PROATM ADAPTERS ON LINUX

Quick Installation Guide

Version 3.1

14/12/2014

1. INTRODUCTION

The PROATM-V155 and PROATM-E155 adapters are based on the IDT77252 SAR. The IDT77252 driver provided by the Linux distribution has been developed for a competitor card model and does not fully support the Prosum 155 Mbps cards. We don't discourage the use of this driver if it can fit your requirements, but of course we cannot provide any technical support concerning its usage with our cards.

Prosum provides the proatm driver, which is an adaptation of the nicstar driver. Be aware that this README does not apply to the former nicstar2 driver.

The proatm driver is compatible with kernel versions 2.6.xx and 3.0.xx. It supports the following features:

- all PCI and PCIe Prosum ATM card models,
- 16384 VC's
- vpi_base and vci_base parameters permitting to use any VPI/VCI
- UBR, CBR, ABR and VBR traffics,
- AAL5 and AAL0 (IDT AAL0 or raw cells),
- OAM cells (automatic responses sent by the driver)
- x86_64 architecture

We permanently work to improve the quality of this driver and to follow the evolutions of the Linux kernel, so there are frequent updates. We recommend you download the latest version of this driver from our web site. You can contact us at the address below for any problem, or to know the state of the pending version.

There are 2 ways for building and installing the proatm driver.

1. The fastest, simplest, and recommend method is straightforward: building and installing the module from an external folder. The only inconvenience of this method is the module must be built and installed each time you recompile your kernel or drivers. You cannot statically link the module with the kernel.

2. Patching the kernel source, configuring and recompiling the kernel and modules. This is a clean solution, but you need to recompile a lot of things and reboot the machine. This is rather complicated above all with proprietary kernels, so we recommend the first solution if you are not an expert or if you don’t want to lose time.

2. FAST INSTALLATION

Copy the files from the CDROM or decompress proatm-linux.tgz into a working folder. Open a root console into this folder and type:

```
# make install
```

Normally, this is enough for building and installing the proatm module. However there may be some missing files:

- if you get a message telling that kernel header files are not in any of the expected locations, into /usr/src, add a link named Linux and pointing to the source of your kernel. For example, suppose the kernel source is into /home/alex/linux, then type:

  ```
  # cd /usr/src
  # ln -s /home/alex/linux linux
  ```

- if you get a message telling "missing version header file" or "missing autoconf.h", open a console in the Linux kernel folder and type:

  ```
  # make prepare
  # make scripts
  ```

After "make install" has been successfully completed, the driver is installed into the current kernel. Load the module typing:

```
# modprobe proatm
```

and check that "lsmod" shows the proatm module.

That's it. You can skip sections 3 and 4.

3. BUILDING THE KERNEL AND MODULES

NOTE: Please see the Linux Kernel HOWTO if you are not familiar with building and installing a new kernel.

Follow the procedure below to add proatm to your kernel. We recommend you install proatm as loadable module and not directly linked with the kernel, so that you can easily unload and reload proatm.

If you have a recent PROATM CDROM, copy the proatm files from the CDROM folder"/drivers/Linux/V3xx" into the "drivers/atm" folder of the kernel source directory (usually /usr/src/linux). If you don't have a recent PROATM CDROM, decompress "proatm-linux.tgz" into Linux kernel folder "drivers/atm".

At the root of your kernel source or headers, open a console and try to retrieve the .config file of your current kernel by typing:

```
# make oldconfig (or "make cloneconfig" for SUSE)
```

Check that the .config file you got is somewhat compatible with the running kernel by comparing its first lines with the print of "uname -r".

Into the "drivers/atm" folder of the kernel source, open a console and type:

```
# patch < proatm.diff
```

Make sure that the package ncurses-dev or libncurses5-dev is installed, and at the root of the kernel source, open a console and type:

```
# make menuconfig (or "make xconfig")
```

Proceed to the kernel configuration as explained into section 4 "Configuring the kernel". Then type:

```
# make prepare
# make modules_prepare
# make modules && make modules_install
# make install
```

Reboot the machine, select the new kernel. Then in any console type:

```
# modprobe proatm
# lsmod"
```

You should see the proatm module into the printed list.

4. CONFIGURING THE KERNEL

In order to have access to the ATM features, enable "Prompt for development and/or incomplete code/drivers" in Code maturity level options (CONFIG_EXPERIMENTAL).

Enable the options as indicated below.

