UNIVERSITY OF HELSINKI
DEPARTMENT OF COMPUTER SCIENCE

Software Development Project

# PAKETTI

User's Manual

Marko Buuri
Lasse Gustafsson
Tommi Rantala
Pekka Rousu
Jaakko Saaristo

16th December 2003

## Version History

| Version | Date | Changes |
|---|---|---|
| 1.1 | 16.12.2003 | Final version |
| 1.0 | 25.11.2003 | First draft |

# Contents

# 1 Introduction

Paketti is a software for graphical management of bandwidth management settings on Linux systems.

## 1.1 Document Purpose

This User's Manual describes how to install, configure and use Paketti on a personal computer running a Linux system. The following information is provided:

- software and hardware requirements,
- installation and uninstallation procedures, and
- user interface explanation.

## 1.2 Document Structure

Section 2 presents a general overview on the software and related security issues. The following sections cover issues on the two software components: Control Component (Section 3) and Graphical User Interface (Section 4).

# 2 General

This section provides an overview on the software and related security issues.

## 2.1 Software Overview

Paketti is a software for managing Linux bandwidth management settings. The software consists of a server and a client named *pakettid* and *paketti* respectively. The service pakettid is referred to as *Control Component* and is installed on every system user needs to configure. The service functionality includes invoking the Linux command line traffic control tool *tc* which, in turn, invokes the kernel networking interface. The user interacts with the software via the second software component: *Graphical User Interface* or *GUI*, which communicates with any Control Component using TCP/IP connectivity. An overview of the software architecture is presented in Figure 1. The server-side networking utilizes the *xinetd* [BRO03] super server for listening a configured network port and invoking pakettid as necessary.

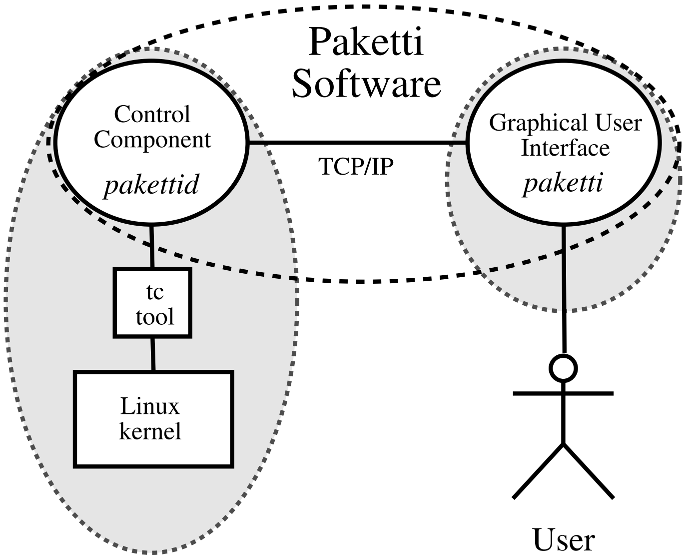


Figure 1: Software architecture

## 2.2 Warning on Security

**WARNING: This software does NOT provide methods of any kind for authenticating or encrypting the data communications between the software components. In addition, Control Component runs under system administrator (root) privileges. Even the ordinary use of the software should be considered a SEVERE security hazard.**

The software network traffic is not only open for observation to any third party that has access to a network device in between, but is also subject for connection hijacking and related man-in-the-middle attacks. The access control is handled through xinetd configuration.

The user of this software must acknowledge all the potential security risks that it bears. It is the recommendation of the authors that the software is not used to configure any system where the network traffic passes through untrusted network components. The authors of this software take no liability regarding any possible damages or mischief of any kind caused by the use of the software.

# 3 Control Component

Control Component is a service process that runs on the Linux server subject to management of the bandwidth management settings, and is responsible for the direct interaction with target system.

## 3.1 Requirements

Your system must meet the following hardware requirements for Control Component installation and usage.

- A standard personal computer.
- Some RAM.
- Some free hard disk space.

Your system must also meet the following software requirements.

- CSL 2 Linux, supported kernel version is 2.4.20-19.9.csl.
- TCP/IP networking interface.
- Linux software package iproute2 (provides the tc binary).
- GNU glibc 2.3.2.
- xinetd.

If you wish to use the HTB qdisc, your tc binary must have support for the same version of HTB your kernel is running. You will otherwise get errors when trying to commit settings that use the HTB qdisc.

Interface information is available only if you have the ifconfig binary installed.

To compile Control Component from sources, GNU gcc 3.2, GNU make, libxml2 and libxslt are also required. The software most likely works on other Linux distributions as well. The kernel should be compiled with the following options enabled under Networking options:

- Kernel/User netlink socket (CONFIG_NETLINK),
- Network packet filtering (CONFIG_NETFILTER),
- QoS and/or fair queueing (CONFIG_NET_SCHED), and
- all the options under QoS and/or fair queueing.

## 3.2 Installation

Quick setup:

- Extract the `paketti` package.
- `cd` to that directory.
- Modify the `Makefile` according to your needs.
- Run `make install-cc` as root.
- Reload xinetd.

Control Component is now ready to be used.

## 3.3 Installation in Detail

First, you must `cd` into the directory called paketti:

```
someuser@somehost:~$ cd paketti
someuser@somehost:paketti$
```

### 3.3.1 Installing with Make Install

Control Component is installed automatically if you run `make install-cc`. Installation paths are defined in the Makefile. If you wish to install eg. the pakettid binary in a certain location, modify the line BINDIR in the Makefile.

