper into webservers document root**

Move the content of the *htdocs*-directory (PHP scripts, images, ...) as it is below your document root of your web server (like */var/www/shaper*) so MasterShaper is reachable via browser:

http://server/shaper/

- **Install jpgraph**

  Download jpgraph from http://www.aditus.nu/jpgraph/ into the MasterShaper directory. Extract the tar.gz file and make a symbolic link from “jpgraph-x.xx” directory to “jpgraph” (you could also rename jpgraph-x.xx to jpgraph).

- **Install phplayersmenu**

  Download phplayersmenu from http://phplayersmenu.sourceforge.net/ into the MasterShaper directory. Extract and make a symbolic link from “phplayersmenu-x.x.x” to “phplayersmenu” (you could also rename phplayersmenu-x.xx to phplayersmenu).

- **Install PHP-Pear Modules**

  If the necessary PHP-PEAR modules aren't installed, do the following

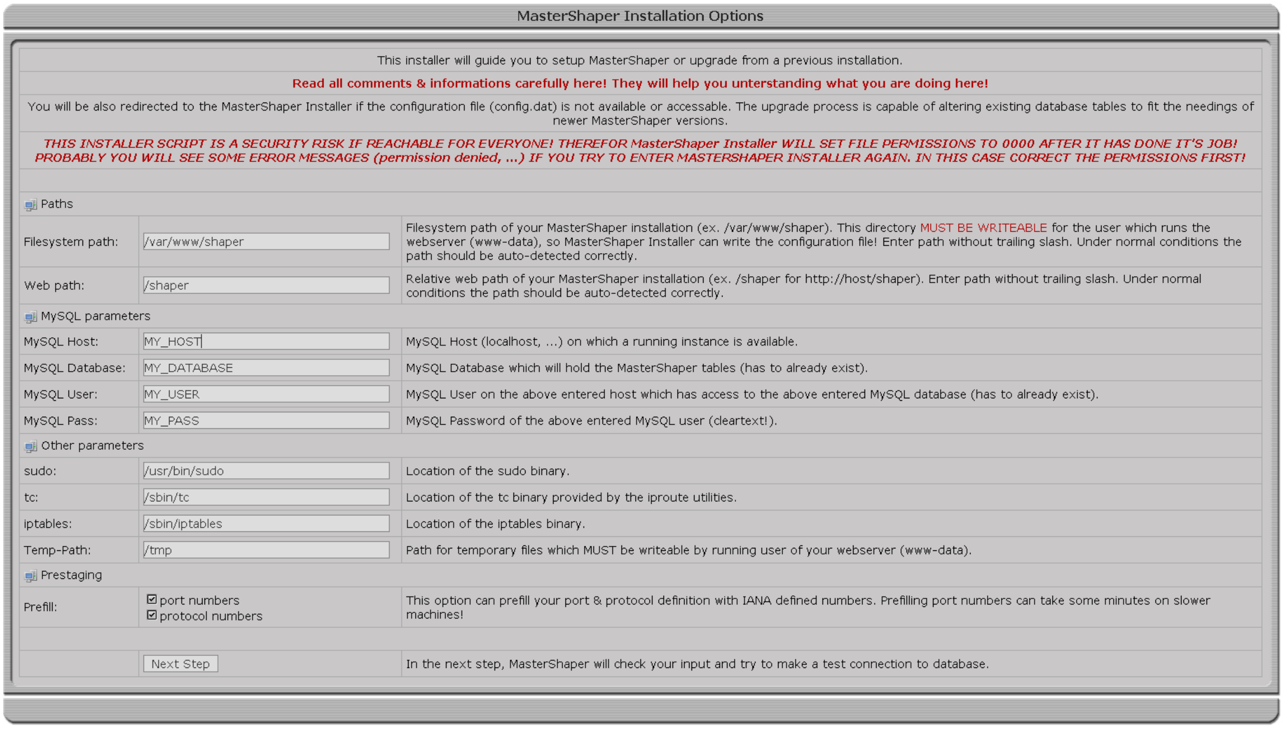
  ```
  pear install DB Net_IPv4
  ```

  to install them.

- **MasterShaper Installer**

  Open a browser and enter the URL for MasterShaper (ex. http://server/shaper/).

  It will automatically forwarded you to the *MasterShaper Installer*. Specify the parameters as shown up in the Installer and click through the installation steps. In case of troubles, the Installer should tell you were the problems are.



MasterShaper Installation Options

This installer will guide you to setup MasterShaper or upgrade from a previous installation.

Read all comments & informations carefully here! They will help you unterstanding what you are doing here!

You will be also redirected to the MasterShaper Installer if the configuration file (config.dat) is not available or accessable. The upgrade process is capable of altering existing database tables to fit the needings of newer MasterShaper versions.

THIS INSTALLER SCRIPT IS A SECURITY RISK IF REACHABLE FOR EVERYONE! THEREFOR MasterShaper Installer WILL SET FILE PERMISSIONS TO 0000 AFTER IT HAS DONE IT'S JOB! PROBABLY YOU WILL SEE SOME ERROR MESSAGES (permission denied, ...) IF YOU TRY TO ENTER MASTERSHAPER INSTALLER AGAIN. IN THIS CASE CORRECT THE PERMISSIONS FIRST!

