

FCM (Flow Control Module) User Documentation  
RAMI (Rate Adjustment by Managing Inflows)  
Team M

Daniel Connor (connor)      Chris Edwards (caedwa)      Rod Howard (rihoward)  
Maya Muthuswamy (mayadm)      Lars Yencken (lljy)

October 30, 2002

Maintained By: Chris Edwards (caedwa@students.cs.mu.oz.au)  
Version: 1.0

**Abstract**

User documentation for the installation and use of the FCM component of the RAMI package.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What is RAMI? . . . . .	3
1.1.1	What is FCM? . . . . .	3
1.2	Requirements . . . . .	3
1.2.1	Kernel Requirements . . . . .	3
1.2.2	Utility Requirements . . . . .	3
<b>2</b>	<b>Kernel Setup</b>	<b>4</b>
2.1	Router Kernel Options . . . . .	4
2.2	Receiver Kernel Options . . . . .	5
2.3	Compiling the Kernel . . . . .	5
<b>3</b>	<b>Compilation and Installation</b>	<b>8</b>
3.1	Compilation . . . . .	8
3.2	Installation . . . . .	8
3.3	Logger Compilation and Installation . . . . .	8
<b>4</b>	<b>Module Usage</b>	<b>9</b>
4.1	Router Usage . . . . .	9
4.1.1	Module Loading and Options . . . . .	9
4.1.2	tc Usage . . . . .	9
4.1.3	Module Unloading . . . . .	10
4.2	Receiver Usage . . . . .	10
4.2.1	Module Loading and Options . . . . .	10
4.2.2	Module Unloading . . . . .	10
4.3	Logger Usage . . . . .	10

## List of Figures

1	The “Networking Options” configuration option, as found when running <code>make menuconfig</code> . . . . .	4
2	The ‘QoS and/or fair queueing’ menu option under ‘Networking Options’ in the Linux kernel configuration . . . . .	5
3	The ‘QoS and/or fair queueing option under ‘Networking Options’ in the Linux kernel configuration . . . . .	6
4	The ‘Netfilter configuration’ option . . . . .	6
5	The ‘Network packet filtering’ option . . . . .	7
6	The ‘Connection tracking’ option . . . . .	7

# 1 Introduction

This section will describe what FCM and RAMI actually are. It will also give a brief introduction to the requirements of FCM.

## 1.1 What is RAMI?

“RAMI” is an acronym for “Rate Adjustment by Managing Inflows”. In other words, it is a system designed to efficiently share the bandwidth of a low bandwidth network link. It uses the algorithm described in [1], which adjusts the advertised window size in TCP packets.

### 1.1.1 What is FCM?

FCM is a set of Linux kernel modules which implement the algorithm used by “RAMI”. There are two kernel modules, one for each of the Router and the Receiver. The Router is the computer on the Internet side of the low bandwidth link, while the Receivers are the computers which access the Internet via the low bandwidth link. For the system to work, the Router must have its module installed, and any Receivers wishing to take advantage of it should have the Receiver module installed.

In addition to the two kernel modules, a logger utility is provided, which can be used to record information about the performance of the kernel modules and the algorithm in particular. The files produced by the logger are in a format suitable for reading by the SAM program, which can be obtained separately.

## 1.2 Requirements

To install the FCM kernel modules, they must first be compiled. Because of differences in compiled kernels due to selected options, it is important that the modules be compiled with the correct kernel source. The modules were developed for kernel version 2.4.18, but may work on other 2.4 kernels, however incompatibilities may occur.

### 1.2.1 Kernel Requirements

Before the Router module can be compiled, the following options must be compiled into the kernel:

- QOS and/or fair queueing

Before the Receiver module can be compiled, the following options must be compiled into the kernel:

- Network packet filtering (replaces ipchains)
- Connections tracking (required for masq/NAT)

If you are unsure how to add these options, the next section will explain where to find them.