Typical output when running `make install-cc`:

```
root@somehost:paketti# make install-cc
make[1]: Entering directory ‘/home/ttrantal/paketti/src/pakettid’
gcc -W -Wall -Winline -pedantic ‘./find-libxml2 --cflags‘
  ‘./find-libxslt --cflags‘ -o pakettid -DXSL_PATH=\"/usr/local/
  paketti/paketti.xsl\" pakettid.c transform.c stccp.c
  tc_control.c arena.c error.c ‘./find-libxml2 --libs‘
  ‘./find-libxslt --libs‘
make[1]: Leaving directory ‘/home/ttrantal/paketti/src/pakettid’
mkdir -p /usr/local/paketti
cp -f pakettid/paketti.xsl /usr/local/paketti
cp -f pakettid/pakettid /usr/local/bin
sed -e "s:PAKETTID_PORT:20000:" \
    -e "s:PAKETTID_PATH:/usr/local/bin/pakettid:" \
        pakettid/pakettid.xinetd >/etc/xinetd.d/pakettid
```

### 3.3.2 Compiling

The software is automatically compiled when running make install. You may also compile it separately. For example:

```
someuser@somehost:paketti$ cd pakettid
someuser@somehost:pakettid$ make
gcc -Wall -O -Winline -pedantic `./find-libxml2 --cflags`
  `./find-libxslt --cflags` -o pakettid pakettid.c transform.c
  stccp.c tc_control.c arena.c error.c `./find-libxml2 --libs`
  `./find-libxslt --libs`
someuser@somehost:pakettid$
```

If the compilation causes errors, make sure you have all the necessary software properly installed.

### 3.3.3 Configuring xinetd

The xinetd configuration is automatically copied in place when running make install. You must, however, reload the xinetd service:

```
root@somehost:pakettid# /sbin/service xinetd reload
Reloading configuration:                                [  OK  ]
root@somehost:pakettid#
```

### 3.3.4 Verifying Installation

First, telnet to the configured port:

```
someuser@somehost:pakettid$ telnet localhost 20000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

If everything has succeeded, `pakettid` should be waiting for input from `stdin`. Press enter. The following outputted text indicates the software is currently running:

```
Protocol:STCCP/1
Length:120
<?xml version="1.0" encoding="UTF-8"?>
<MESSAGE><TYPE>ERROR</TYPE><DATA>Received an unexpected
newline.</DATA></MESSAGE>
```

To close the telnet session press Control-AltGr-9 and type quit:

```
^]
telnet> quit
Connection closed.
someuser@somehost:pakettid$
```

### 3.4 Uninstallation

Remove the configuration from xinetd and remove all installed binaries from your system.

## 4 Graphical User Interface

The traffic control settings are edited with Graphical User Interface (GUI). This chapter explains GUI in detail.

### 4.1 Requirements

Your system must meet the following hardware requirements for you to install and use GUI.

- A standard personal computer.
- A CPU that you find adequate for running Swing applications.
- Some RAM.
- Some free hard disk space.

Your system must also meet the following software requirements:

- Java 2 Standard Edition SDK 1.4 or later.
- Java 2 Runtime Environment, version 1.4 or later.
- Xerces XML parser, version 2.2.1 or later.

## 4.2 Installation

Quick setup:

- Extract the `paketti` package.
- `cd` to that directory.
- Modify the `Makefile` according to your needs.
- Run `make install-gui`.

The installation paths are defined in the Makefile and can be modified by editing the file. GUI is compiled first. To compile it without installing, see Section 4.3.

## 4.3 Compilation

GUI compilation requires the Xerces jar files located in the `paketti` directory. They are included in this working status package. The following files are required:

- xercesImpl.jar
- xercesSamples.jar
- xmlParserAPIs.jar

The files are available at Xerces home page [ASF03b]. After acquiring the required files the contents of `paketti` directory should look like this:

```
someuser@somehost:~$ ls paketti
dtdparser.jar paketti.jar    xercesSamples.jar
MANIFEST.MF   paketti/       xmlParserAPIs.jar
Makefile      paketti.dtd    xercesImpl.jar
```

GUI is compiled by running `make` in the directory `paketti` package was extracted. Compilation should look like this:

```
someuser@somehost:~$ make
make[1]: Entering directory /somepath/paketti'
javac -source 1.4 -classpath ./:xercesImpl.jar:
xercesSamples.jar:xmlParserAPIs.jar:dtdparser.jar
```

```
paketti/*.java
jar xf xercesImpl.jar
jar xf xercesSamples.jar
jar xf xmlParserAPIs.jar
jar xf dtdparser.jar
jar cmf MANIFEST.MF paketti.jar paketti/*.class
paketti.dtd dom/ javax/ org/ sax/ socket/ ui/ xni/
rm -rf dom/ javax/ org/ sax/ socket/ ui/ xni/
META-INF/
paketti.jar created.
make[1]: Leaving directory /somepath/paketti'
```

## 4.4 Running

If GUI is properly installed, it is run with the following command:

```
paketti IP PORT.
```

The parameter `IP` is either the host name or the IP address of the target system and Control Component. The parameter `PORT` is the TCP port used in the xinetd configuration during Control Component installation. For example:

```
someuser@somehost$ paketti localhost 20000
```

Alternatively, GUI can be run without installation with the following command:

```
java -jar somepath/paketti.jar IP PORT.
```

Upon running, GUI will first try to connect to Control Component using given IP address and port. If the connection fails for some reason, GUI exits with an error message.

## 4.5 Uninstallation

Remove the GUI jar package from your system.

# 5 Using the Software

The software can be used by first properly installing and running Control Component on wanted target system, following instructions given in Section 3. Next, GUI should be installed and run in any wanted system, following instructions given in Section 4. This chapter covers information on using the software with GUI.

## 5.1 The Main View

When successfully started, GUI will automatically execute *Load from daemon* (see Section 5.4.2 for details). If an error occurs upon the action, GUI exits after displaying an error dialog.

After GUI has succesfully received the settings from Control Component, the main window of GUI (Figure 2) will be shown. The bandwidth management settings are shown as a combination of an editable *tree* on the left and the tree elemens' attributes on the right. The tree elements are namely the qdiscs, classes and filters the bandwidth management settings consist of. Tree elements may be selected clicking the left mouse button on the tree elements, which will display the selected attributes of the selected element in the frame on the right.

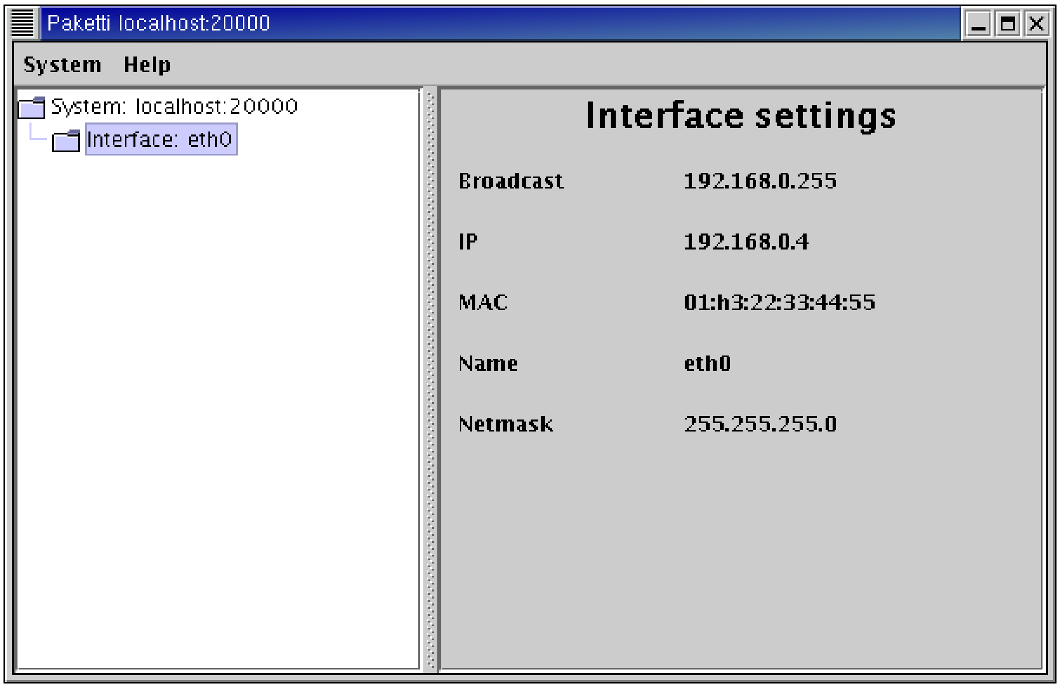


Figure 2: The Main View

## 5.2 Editing the Tree

The tree can be edited using mouse according to the instructions in the next subsections.