Paths		
Filesystem path:	/var/www/shaper	Filesystem path of your MasterShaper installation (ex. /var/www/shaper). This directory MUST BE WRITEABLE for the user which runs the webserver (www-data), so MasterShaper Installer can write the configuration file! Enter path without trailing slash. Under normal conditions the path should be auto-detected correctly.
Web path:	/shaper	Relative web path of your MasterShaper installation (ex. /shaper for http://host/shaper). Enter path without trailing slash. Under normal conditions the path should be auto-detected correctly.
MySQL parameters		
MySQL Host:	MY_HOST	MySQL Host (localhost, ...) on which a running instance is available.
MySQL Database:	MY_DATABASE	MySQL Database which will hold the MasterShaper tables (has to already exist).
MySQL User:	MY_USER	MySQL User on the above entered host which has access to the above entered MySQL database (has to already exist).
MySQL Pass:	MY_PASS	MySQL Password of the above entered MySQL user (cleartext!).
Other parameters		
sudo:	/usr/bin/sudo	Location of the sudo binary.
tc:	/sbin/tc	Location of the tc binary provided by the iproute utilities.
iptables:	/sbin/iptables	Location of the iptables binary.
Temp-Path:	/tmp	Path for temporary files which MUST be writeable by running user of your webserver (www-data).
Prestaging		
Prefill:	☑ port numbers ☑ protocol numbers	This option can prefill your port & protocol definition with IANA defined numbers. Prefilling port numbers can take some minutes on slower machines!
	Next Step	In the next step, MasterShaper will check your input and try to make a test connection to database.

- **Prepare IMQ**

If you are using IMQ you need some iptables rules to get the traffic to the IMQ interfaces (EXT_DEV means your interface where you want to shape traffic):

```
ip link set imq0 up
ip link set imq1 up

iptables -t mangle -I PREROUTING -i ${EXT_DEV} -j IMQ --todev 0
iptables -t mangle -I POSTROUTING -o ${EXT_DEV} -j IMQ --todev 1
```

This rules aren't set by MasterShaper. So make sure that they are available when you start shaping.

- **sudo Configuration**

Mostly your web server isn't running with *root privileges* and so it hasn't permissions to talk with the kernel and load the ruleset (*tc* or *iptables*). To activate the rules MasterShaper uses the scripts *shaper_loader.sh*, a little shell script, which handles both *tc*- and also *iptables-rules*. To gain this script root privileges open the *sudo config file*

```
/etc/sudoers
```

and add the line

```
USER        ALL= NOPASSWD: PATH/shaper_loader.sh
```

where *USER* is the running user of your web server (*www-data, apache, ...*) and *PATH* is the full (absolute!) file system path to your MasterShaper installation (*/var/www/shaper*).

Don't forget that your system must meet some requirements for traffic shaping with MasterShaper. Take a look at 1.2 Requirements.

2.2.3 Security

The database connection parameters are stored in the configuration files

config.dat

in the MasterShaper's web path directory (ex. /var/www/shaper/config.dat).

This is a critical file – plain text passwords - access to this File need to be secured!

The MasterShaper Installation Package includes an *.htaccess* file in the htdocs directory, which limits the access to the *config.dat* file.

Double check if this file is in its location and if your web server is configured probably, to support *.htaccess*. If not referrer to your web server documentation how to limit access to a file in the webserver configuration (<*FILE*>).

Make sure, that it's not possible to download this file via web browser:

http://server/shaper/config.dat

Every time *MasterShaper Installer* finish it's job it tries do limit the access to the *index.php* file in the *setup* directory. If you see an error message, MasterShaper can't change file permissions, so please take care that the *MasterShaper Installer* is not public available from the Internet (**passwords in plain text**!).

2.3 Statistic collector tc_collector.pl

tc_collector.pl is a little Perl application which collects traffic statistics from the tc utility.

Cause there is no usable mechanism to get the current pipes distributions, it collects the total amount of bytes transferred within 10 seconds intervals and calculate a kilobits per second average from this.

Run the tc_collector by calling:

```
./tc_collector.pl
```

It will start collecting transfer rates from the tc binary and record them into MySQL database. It will get it's configuration also from *config.dat* – no adaptation to the Perl t is needed.

If you call it with

```
tc_collector.pl -d
```

it will fork into background and run daemonized.

If you are expecting troubles with *tc_collector.pl*, open the Perl script within your favourite editor and set the variable “$debug” to 1. If you start tc_collector.pl in foreground it will output what it's currently collecting.

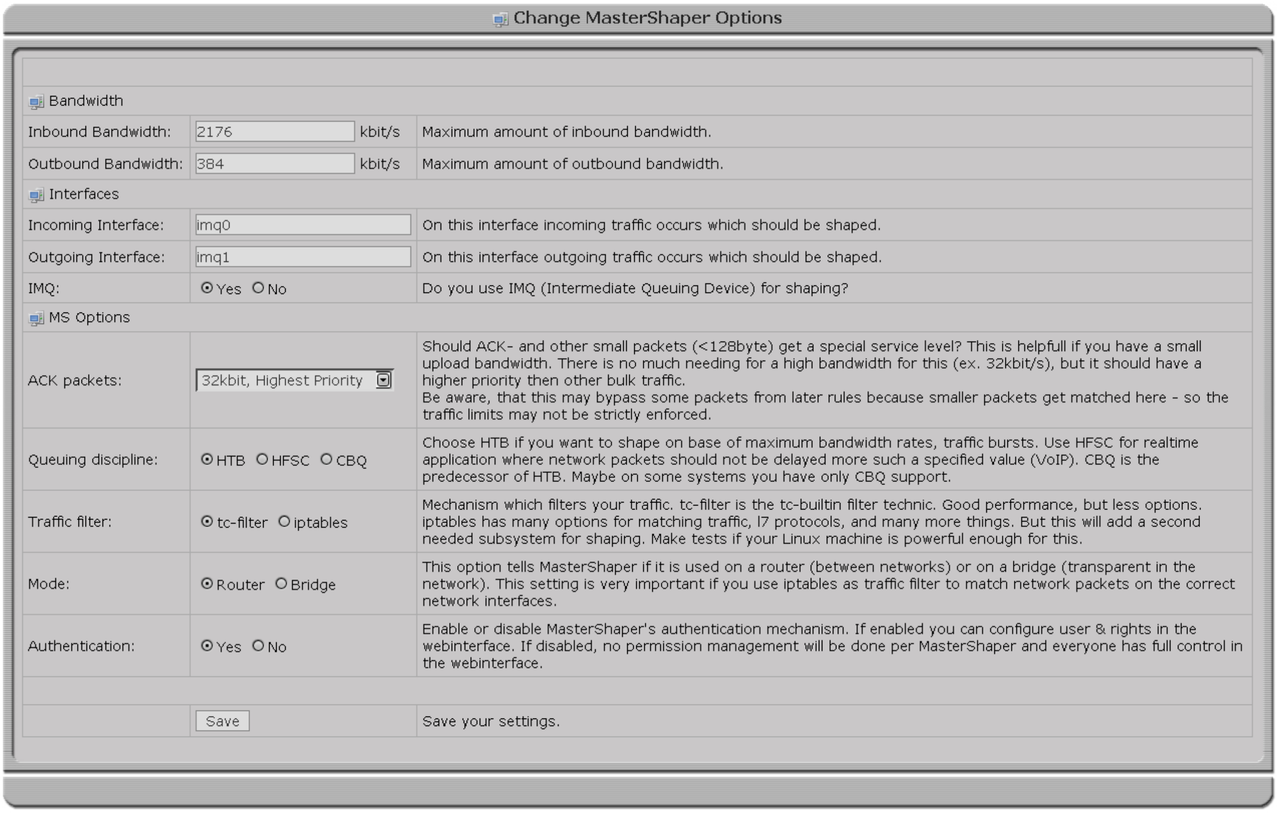
Without tc_collector.pl you will get no monitoring statistics and the graphs in the web interface will not work!