### 1.2.2 Utility Requirements

Use of the Router module requires the ‘tc’ (Traffic Control) utility. In the Debian 3.0 distribution, this can be found in the iproute package. To install this, the following command should be run as root: `apt-get install iproute`

The Receiver module requires no utilities other than the standard module loading tools.

The logger requires no kernel options or extra utilities. It does however require the Router to be installed and working.

## 2 Kernel Setup

As explained in the previous section, the system's kernel must have certain options present. These options are added when the kernel is compiled. They may already be present in the system's kernel, but if not, they must be added and the kernel recompiled.

While a full explanation of how to compile a kernel is not provided, a brief description will be given.

Before compiling a kernel, the configuration must be created. To do this, change into the kernel source directory (typically `/usr/src/linux` and run the command: `make menuconfig`

This provides you with a simple menu system to change options.

All of the following options are found under the "Networking Options" configuration item, as shown in Figure 1.

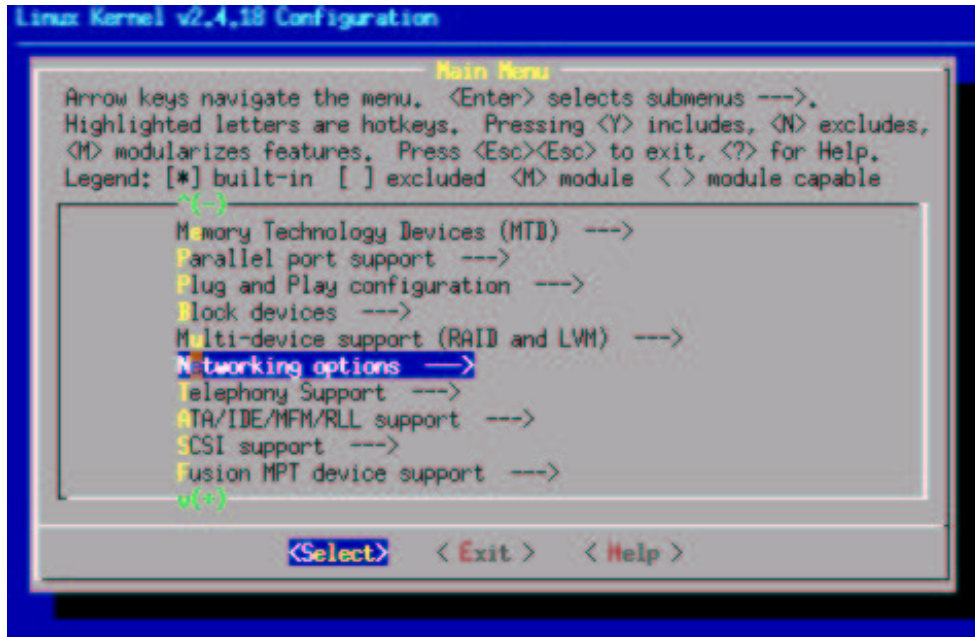


Figure 1: The "Networking Options" configuration option, as found when running `make menuconfig`

### 2.1 Router Kernel Options

The 'QoS and/or fair queuing' option must be selected. This can be found under the menus shown in Figures 2 and 3.

Without these options selected, the FCM Router module will not compile.