There are restrictions to the removal and addition of elements, and GUI only allows you to build a tree that is valid according to the rules on generating the settings in the Linux kernel. The possible elements and the restrictions on child element types are listed in table 1. The first column indicates the element name, the second column the possibility of removing the element and the third column the possible child elements for the element. GUI also checks through a rather complex pattern of restrictions regarding the number of the elements' child elements.

Table 1: The elements in the tree

| Element | Removable | Possible child elements |
|---|---|---|
| system | no | interface |
| interface | no | pfifoqdisc, bfifoqdisc, redqdisc, htbqdisc |
| redqdisc | yes | - |
| pfifoqdisc | yes | - |
| bfifoqdisc | yes | - |
| police | yes | - |
| htbqdisc | yes | htbclass, u32filter |
| htbclass | yes | pfifoqdisc, bfifoqdisc, redqdisc, htbqdisc, u32filter |
| u32filter | yes | match, police |
| match | yes | - |

### 5.2.1 Adding a New Element

New elements are added to the tree as children to existing elements. Click on an element with the right mouse button to see a context menu (Figure 3). The menu only displays the elements that are possible to add as a child to the chosen element. When an element is added, it instantly displayed (Figure 4) as a child element of the chosen element.

### 5.2.2 Removing an Element

Elements are removed by selecting remove from the context menu that is shown when clicking on an element with right mouse button (Figure 5). You will be prompted for confirmation on element removal (Figure 6). When an

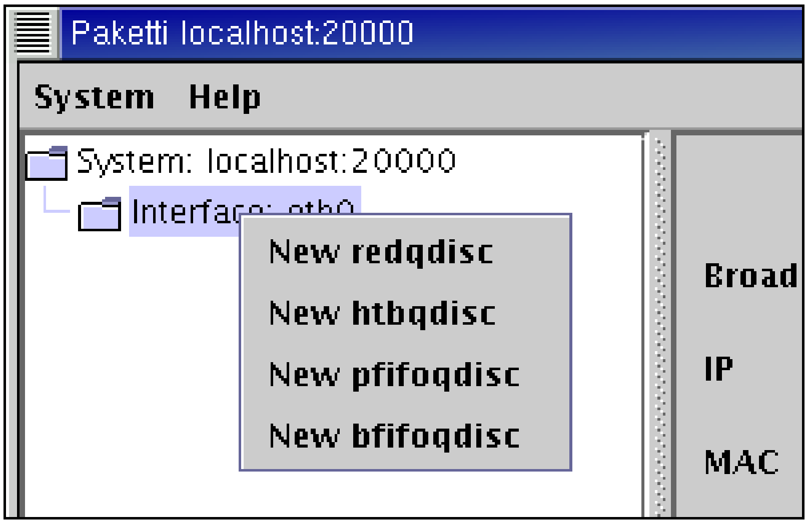


Figure 3: Adding a New Element

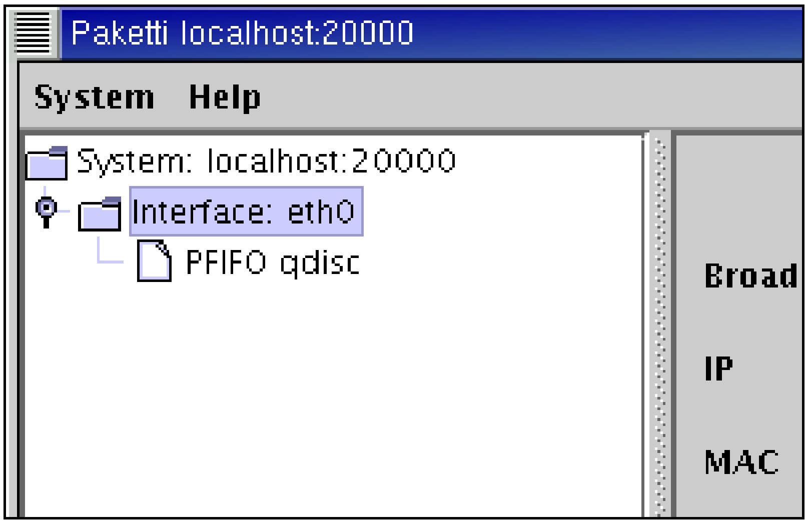


Figure 4: The Tree After Adding an Element

element is removed, it is instantly removed from the tree along with all it's child elements (Figure 7).

## 5.3 Editing Element Attribute Values

The element attribute values can be edited by first selecting the element you wish to modify by clicking it with the left mouse button (Figure 8). The attributes appear on the frame on the right. The attributes of the element