Into “Networking support” ---> “Networking options”, enable

"Asynchronous Transfer Mode" (ATM) (EXPERIMENTAL) (CONFIG_ATM). Select CLIP (recommended), LANE, and MPOA according to your needs.
Into “Device Drivers” ---> “Network device support” ---> “ATM drivers”, if the proatm patch as been applied successfully, you should see

"Prosum PROATM-V155 and PROATM-E155". Select M (recommended) or * for this option.
We recommend you disable all other ATM drivers if you don't have the corresponding adapters, especially idt77252 to prevent any conflict with proatm.

5. MODULE COMPILATION OPTIONS

GENERAL_DEBUG in proatm.c allows printing traces for debug purpose.

EXTRA_DEBUG in proatm.c prints even more traces. Don't forget to "undef" these options and recompile as soon as the problem is fixed.

Look for section "Options" in proatm.h.

NS_VPIBITS

Defines the number of bits used to code the VP number.

Default value is 2. Authorized values are 0, 1, 2, or 8. Refer to the table below giving the VPI and VCI ranges according to the card memory size and the value of PROATM_VPI_BITS:

PROATM_VPIBITS	VPI	VCI (128KB)	VCI (512KB)	VCI (2MB)
0	0	0 to 1023	0 to 4095	0 to 16383
1	0, 1	0 to 511	0 to 2047	0 to 8191
2	0 to 3	0 to 255	0 to 1023	0 to 4097
8	0 to 255	0 to 3	0 to 15	0 to 63

Note: The cards equipped with 128KB memory can support 1024 VCs. The cards equipped with 512KB memory can support 4096 VCs. The new cards equipped with 2MB memory can support 16384 VCs.

PROATM_VPIBASE and PROATM_VCIBASE
You can normally transmit and receive on VCs with VPI/VCI values depending on PROATM_VPIBITS and the card memory size. (Refer to the table above) However in certain special cases you may need to use normally-unreachable VPI/VCI couples such as 8.35 for example. To do that, set PROATM_VCIBASE and PROATM_VPIBASE according to your needs. For example setting PROATM_VPIBASE to 8 and PROATM_VCIBASE to 0 would allow you to play with the 8.35 VC.

WARNING: Please be aware that VPIs below PROATM_VPIBASE and VCIs below PROATM_VCIBASE are not reachable anymore when you set them to non-zero values.

PROATM_VPIBASE is smaller than 256. It must be a multiple (0 included) of the number of VPIs, which depends on PROATM_VPIBITS. For example, if PROATM_VPIBITS = 2:

- the number of VPIs is limited to 4,

- PROATM_VPIBASE can be set to 0, 4, 8, 12 ... 254.

PROATM_VCIBASE plays for the VCs the same role as PROATM_VPIBASE for the VPs. It is smaller than 64536 and must be a multiple of the number of VC's that the card is able to manage.

For example let's suppose the card has 128 KB memory and PROATM_VPIBITS = 2, the number of VCI bits is 8 (10-2) and thus the number of VCIs is 256, then PROATM_VPIBASE may be set to any value from among 0, 256, 512 ... 64280.

PROATM_RCQ_SUPPORT
Enable RAW CELL receipt. You can implement and add your own raw-cell process into ns_rsqe_process().

PROATM_BUILD_AAL0_HEADER
Specify that the header of AAL0 cells is to be built automatically with regard to internal connection information.

When this option is NOT set (default), according to the Linux ATM API, the header is simply extracted from the first 4 bytes of each SKB data. The header is under the application responsibility. Please note that incorrect headers may hang the driver and the computer.

When this option is set, the header of AAL0 cells is built by the driver the same way as used for the header of AAL5 cells. The first 4 bytes of packets are ignored except the GFC and PTI fields.

PROATM_AAL0_TO_RCQ
Enable all incoming AAL0 cells to produce a receive interrupt regardless of the value of the PTI bit. Otherwise only incoming cells with PTI = 1 can trigger a receive interrupt.

PROATM_PRINT_RAW_CELL_MESSAGE
Enable raw cell warning message. PROATM_TST_RESERVED
Number of Schedule Table entries (over 2048) that are reserved for non-CBR VCs (UBR/VBR/ABR)

PROATM_B1BUFSIZE
Specify the size input buffers. Don't modify these options unless you know exactly why.

PROATM_SUNI_FULL_SDH
By default, the physical interface of PROATM-155 cards implements OC3 and partial SDH. This works in most cases. Set this option to #define to make the physical interface implement full SDH - no more OC3 compatible.

PROATM_OAM_SUPPORT
Enable support of F4 and F5, end-to-end and segment OAM cells

PROATM_LLID

If you need it, you can write here the 16 bytes of the specific "Loopback Location ID". Otherwise don't touch this option.