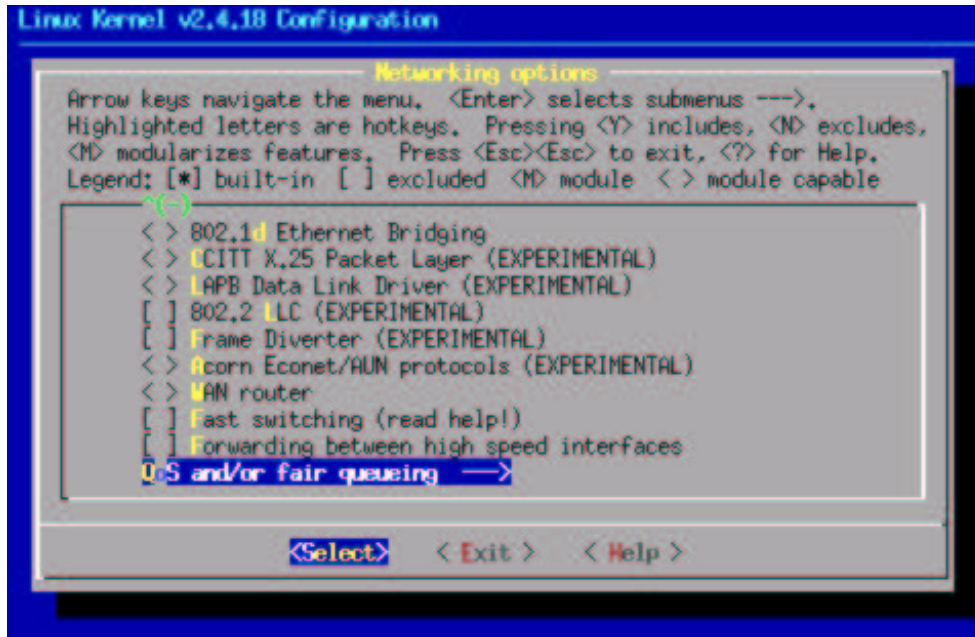


Figure 2: The ‘QoS and/or fair queueing’ menu option under ‘Networking Options’ in the Linux kernel configuration

## 2.2 Receiver Kernel Options

The ‘Netfilter’ and ‘Connection Tracking’ options must be selected in the kernel configuration. These can be found in the menus shown in Figures 4, 5 and 6.

Without these options selected, the FCM Receiver module will not compile. They may be compiled as modules if possible.

## 2.3 Compiling the Kernel

For a more detailed explanation of how to compile the kernel, and problems which may occur, see the Kernel HOWTO at <http://www.tldp.org/HOWTO/Kernel-HOWTO.html>

However, in Debian, creating a kernel image package is simple a matter of using the `kernel-package` package, and typing the following commands:

```
make-kpkg clean
fakeroot make-kpkg kernel_image
```

This will create a package in the `/usr/src` directory.



Figure 3: The 'QoS and/or fair queueing option under 'Networking Options' in the Linux kernel configuration