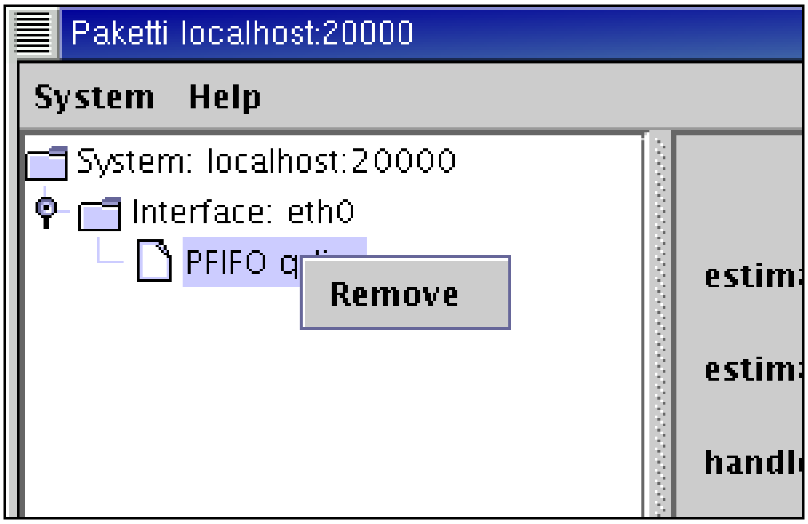


Figure 5: Removing an Existing Element


Figure 6: Remove Confirmation

are presented as text fields, if they are editable, and plain text labels if non-editable. The star (*) after an attribute name indicates the attribute is mandatory. You may edit the attribute values according to your needs.

The attribute value syntax is basically the same as in corresponding Linux traffic control tool (tc) command parameters. See for example Linux Advanced Routing HOWTO [HUB00] for more information. For example, the value for RED qdisc attribute "ecn" can be either "true" or "false".

## 5.4 Menu Actions

The menu bar has two menus: System and Help.

The System menu (Figure 9) includes most of the real functionality in the software. The System menu includes menu items for actions Commit, Load from daemon, View as tc script, Show stats, Export XML, Import XML, Import tc script and Quit. The Help menu (Figure 10) includes a single

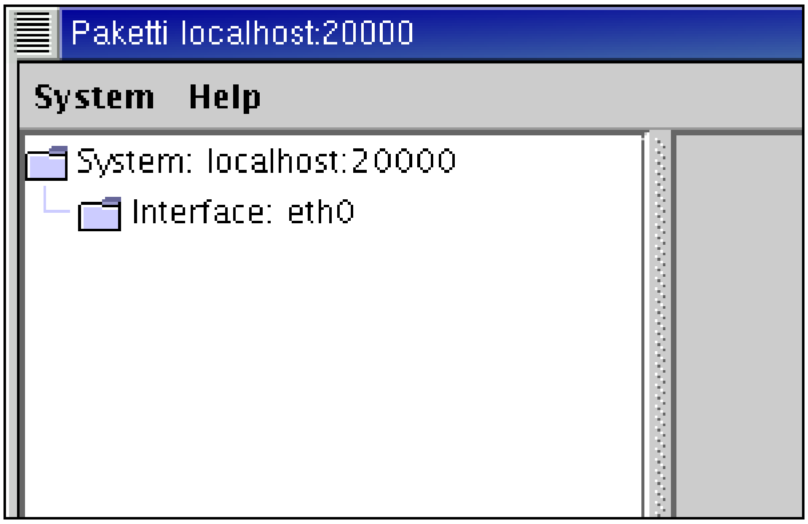


Figure 7: The Tree After Removing the Element

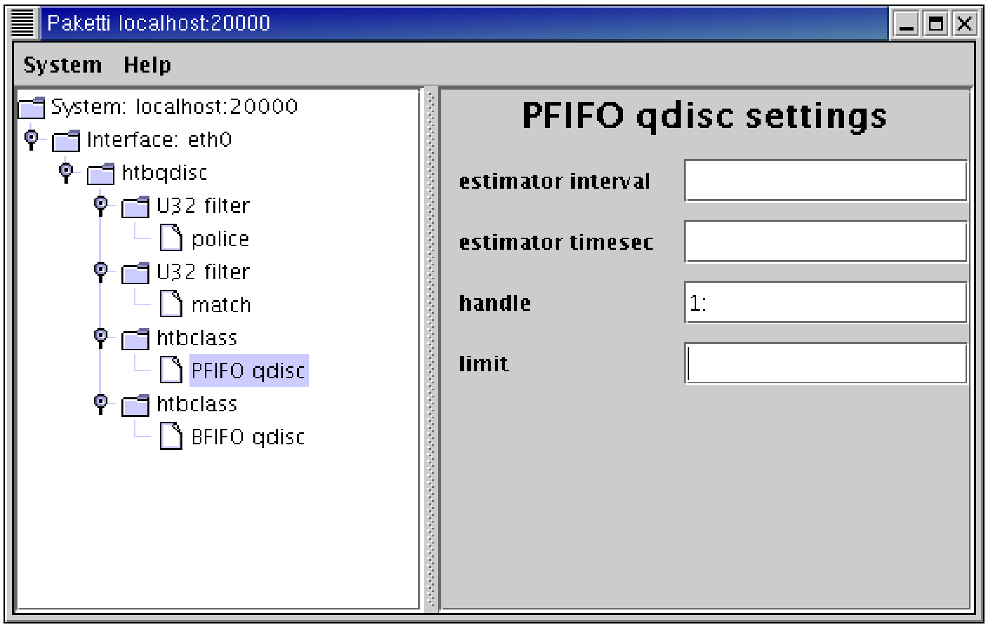


Figure 8: Editing Element Attribute Values

menu item for action About.

The actions Commit, Load from daemon, View as tc script, Show stats and Import tc script involve network communication with Control Component. During the data transfer you will see a progress bar (Figure 11) indicating the time left to connection timeout, which is *five seconds*. You cannot interact with GUI while the transfer is in progress. If the transfer succeeds, you will see the progress bar disappear. If an error occurs in Control Component, it will reply with an error message, which will be shown. If there is no reply from Control Component, timeout will occur and an error message will be shown (Figure 12).

The actions will be explained in detail in the following.