2.4 Configuration

Mostly all options are plenty documented within the web interface. So this document only include short summaries.

2.4.1 Settings

2.4.1.1 Options



Change MasterShaper Options

Bandwidth		
Inbound Bandwidth:	2176 kbit/s	Maximum amount of inbound bandwidth.
Outbound Bandwidth:	384 kbit/s	Maximum amount of outbound bandwidth.
Interfaces		
Incoming Interface:	imq0	On this interface incoming traffic occurs which should be shaped.
Outgoing Interface:	imq1	On this interface outgoing traffic occurs which should be shaped.
IMQ:	● Yes ○ No	Do you use IMQ (Intermediate Queuing Device) for shaping?
MS Options		
ACK packets:	32kbit, Highest Priority	Should ACK- and other small packets (<128byte) get a special service level? This is helpfull if you have a small upload bandwidth. There is no much needing for a high bandwidth for this (ex. 32kbit/s), but it should have a higher priority then other bulk traffic. Be aware, that this may bypass some packets from later rules because smaller packets get matched here - so the traffic limits may not be strictly enforced.
Queuing discipline:	● HTB ○ HFSC ○ CBQ	Choose HTB if you want to shape on base of maximum bandwidth rates, traffic bursts. Use HFSC for realtime application where network packets should not be delayed more such a specified value (VoIP). CBQ is the predecessor of HTB. Maybe on some systems you have only CBQ support.
Traffic filter:	● tc-filter ○ iptables	Mechanism which filters your traffic. tc-filter is the tc-builtin filter technic. Good performance, but less options. iptables has many options for matching traffic, l7 protocols, and many more things. But this will add a second needed subsystem for shaping. Make tests if your Linux machine is powerful enough for this.
Mode:	● Router ○ Bridge	This option tells MasterShaper if it is used on a router (between networks) or on a bridge (transparent in the network). This setting is very important if you use iptables as traffic filter to match network packets on the correct network interfaces.
Authentication:	● Yes ○ No	Enable or disable MasterShaper's authentication mechanism. If enabled you can configure user & rights in the webinterface. If disabled, no permission management will be done per MasterShaper and everyone has full control in the webinterface.
	Save	Save your settings.

In this view you configure:

- *Bandwidth*

This bandwidth is essential for the init class and should be as high as the maximum speed of the specified interfaces (Ethernet, DSL, ...).

- *Interfaces*

Also you need to specify the incoming and outgoing interfaces on which the shaping will happen. Either configured as router or bridge you enter here the physical interfaces of your shaping device. If you are using IMQ you have to tell MasterShaper via the *IMQ-Option*.

- *MS Options*

Further more you can define a special preferred handling of *ACK packets* and other *small packets*. You have to create a service level first to handle this packets.

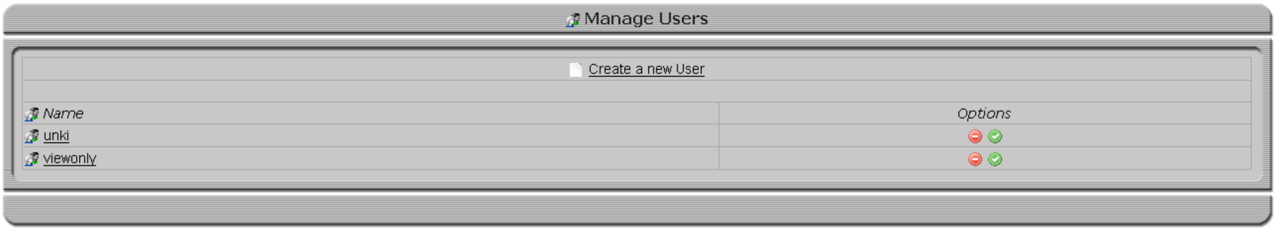
Configure a *queueing discipline* that fit your needing. More informations about queuing disciplines can be found in the chapter 1.4.Queuing Disciplines – HTB, HFSC or CBQ.

Choose between *tc-filter* & or *iptables-matching*. tc-filter is included in the iproute2 package and very fast. iptables on the other side is widely used, many additional modules (ipp2p, layer-7, ...) are available and is very stable. iptables consumes a bit more additional cpu & memory for matching packets. If you don't need the features of iptables simple rely on tc-filter.

You have to tell MasterShaper if it is running on a *router* or *bridge*. This setting is very important if you are using iptables-matching because MasterShaper has to adept the iptables rules in bridge mode to match the physically interfaces (*physdev*) of a bridge.