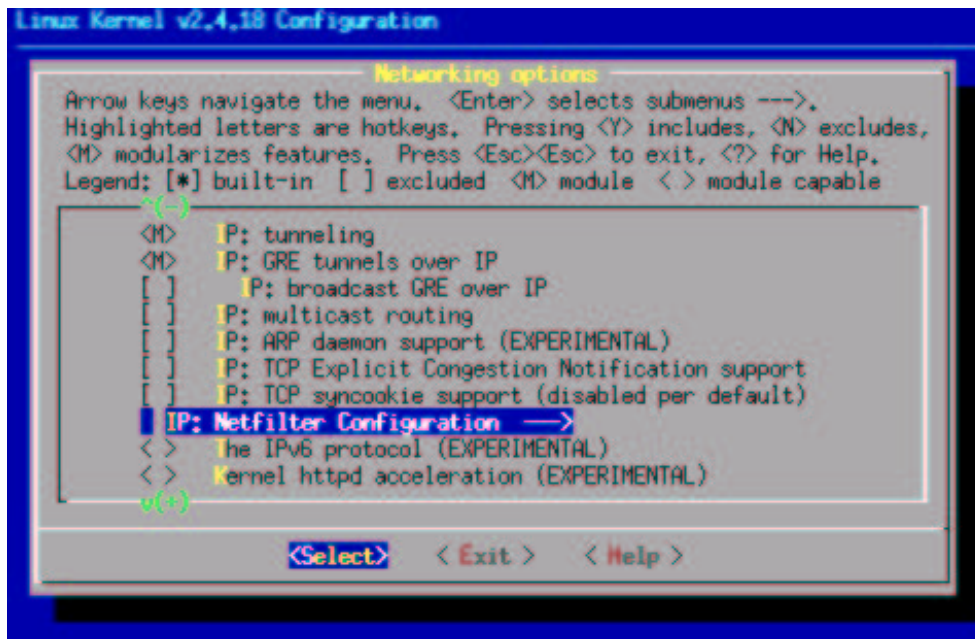


Figure 4: The 'Netfilter configuration' option

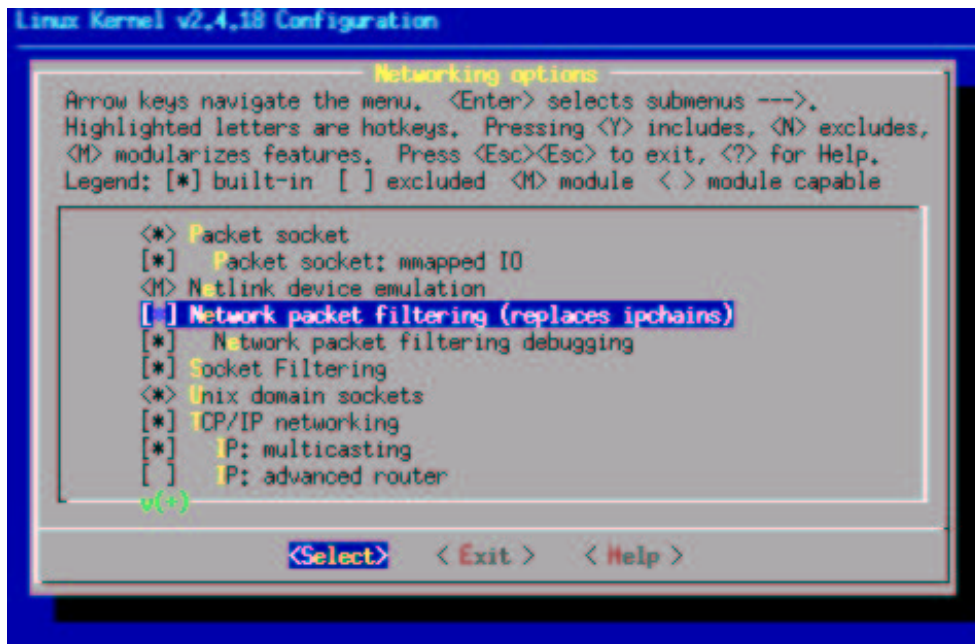


Figure 5: The 'Network packet filtering' option

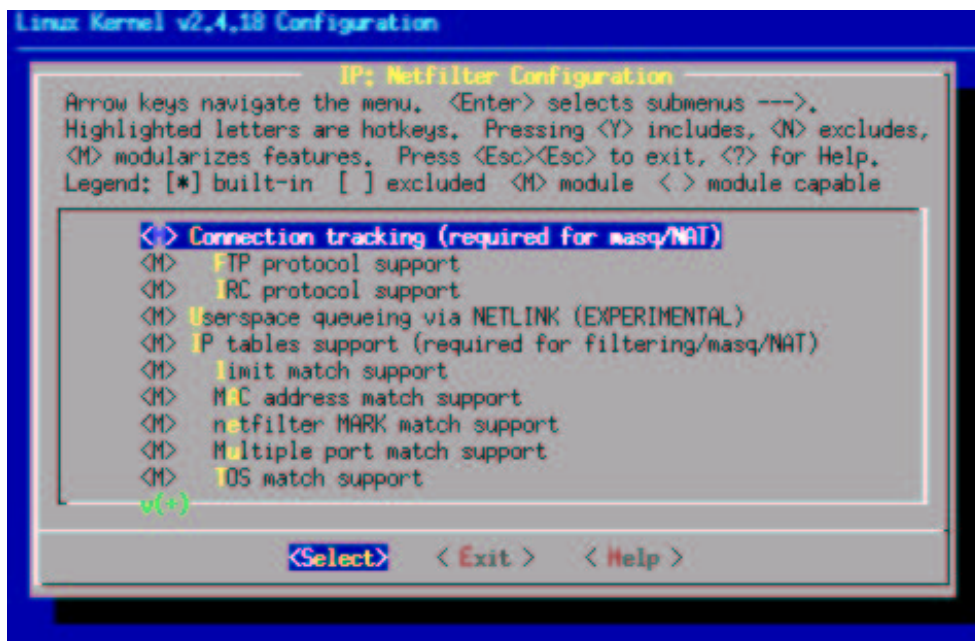


Figure 6: The 'Connection tracking' option

## 3 Compilation and Installation

This section describes the steps necessary to compile the Router and Receiver kernel modules to binary module files after their source has been installed, and how to place these files in the correct location to be loaded. It also describes installation of the logger.

### 3.1 Compilation

Once you are running a kernel with the the correct options compiled into it, both the Router and Receiver modules are simple to compile. First the kernel must have been compiled (with the correct options selected as described in Section 2). Then it is simply a matter of running `make` in the relevant directory, depending on which module is being compiled. The directories are:

**router** - The Router module

**receiver** - The Receiver module

After running ‘make’, assuming no errors occurred, the module will then be ready to load using a tool such a ‘insmod’ or ‘modprobe’. However it is suggested that the module be installed in the kernel’s module directory, to facilitate easy loading by the system. This procedure is described in Section 3.2. Note that it is not necessary to be root until the module is installed (see section 3.2). In fact, it is recommended that the module be compiled as a non-root user.

### 3.2 Installation

For this step, it is necessary to be logged in as the root user, either directly or with the `su` command. Installation follows a similar procedure to that for compilation (as described in Section 3.1), however the running of `make` is replaced with `make install`. This will install into the currently running kernel’s module directory. If you wish to use a non-standard directory, please edit the included `Makefile`.

### 3.3 Logger Compilation and Installation

The logger is available as a binary package for Debian, and hence does not need to be compiled. Instead, `dpkg`, should be used to install it. The commandline to install it is: `dpkg --install <filename>`, where `<filename>` is the filename of the Debian package provided.