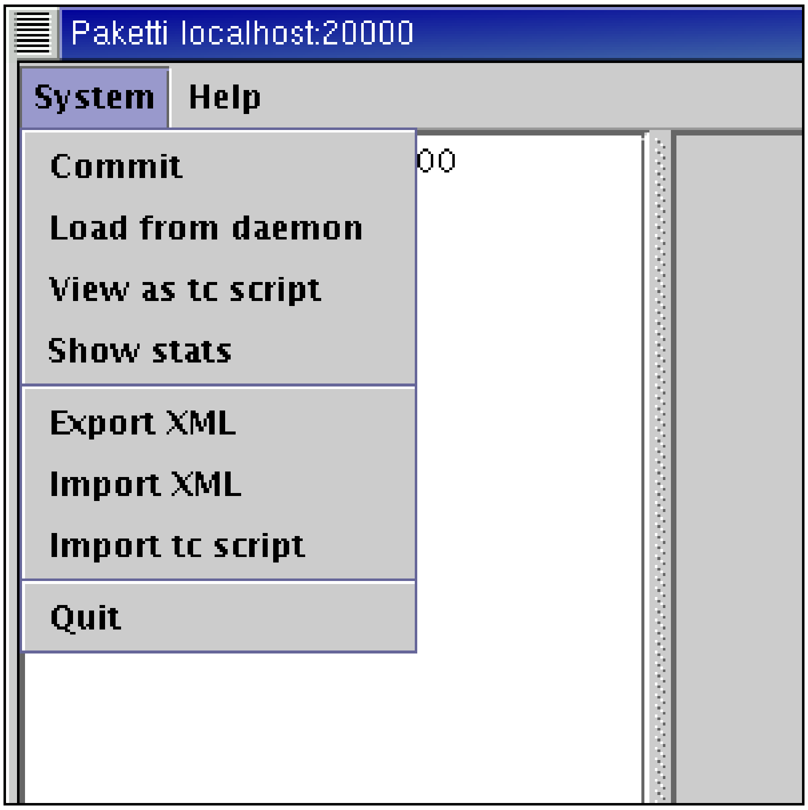


Figure 9: System Menu

### 5.4.1 Commit

Commit will send the currently edited settings to Control Component, which will run them into the target system. You will be prompted for confirmation (Figure 13). *The current settings in the target system will be overwritten!* If an error occurs in Control Component during Commit, for example due to invalid element attribute values, Control Component restores the settings to the state preceding the Commit.

**IMPORTANT!** After Control Component has run the settings in, it will query the target system for current settings and transfer back the settings that are reported to really exist in the system. These settings will be updated

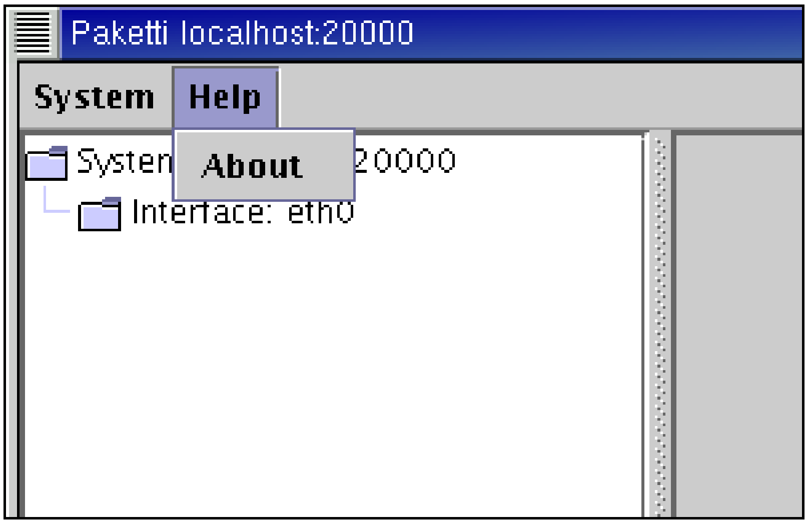


Figure 10: Help Menu

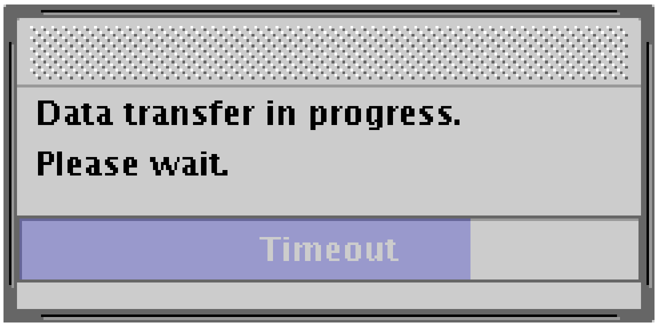


Figure 11: Waiting for Response

in the tree after Commit. *The tree contained elements and their structure along with certain element attribute values may be changed upon Commit!* This is due to incomplete functionality of the current available implementation of the Linux traffic control tool. The tree is updated to guarantee that what you see is what you got. The following anomalies are known to occur upon Commit:

- HTB qdisc contained class and filter structures will most likely be changed. The number of filters actually seems to increase.

- The “estimator interval” and “estimator timesec” attributes in every qdisc, HTB class and U32 filter element, along with RED qdisc element attributes ‘’avpkt”, ‘’bandwidth”, ‘’burst” and ‘’probablity”, will be set to empty. The values are actually run into the target system, but since

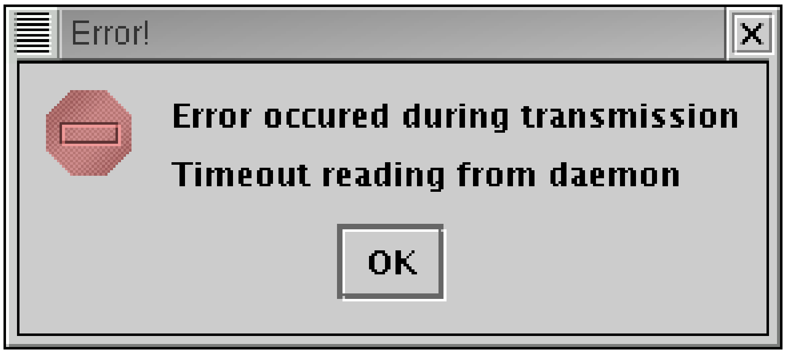


Figure 12: Timeout

they are not reported back, they will disappear from GUI.

- The Match element attribute “sampleargs” is set to the representation it is reported back in, for example value “ip tos 0xb8 0xff” will be set to “00b80000/00ff0000”.