New since V0.40 is an integrated *User-Management*. To use this feature you have to activate the authentication mechanism. It's now possible to gain users access to selective functions of MasterShaper. It's possible now to create only a user which has access to the monitoring graphs but can not change any settings. In the next versions a finer granulation of permissions will be implemented, so users can have the permissions to change pipes & filters settings within their own chains.

2.4.1.2 Users



Manage Users

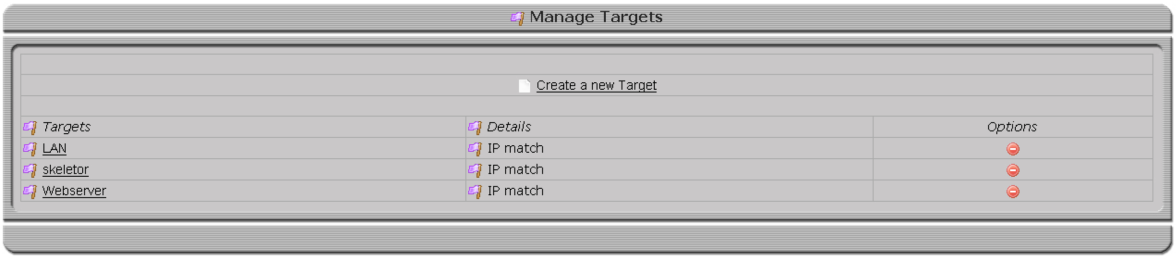
Create a new User

Name	Options
unki	⊖ ✓
viewonly	⊖ ✓

If you have a fresh MasterShaper installation the initial user is “*admin*” and password “*changeme*”.

If you have upgraded from a previous version and activated the authentication before created a new user, you will be locked out of MasterShaper because there is no user available to login. In this case delete the option “authentication” from the MySQL table *shaper_settings* via SQL commands or via some GUI's (phpMyAdmin, http://www.phpmyadmin.net).

2.4.1.3 Targets



Manage Targets

Create a new Target

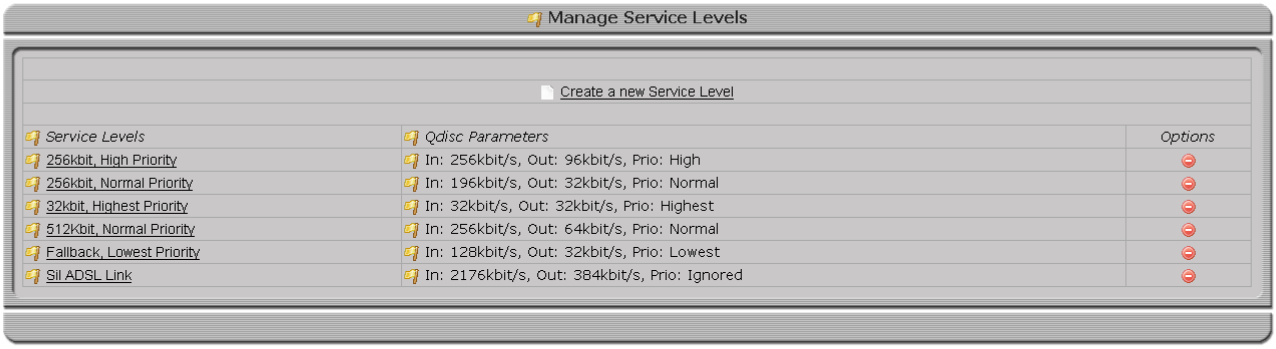
Targets	Details	Options
LAN	IP match	⊖
skeletor	IP match	⊖
Webserver	IP match	⊖

If you want to shape traffic for specific *IP addresses* or *MAC addresses*, you define them here. These definitions will then be used in the chains setup. Several target-definitions can be grouped together to a *target-group* for easier usage in chains.

If you have a *dynamic* external IP address, you don't need to specify anything here and use "any ⇔ any" in the chain setup.