For distributions other than Debian, or if using the source, the logger must be compiled. This can be done by simply typing `make` in the `logger` directory of the source package.

## 4 Module Usage

This section describes the procedure for making use of the features of the given modules. It assumes the correct module(s) have been compiled and installed as described in Section 3. To install, load and use the kernel modules, you must be logged in as the root user, either directly or with the `su` command.

### 4.1 Router Usage

The Router module is provided as a ‘Queueing Discipline’ to be used by the QoS feature of the kernel. The module must first be loaded into the kernel, and must then be activated using the ‘tc’ utility.

#### 4.1.1 Module Loading and Options

The Router module is named `sch_qsize`, and hence can be loaded with the command: `modprobe sch_qsize`. However, there are options which it can be passed, including one required option. The options are:

**maxrate** - **(Required)** The data rate of the Router’s outgoing link. Due to limitations with ‘tc’ it is not possible to specify this separately for each interface without modifications to ‘tc’. The units are kilobits/sec.

**bval** - The value of the constant  $b$  used by the Router’s calculations. The default value is 10.

**q\_min** - The length of the queue upon which  $p(q)$  becomes non-zero. This is the point where Receiver connections start reducing their window, which slows down connections. The default value is 30.

**limit** - The maximum length of the queue. This is used to set a limit to the amount of memory which the queue can use, and also to prevent excessive latency, by dropping excess packets. If this argument is set to 0, then FCM will automatically use the standard queue length. This is the default.

The options listed above are passed on the `modprobe` command line in the form: `name=value`. For example, to set the value of `maxrate` to 10000, and `bval` to 10, the following command line would be given:

```
modprobe sch_qsize maxrate=10000 bval=10
```

After being loaded, the module will still not be operational until it is attached to a network interface using `tc`, as described in the next section.