PROATM_USE_WORKQUEUE

Use a work queue to transmit SKBs to upper ATM network layers outside the interrupt context. This option degrades the ping response-time but improves the average throughput of small packets. It may also prevent some hardlock problems on some architectures. By default this option is NOT enabled.

6. MODULES PARAMETERS

vpibits, vpibase and vcibase can be assigned at load time by insmod or modprobe. When specified, these parameters overwrite the default values set by PROATM_VPIBITS, PROATM_VPIBASE, and PROATM_VCIBASE.

Example: modprobe proatm vpibits=1 vpibase=16 vcibase=512

SDH framing can be forced or cleared by using fullsdh. If used, this parameter overwrites the PROATM_SUNI_FULL_SDH compilation option.

Example: modprobe proatm fullsdh=1

7. ATM ON LINUX

Install the linux-ATM package from your distribution or from linux-ATM.sourceforge.net. Do not hesitate to install the lib-atm and atm-devel packages.

Please, carefully read the HOWTO in /usr/share/doc/linux-atm-x.x.x/doc/.

You can also find a linux-ATM package on Prosum web site. It is basically the same as which one you can find on linux-ATM.sourceforge.net but it includes a few options and tools we added for our own needs or for some customers. It also uses to hang less easily that the distribution or official packages, especially if the proatm card has been wrongly automatically mounted by the system at boot time.

8. USING PVC'S

If you are NOT using an ATM switch, i.e., your PCs are connected 'back-to-back', you can base your configuration on examples below to quickly start a CLIP connection on PVC 0.0.32 (UBR).

Note: Without an ATM switch, some ATM features such as LAN Emulation and SVCs are not available. Therefore, only classical IP over ATM, remote PPP Services and PVC functions are available.

script atmstart on computer1:
#!/bin/sh modprobe proatm atmarpd -b
atmarp -c atm0
ifconfig atm0 192.168.1.1 sleep 1
atmarp -s 192.168.1.2 0.0.32 script atmstart on computer2:
#!/bin/sh
modprobe proatm atmarpd -b
atmarp -c atm0
ifconfig atm0 192.168.1.2 sleep 1
atmarp -s 192.168.1.1 0.0.32

Run atmstart on each computer, then you should be able to run any IP application such as FTP or NFS. Start by "pinging" each computer from the other one.

Note that the last two lines are not mandatory. You may type as many "atmarp -s" commands by hand as you need. The scripts build a UBR connection with unspecified PCR. You could type for example something more complex like:

atmarp -s 192.168.2.1 0.0.32 qos ubr:pcr=20Mbps or
atmarp -s 192.168.2.1 0.0.32 qos ubr:pcr=1Mbps

To change the QoS of a connection, first delete it by typing

atmarp -d 192.168.2.1

Refer to atmarp(8) and qos(7) man pages to get more explanation.

9. NOTE ON VBR USAGE

VBR QoS is characterized by the pcr, scr and mbs parameters. So far the

atm_trafprm structure does not provide neither scr nor mbs. We could modify atm_traf_prm however we found simpler to reuse existing parameters for ABR. This makes the job without impacting other ATM layers and drivers.

So when using VBR VC's, enter:

- the pcr value into max_pcr,
- the scr value into pcr,
- the mbs value into min_pcr.

This VBR implementation is intended for writing atm applications that manage the qos structure directly since the Linux-atm tools do not provide direct support of VBR. You may also add the VBR qos support into qos2text.c and text2qos.c and recompile linux-atm. A version of linux-atm supporting VBR is available on our web site.

10. OAM CELLS

The proatm driver can automatically process F4 and F5 ATM fault management, loopback, and continuity check cells. It behaves like an endpoint and automatically generates the appropriate response. It also permits to send OAM cells from applications running in the user space. It does not implement send_oam () but instead detects the AAL0 raw-cells that match the VCI/PTI OAM characteristics.

The CRC10 is automatically generated and put at the end of OAM cells before the transmission. So there is no need to compute it into the application.

Please be aware that OAM cells can only be received and sent on VCs that are open for reception and transmission.

The linux-atm package on our web site contains an atmdump tool that has been slightly modified to permit an easy generation of any raw cell in hexadecimal. It can be used to generate all types of OAM cells. It also provides aloop, a test that allows evaluating the performance of the card on your machine. There is a compilation option into "proatm.h" that permits to enable/disable the support of OAM cells. You can also define the Loopback Location ID of the ATM interface. However, please note that this LLID applies to all PROATM cards into the computer.

11. TECHNICAL AND SALES SUPPORT

Technical support: Send E-mails to support@prosum.net

Sales and information: (33) 1 45 90 62 70 or E-mail contact@prosum.net