2.4.2 Manage

2.4.2.1 Service Levels



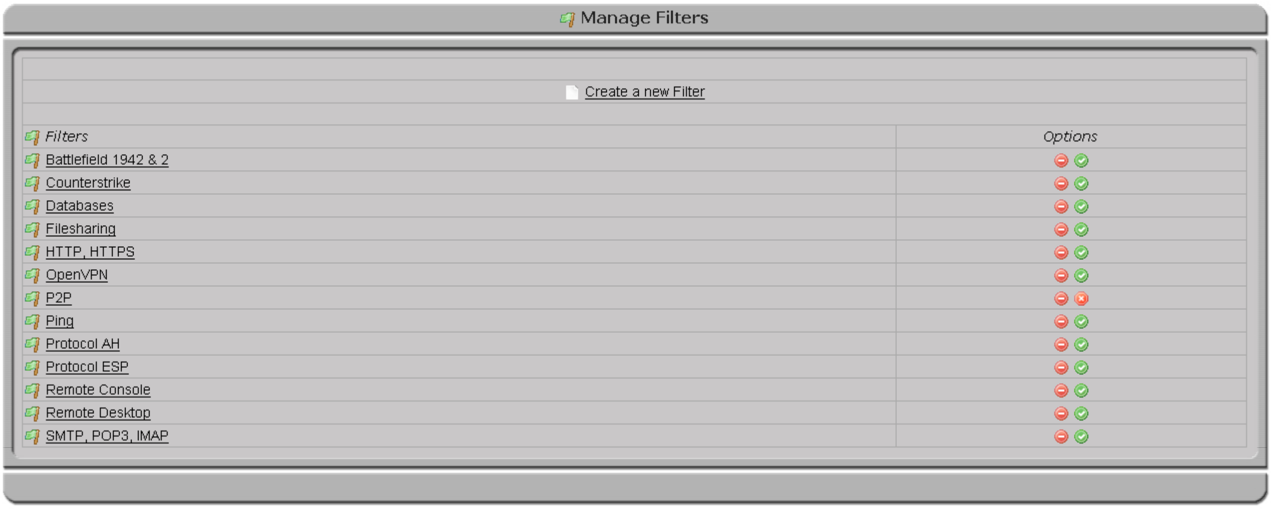
Manage Service Levels

Create a new Service Level

Service Levels	Qdisc Parameters	Options
256kbit, High Priority	In: 256kbit/s, Out: 96kbit/s, Prio: High	⊖
256kbit, Normal Priority	In: 196kbit/s, Out: 32kbit/s, Prio: Normal	⊖
32kbit, Highest Priority	In: 32kbit/s, Out: 32kbit/s, Prio: Highest	⊖
512Kbit, Normal Priority	In: 256kbit/s, Out: 64kbit/s, Prio: Normal	⊖
Fallback, Lowest Priority	In: 128kbit/s, Out: 32kbit/s, Prio: Lowest	⊖
Sil ADSL Link	In: 2176kbit/s, Out: 384kbit/s, Prio: Ignored	⊖

Here you specify service levels. Service Levels are used in Chains, Pipes and in Options-View. Each service level has a separate definition for HTB- , HFSC- and CBQ-parameters (queuing disciplines).

2.4.2.2 Filters



Manage Filters

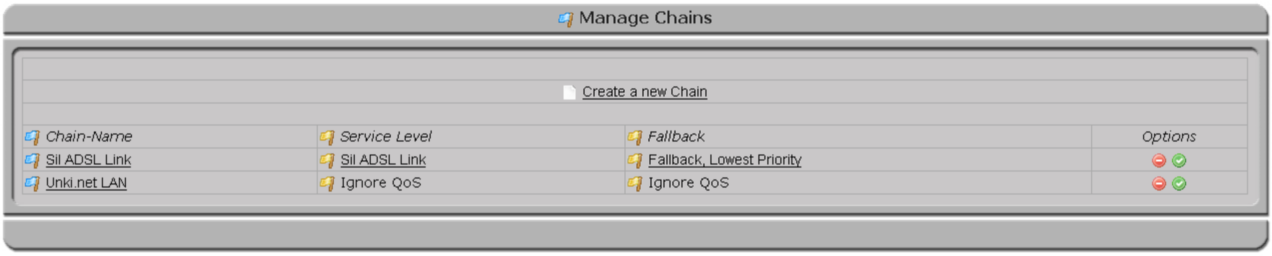
Create a new Filter

Filters	Options
Battlefield 1942 & 2	⊖ ✓
Counterstrike	⊖ ✓
Databases	⊖ ✓
Filesharing	⊖ ✓
HTTP, HTTPS	⊖ ✓
OpenVPN	⊖ ✓
P2P	⊖ ✗
Ping	⊖ ✓
Protocol AH	⊖ ✓
Protocol ESP	⊖ ✓
Remote Console	⊖ ✓
Remote Desktop	⊖ ✓
SMTP, POP3, IMAP	⊖ ✓

In this view you manage your filter definitions. Filters are traffic match mechanisms which classifies your traffic so it get divided up into the correct pipes.

Which sort of filters you create here is dependent on the "*Traffic filter*" option.

2.4.2.3 Chains



Manage Chains

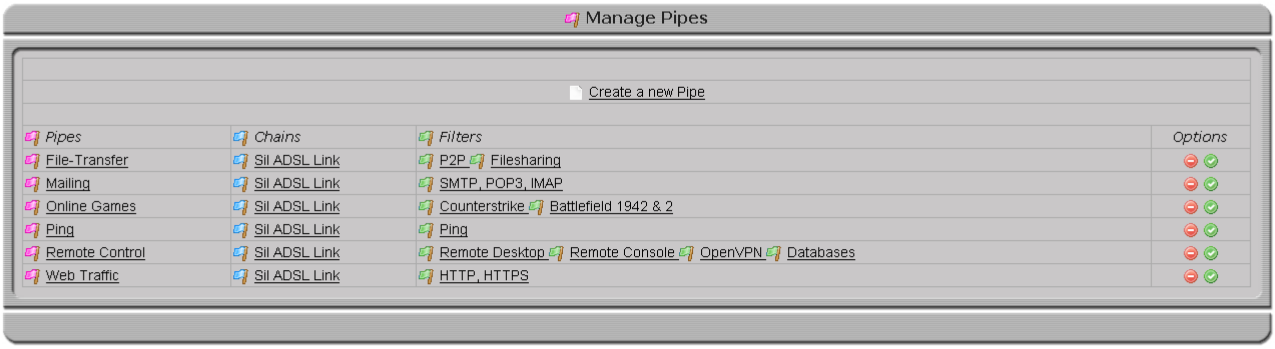
Create a new Chain

Chain-Name	Service Level	Fallback	Options
Sil ADSL Link	Sil ADSL Link	Fallback, Lowest Priority	⊖ ✓
Unki.net LAN	Ignore QoS	Ignore QoS	⊖ ✓

Here you manage your chain rulesets. Chains are necessary to match the traffic against targets. If the target definition match your network traffic, the network flow will be redirected into this chain so it can be matched by the following pipe definitions.

A chain needs to get defined a total amount of bandwidth and a fall-back service level. Any traffic which comes into this chain and don't get matched by any pipe definitions will fall into the fall-back service level.