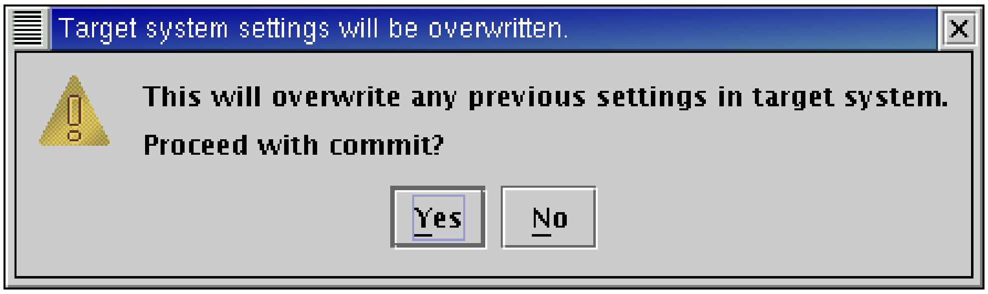


Figure 13: Commit Confirmation

#### 5.4.2 Load from Daemon

Load from daemon will load the current settings from the Control Component. You will be prompted for confirmation (Figure 14). If the action succeeds, the tree is updated corresponding to the received traffic control settings. *The currently edited settings in the tree will be overwritten!*

#### 5.4.3 View as tc Script

View as tc script shows the currently edited settings in the form of a tc script. This involves a network transfer of the edited settings to Control Component, which translates them to script form and sends them back. If

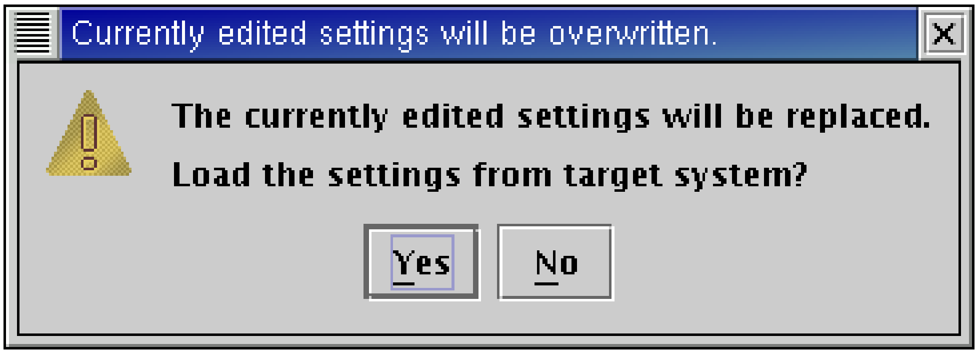


Figure 14: Load from Daemon Confirmation

the action succeeds, you will see a new window containing the tc script. On the bottom of the window there are two buttons: Save to file and OK. Save to file will display a file chooser window (Figure 17) with which you can choose a file to save the tc script in. OK will plainly close the window.

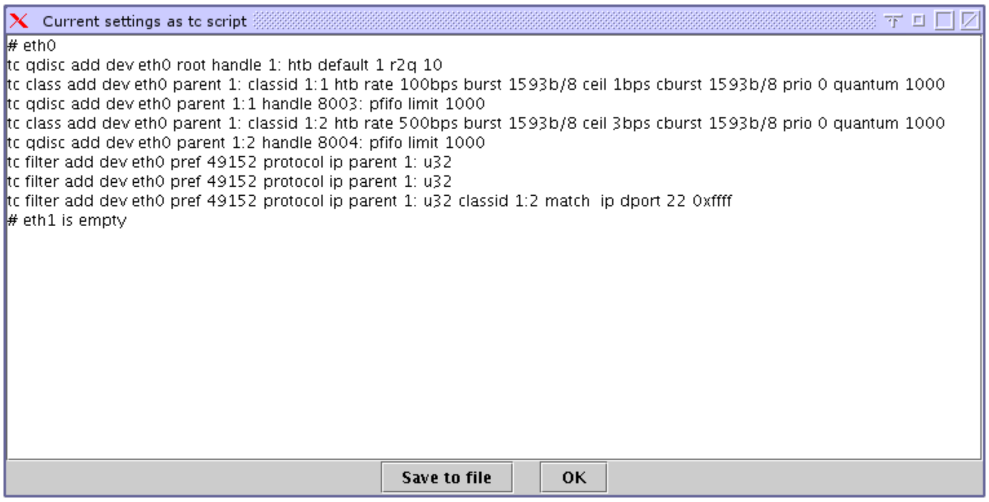


Figure 15: View as tc Script Window

### 5.4.4 Show Stats

Show stats shows bandwidth management related statistical information from target system. The information is displayed in a separate window (Figure 16) similar to the one in View as tc script action.

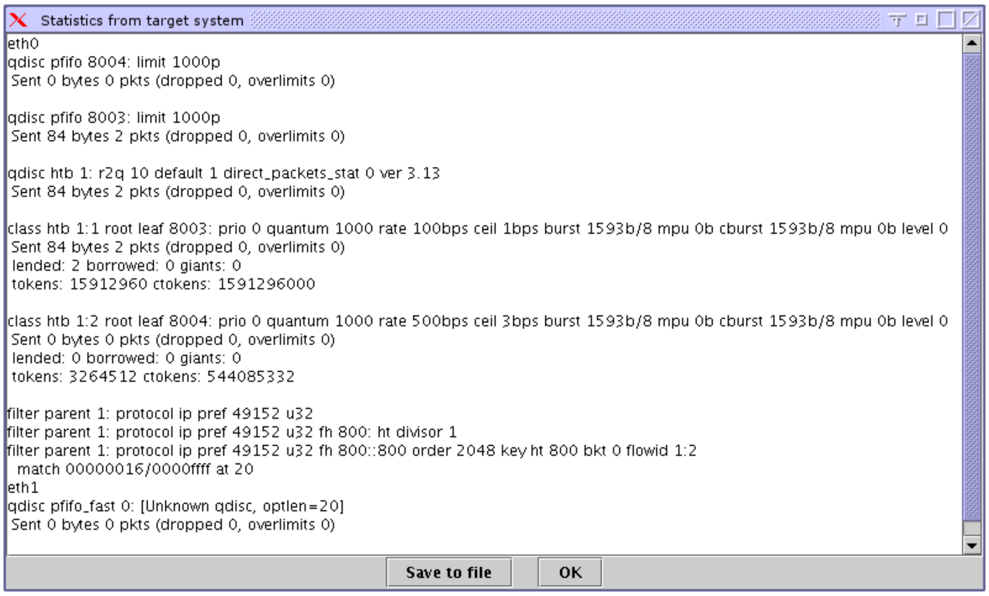


Figure 16: Show Stats Window

#### 5.4.5 Export XML

Export XML allows you to save the currently edited settings in a local file. You will see a file chooser window (Figure 17) which you can use to select any target file in the local file system. The file chooser defaults to the user's platform dependent default directory, typically the user's home directory on UNIX and “My Documents” on Windows platform. The created file is a plain text file containing the data in XML-encoded XML form. You will be prompted for confirmation in case the file exists and will be overwritten.