#### 4.1.2 tc Usage

Once the Router module has been loaded, as described in Section 4.1.1, it can be attached to network interfaces. Doing so puts it in the path of all packets which leave the interface. These packets are, if possible, modified to transmit extra information about the status of the Router to the Receivers which are reached by the appropriate network interface.

Full usage of the ‘tc’ command is beyond the scope of this document, however for simple cases where no extra QoS modules are used, this document’s examples are sufficient. If you wish to combine QoS modules, more information can be found at <http://lartc.org/howto/>. Otherwise, the following command attaches the Router module to the `eth0` network interface, and is suitable for all but the most exotic setups:

```
tc qdisc add dev eth0 root handle 1: qsize
```

To remove it from the interface, the following command can be used:

```
tc qdisc del dev eth0 root handle 1: qsize
```

For a network interface other than `eth0`, (such as `ppp0` for a PPP connection), replace `eth0` with the interface’s name in the examples above.

### 4.1.3 Module Unloading

It's not usually necessary to unload the Router module, as simply running `tc` to remove it from an interface (as described in Section 4.1.2) prevents it from being used. However, the module can be unloaded from memory entirely. This is done with the following command:

```
rmmod sch_qsize
```

However note that before the module can be unloaded, it must be removed from use using the `tc` command.

## 4.2 Receiver Usage

The Receiver module is provided as a Netfilter module (unlike the Router module). Once loaded, it automatically activates itself until it is unloaded.

### 4.2.1 Module Loading and Options

The Receiver module is named `rami_receiver`, and hence can be loaded with the command: `modprobe rami_receiver`. However there are options which it can be passed, but none of these are required. The options are:

**ps\_thresh** - The value of `PS_THRESH` used to determine the ending of the prime start phase. The default value is 0.

**alpha** - The smoothing factor used to calculate the received rate. Values between 1 and 100 are valid. A higher value gives a smoother and more consistent receiver rate, however it may lag behind the actual rate. The default value is 250.

**tau** - Determines the rate at which the window size tries to increase. The default value is 1.

The format of these options is the same as described in Section 4.1.1 for the Router module. However, the following example loads it with a value of 0 for `ps_thresh` and a value of 87 for `alpha`:

```
insmod rami_receiver.o ps_thresh=0 tau=87
```

### 4.2.2 Module Unloading

To stop using the Receiver module, it must be unloaded from memory. This is done with the following command:

```
rmmod rami_receiver
```

## 4.3 Logger Usage

The FCM logger requires that the Router module be loaded and active on the system. See Section 4.1 for information on how to use the Router module. This section will assume that the Router module has been loaded as described in Section 4.1, and also that the Logger has been installed as described in Section 3.

The logger is named `rami-logger` and is run using this name.

The logger accepts two options, both of which are required. Both options require an argument. The arguments are:

**-interval** (Or abbreviated as **-i**) - **(Required)** This option specifies the interval at which measurements are made by the logger, and hence the rate at which output is produced. The units are tenths of a second. For example, a value of 10 would mean that a record of output is produced once every second.

**--duration** (Or abbreviated as -d) - **(Required)** This option specifies the length of time over which the logging will take place. Once logging has occurred for this length of time, the logger will exit. The units for the required argument is seconds. For example, a value of 60 would mean that logging lasts for 1 minute (or 60 seconds).

Following the two arguments (which may be specified in either order), a filename must be given. This filename is used for the output file. The user must have permission to open a file of that name for writing. Any data in the file will be erased and a new file created.

For example, to create a log for 1 minute with an interval of 1 second between records, into the file `output.nlog` the following command would be used:

```
rami-logger --interval 10 --duration 60 output.nlog
```

If you wish to read the files written, the format should be fairly self explanatory. Values are separated by spaces.

## References

- [1] S. Hanly, R. Mukhtar, and L. Andrew. A novel algorithm for receiver-based management of low bandwidth access links.