2.4.2.4 Pipes



Manage Pipes

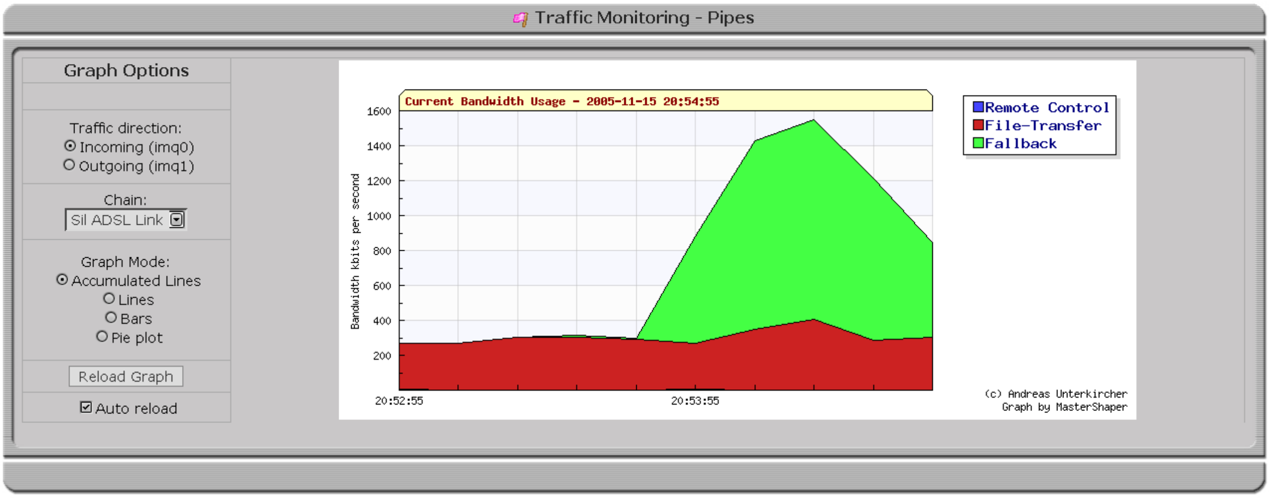
Create a new Pipe

Pipes	Chains	Filters	Options
File-Transfer	Sil ADSL Link	P2P Filesharing	⊖ ✓
Mailing	Sil ADSL Link	SMTP, POP3, IMAP	⊖ ✓
Online Games	Sil ADSL Link	Counterstrike Battlefield 1942 & 2	⊖ ✓
Ping	Sil ADSL Link	Ping	⊖ ✓
Remote Control	Sil ADSL Link	Remote Desktop Remote Console OpenVPN Databases	⊖ ✓
Web Traffic	Sil ADSL Link	HTTP, HTTPS	⊖ ✓

Pipes are assigned to chains and match filter-definitions against the network traffic which virtually flows through this chains. Pipes also manage how much bandwidth a service (matched by filters) can really consume.

2.4.3 Monitoring

2.4.3.1 Chains, Pipes and Bandwidth



Traffic Monitoring - Pipes

Graph Options

Traffic direction:
● Incoming (imq0)
○ Outgoing (imq1)

Chain:
Sil ADSL Link

Graph Mode:
● Accumulated Lines
○ Lines
○ Bars
○ Pie plot

Reload Graph

☑ Auto reload

If Mastershaper's rules are loaded correctly and tc_collector.pl is active MasterShaper will draw fancy graphs:

- Chains

This view will show you the current bandwidth distribution between chains.

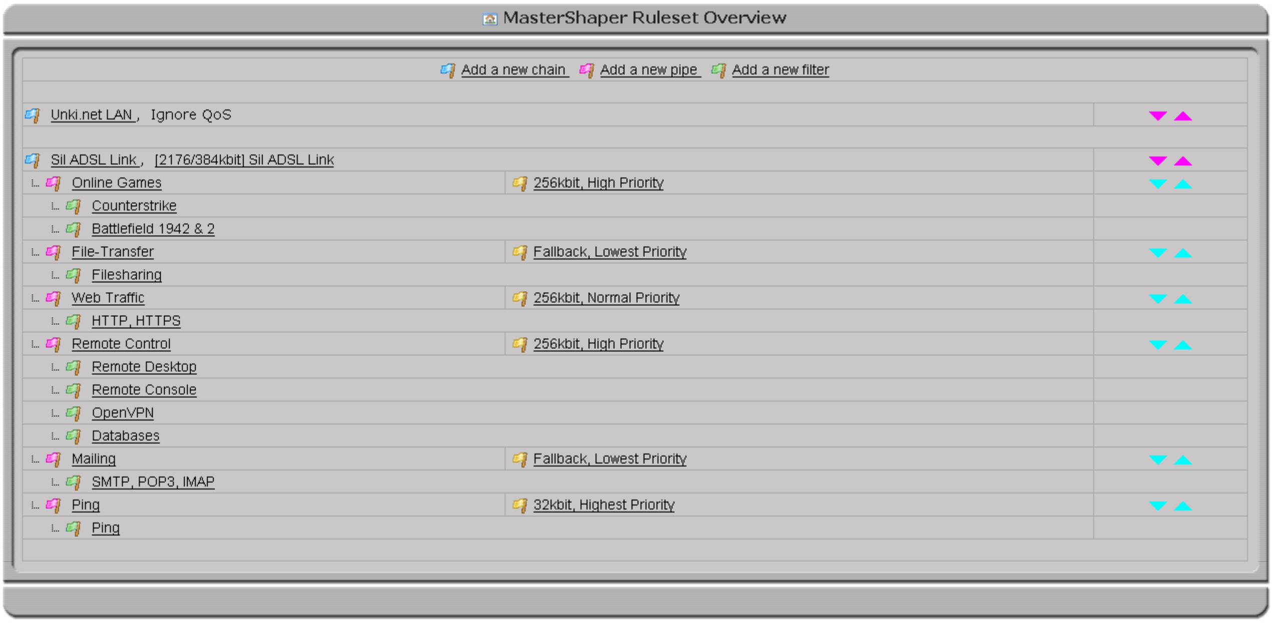
- Pipes

This view will show you the current bandwidth distribution of pipes. Also available is a dropdown box where you can switch between chains.

- Bandwidth

This view will present the total inbound and outbound bandwidth.