#### 5.4.6 Import XML

Import XML allows you to load previously saved settings from a local file. The file must be one previously saved using Export XML, and can be selected using a similar file chooser window as in Export XML. If the file chosen was valid, the tree will be updated corresponding to the imported settings. *The currently edited settings in the tree will be overwritten!*

#### 5.4.7 Import tc Script

Import tc script functions similarly to Import XML but instead of choosing a previously exported file you should choose a file containing a valid tc script. A tc script is a text file containing valid Linux tc commands separated with line feeds. It may also include empty lines and comment lines beginning

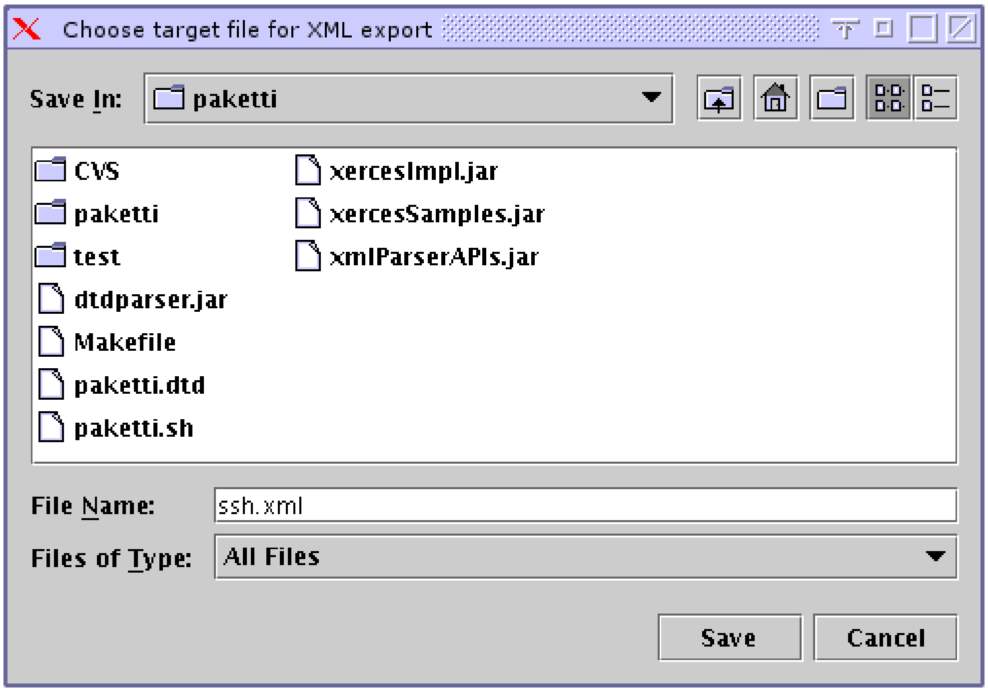


Figure 17: Target File Selection

with '#'. When a valid file is chosen, it will be transferred via network to Control Component which will translate the commands and transfer the settings back. After the transfer the tree will be updated corresponding to the imported settings. *The currently edited settings in the tree will be overwritten!*

### 5.4.8 Quit

Quit will exit the program. You will be prompted for confirmation (Figure 18). *The currently edited settings in the tree after last commit will be discarded!*

### 5.4.9 About

About shows information on the software and the names of the authors in a new window (Figure 19).


Figure 18: Quit Confirmation


Figure 19: The About Window

# List of Figures

# Bibliography


[ASF03b] Apache Software Foundation, Xerces2 Java Parser home page. http://xml.apache.org/xerces2-j/index.html,12.12.2003

[BRO03] Brown, R., Xinetd home page. http://www.xinetd.org,14.9.2003

[HUB00] Hubert, Bert et.al., Linux 2.4 Advanced Routing HOWTO, 2000. http://www.linuxdocs.org/HOWTOs/Adv-Routing-HOWTO.html,1.10.2003