2.4.4 Overview



MasterShaper Ruleset Overview

Add a new chain Add a new pipe Add a new filter

Unki.net LAN , Ignore QoS		▼▲
Sil ADSL Link , [2176/384kbit] Sil ADSL Link		▼▲
└ Online Games	256kbit, High Priority	▼▲
└ Counterstrike		
└ Battlefield 1942 & 2		
└ File-Transfer	Fallback, Lowest Priority	▼▲
└ Filesharing		
└ Web Traffic	256kbit, Normal Priority	▼▲
└ HTTP, HTTPS		
└ Remote Control	256kbit, High Priority	▼▲
└ Remote Desktop		
└ Remote Console		
└ OpenVPN		
└ Databases		
└ Mailing	Fallback, Lowest Priority	▼▲
└ SMTP, POP3, IMAP		
└ Ping	32kbit, Highest Priority	▼▲
└ Ping		

These view presents a good overview through your rule sets.

Disabled chain, pipes or filter definitions are not shown up here.

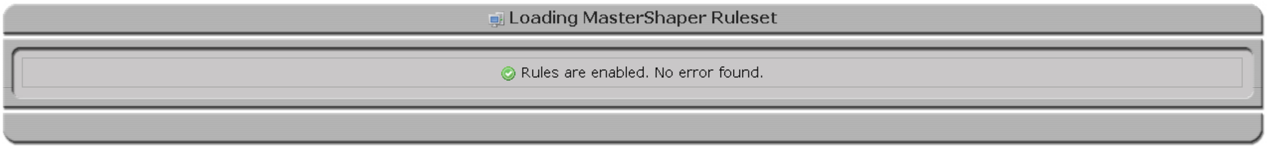
Don't forget - the first matching chain will get the traffic.

You can change the chain- & pipe-positions with the purple and turquoise arrows.

2.4.5 Rules

2.4.5.1 Load

This will make a bulk load of all MasterShaper rules. After every configuration change the rules have to be reloaded. From the technical view MasterShaper will first unload all rules and then load the new configuration.



Loading MasterShaper Ruleset

✓ Rules are enabled. No error found.

If you see a green check – everything is OK and rules are enabled. If you see a red X and some error messages then try to load the ruleset by *Rules* → *Load (debug).*

2.4.5.2 Load (debug)

This will load the ruleset rule by rule and return every error a rule makes.

2.4.5.3 Show

Show will displays you every command which would get loaded when enabling MasterShaper. This includes tc commands as well iptables commands (if iptables-matching is used).

2.4.5.4 Unload

This will disable MasterShaper's shaping functionality (if loaded).

2.4.6 shaper_loader.sh

With *MasterShaper* V0.31 a new rules-loader script was introduced. This was necessary, because a script has to be able to totally clean-up any residues from MasterShaper iptables rules. This is done by the *shaper_loader.sh* script now.

Also it loads the tc- and iptables-ruleset so this is the only script now which needs root privileges (*sudo*). This will also speedup activating iptables ruleset because sudo hasn't to be accessed for every rule.

Follow the installation procedures!

2.5 Tools

2.5.1 Runlevel-Init-Script

If you extract the MasterShaper install archive (tar.bz2), you will find a file called *mastershaper.init* in the *tools-directory*.

It's a first version of a runlevel init script. If you want to be able, that the shaper settings will be immediate loaded after a reboot you can use this file in the runlevel scripts.

You can also use this file as an ip-up script for the pppd daemon. You have the adept the MasterShaper path with the variable PATH_TO_MS in the script. This script has to be called with root privileges.

2.6 Shaping Examples

2.6.1 Shaping on a Web-Server

2.6.1.1 Guidelines

- A standard LAMP-Web-Server (Linux, Apache, MySQL, PHP) with an FTP-Server.
- Connected trough DSL (PPPOE) on Ethernet with a synchronous 2Mbit/s link. IP 1.1.1.1.
- 64kbit/s should always be guaranteed to SSH (TCP/22) but can use the whole bandwidth if available. It should have a high priority.
- HTTP should have a fixed rated at 1024kbit/s but can use the whole bandwidth if available.
- FTP should have 512kbit/s available but can use the whole bandwidth if available.
- All other traffic become 256kbit/s and can't use more then 768kbit/s.

2.6.1.2 Implementation

1. Go to Settings→Options and define the *inbound* and *outbound bandwidth*. Specify *HTB* as *Queuing Discipline*. Select iptables as Traffic Filter.
2. Create a *service level* with a input & output rate definition which is *equal* the *maximum bandwidth* (2048/2048/kbit/s).
3. Create a *service level* which has a rate of 64kbit/s and enter 2048kbit/s as ceil parameter. Set the *priority* to *high*.
4. Create a *service level* which has a rate of 1024kbit/s and enter 2048kbit/s as ceil parameter. Set the *priority* to *normal*.
5. Create a *service level* which has a rate of 512kbit/s and enter 2048kbit/s as ceil parameter. Set the *priority* to *normal*.
6. Create a *service level* which has a rate of 256kbit/s and enter 768kbit/s as ceil parameter. Set the *priority* to *low*.
7. Take a look on the ports-listing if you can find “http”, “https”, “ftp”, “ftp-data” and “ssh”. If not available, create new port definitions for these filters.
8. Create a *target* for the address 1.1.1.1.
9. Create a *filter* “Web-Traffic”. Select protocol *TCP*, assign the *ports* “http” & “https”.
10. Create a *filter* “FTP-Traffic”. Select protocol *TCP*, assign the *ports* “ftp” & “ftp-data”. Select “Match FTP data channel”.
11. Create a *filter* “SSH-Traffic”. Select protocol *TCP*, assign the *port* “ssh”.
12. Create a new *chain*. The *service level* for this chain is the 2048kbit/s level. For *fall-back* use the service level with the rate of 256kbit/s. On “Affecting” choose as source “any”, as *target* 1.1.1.1 and select both directions.
13. Create a *pipe*. Choose the created *chain*, select the *filter* “Web-Traffic”, the *service level* with the rate of 1024kbit/s and choose both directions.
14. Create a *pipe*. Choose the created *chain*, select the *filter* “FTP-Traffic”, the *service level* with the rate of 512kbit/s and choose both directions.
15. Create a *pipe*. Choose the created *chain*, select the *filter* “SSH-Traffic”, the *service level* with the rate of 64kbit/s and choose both directions.
16. Click “Overview” and take a look on your MasterShaper configuration.
17. Select Rules → Load and activate your new rule set.

2.6.2 Shaping on a gateway

2.6.2.1 Guidelines

- A standard Linux-based Internet router with a mail server and pop3/imap access.
- Connected trough Ethernet with a synchronous 2Mbit/s link. IP 1.1.1.1.
- For remote control of the clients (rdp, vnc, radmin) and SSH (TCP/22) 128kbit/s should always be guaranteed. It should have a high priority.
- Mailing (SMTP, POP3, IMAP) should not block the whole bandwidth. Guaranteed 256kbit/s. A Maximum of 1024kbit/s. Lowest Priority
- HTTP & FTP should have 512kbit/s available but can use the whole bandwidth if available.
- All other traffic become 256kbit/s and can't use more then 768kbit/s.

2.6.2.2 Implementation

1. Go to Settings→Options and define the *inbound* and *outbound bandwidth*. Specify *HTB* as *Queuing Discipline*.
2. Create a *service level* with a input & output rate definition which is *equal* the *maximum bandwidth* (2048/2048/kbit/s).
3. Create a *service level* which has a rate of 128kbit/s and enter 2048kbit/s as ceil parameter – for both directions. Set the priority to high.
4. Create a *service level* which has a rate of 256kbit/s and enter 1024kbit/s as ceil parameter – for both directions. Set the priority to low.
5. Create a *service level* which has a rate of 512kbit/s and enter 2048kbit/s as ceil parameter – for both directions. Set the priority to normal.
6. Create a *service level* which has a rate of 256kbit/s and enter 768kbit/s as ceil parameter – for both directions. Set the priority to lowest.
7. Take a look on the ports - listing if you can find "http", "https", "ftp", "ftp-data", "ssh", "rdp", "vnc", "radmin". If not available, create new port definitions for this filters.
8. Create a *target* for the address 1.1.1.1.
9. Create a *filter* "Web-Traffic". Select protocol *TCP*, assign the ports "http", "https" and "ftp".
10. Create a *filter* "Mail-Traffic". Select protocol *TCP*, assign the ports "smtp", "pop3" and "imap".
11. Create a *filter* "Remote-Control". Select protocol *TCP*, assign the port "ssh", "rdp", "vnc" and "radmin".
12. Create a new *chain*. The *service level* for this chain is the 2048kbit/s level. For fall-back use the service level with the rate of 256kbit/s and lowest priority. On "Affecting" choose as source "any", as target 1.1.1.1 and select both directions.
13. Create a *pipe*. Choose the created *chain*, select the *filter* "Web-Traffic", the *service level* with the rate of 512kbit/s and choose both directions.
14. Create a *pipe*. Choose the created *chain*, select the *filter* "Mail-Traffic", the *service level* with the rate of 1024kbit/s and choose both directions.
15. Create a *pipe*. Choose the created *chain*, select the *filter* "Remote Control", the *service level* with the rate of 128kbit/s and choose both directions.
16. Click "Overview" and take a look on your MasterShaper configuration.
17. Select Rules→Reload now and activate your new rule set.

2.6.3 Shaping per department

2.6.3.1 Guidelines

- A standard Linux-based internet router which manage the internet access of 4 departments.
- The router uses IMQ in BB mode on the external interface, so you see LAN addresses on the outgoing IMQ device before NAT.
- Connected trough Ethernet with a synchronous 2Mbit/s link. IP 1.1.1.1.
- The internal networks of the departments are 172.16.1.0/26, 172.16.1.64/26, 172.16.1.128/26, 172.16.1.196/26.
- Each department has guaranteed 512kbit/s but can lend unused bandwidth from other departments.
- Departments are using VoIP from a SIP-Provider and so connecting to a SIP-server. Low Latency has to be guaranteed for this service.
- HTTP & FTP should have 128kbit/s and maximal use 256 of the bandwidth. Priority low.
- All other traffic become 64kbit/s and can't use more then 196kbit/s.

2.6.3.2 Implementation

1. Go to *Settings→Options* and define the *inbound* and *outbound bandwidth*. Specify *HFSC* as Queuing Discipline.
2. Create 4 targets – each one with the assigned subnets of the departments.
3. Create a *service level* which has a rate of 512kbit/s and enter 2048kbit/s as max parameter
4. Create a *service level* which has a rate of 256kbit/s and enter 100ms for max delay.
5. Create a *service level* which has a rate of 128kbit/s and enter 256kbit/s as max parameter.
6. Create a *service level* which has a rate of 64kbit/s and enter 196kbit/s as max parameter.
7. Take a look at the ports - listing if you can find "http", "https", "ftp", "ftp-data". Check out, on which port-range you connect to the SIP-Provider and create a port definition for this.
8. Create a *filter* "VoIP-Traffic". Select protocol UDP, assign the ports "voip".
9. Create a *filter* "Web-Traffic". Select protocol TCP, assign the ports "http", "https", "ftp" and "ftp-data".
10. Create 4 new *chains*. The service level for this chains is the 512kbit/s level. For *fall-back* use the service level with the rate of 64kbit/s. On "Affecting" choose as source "any" and as target on of your defined department targets and select both directions.
11. Create 4 *pipes*. Choose the created department chains, select the filter "VoIP-Traffic", the service level with the rate of 256kbit/s and choose both directions.
12. Create 4 pipes. Choose the created department chains, select the filter "Web-Traffic", the service level with the rate of 128kbit/s and choose both directions.
13. Click "Overview" and take a look on your MasterShaper configuration.
14. Select Rules→Reload now and activate your new rule set.

3 License

MasterShaper is licensed by GNU General Public License (GPL) Version 2. All rights reserved by author *Andreas Unterkircher (andreas.unterkircher@netshadow.at), http://shaper.netshadow.at*.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

4