



# PROATM ADAPTERS ON LINUX

## *Quick Installation Guide*

Version 2.0

8/23/2011

# 1. INTRODUCTION

The PROATM-V155 and PROATM-E155 adapters are based on the IDT77252 SAR. They are not supported by the nicstar driver included in drivers/atm folder of the Linux kernel. The original nicstar driver can only support older SARs such as 77201 and 77211.

The IDT77252 (nicstar II) driver that is provided by the 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 regarding its usage with our cards.

Prosum provides the Nicstar2 driver, which is an adaptation of the original nicstar for the PROATM cards. Nicstar2 is compatible with kernel versions 2.4.xx and 2.6.xx. It supports the following features:

- All 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 AALO (IDT AALO or raw cells),
- OAM cells (automatic responses sent by the driver)
- x86\_64 architecture

We work permanently 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.

## 2. BUILDING THE KERNEL AND MODULES

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

Follow the below procedure to add Nicstar2 to your kernel. We recommend you install nicstar as loadable module and not directly linked with the kernel so that you can easily unload and reload nicstar.

If you have the PROATM CDROM, copy the source files from the CDROM folder

/drivers/Linux/Vxxx into /usr/src/linux/drivers/atm, replacing the source files of the genuine nicstar driver.

If you don't have the CDROM, download and uncompress nicstar2.tar.gz and copy the source files to /usr/src/linux/drivers/atm, replacing the source files of the genuine nicstar driver.

If you are compiling the driver for a 2.6 kernel, apply the prosum patch file. To do that, proceed as following:

If your kernel version is 2.6.21 or previous, check that "prosum.diff" is standing into /usr/src/linux/drivers/atm, open a console into this folder and type:

```
# patch < prosum.diff
```

If your kernel version is 2.6.22 or following, do the same with the corresponding prosum.diff-2.6.xx file.

**NOTE:** The nicstar.o module needs the suni.o module to run properly.

Do the usual make menuconfig or make xconfig in /usr/src/linux. First, enable "Prompt for development and/or incomplete code/drivers" in Code maturity level options (CONFIG\_EXPERIMENTAL). Enable the options as indicated below.

Networking options:

1. Enable "Asynchronous Transfer Mode" (ATM) (EXPERIMENTAL) (CONFIG\_ATM)
2. Select module for "Classical IP over ATM" (CONFIG\_ATM\_CLIP)
3. Enable "Do NOT send ICMP if no neighbour" (CONFIG\_ATM\_CLIP\_NO\_ICMP)
4. Select module for "LAN Emulation (LANE) support" (CONFIG\_ATM\_LANE)
5. Select module for "Multi-Protocol Over ATM (MPOA)" (CONFIG\_ATM\_MPOA)

### **Vanilla kernel**

If you are using a standard kernel from kernel.org (Vanilla), continue as following:

ATM drivers:

1. For a 2.4.x kernel, select module for "IDT 77201 (NICSTAR)" (CONFIG\_ATM\_NICSTAR)

For a 2.6.x kernel, if "prosum.diff" or "prosum.diff-2.6.22" has been applied correctly, you should see the option "Prosum PROATM-V2".

2. Check "Use suni PHY driver (155Mbps)" (CONFIG\_ATM\_NICSTAR\_USE\_SUNI)

3. We recommend you disable all other ATM drivers if you don't have the corresponding adapters, especially idt77252 to prevent any conflict.

Then build and install your kernel and modules and reboot.

To install a new nicstar2 version, replace the source code by the new one and type:

```
# make modules && make modules_install
```

It takes less than one minute. Then reboot or simply type:

```
# modprobe -r nicstar
```

```
# modprobe nicstar.
```

## *Distribution kernels*

Most of time you will use a kernel from a distribution such as Red Hat, Fedora, Ubuntu, Mandriva, Suse, CentOS, etc. The installation process is a little bit less easy because each distribution has particularities. We don't recommend building the complete kernel in this case. Here are some tips allowing building and installing the nicstar module only.

**NOTE:** Some distributions do not actually provide the sources. In this case you only get a directory containing the kernel headers. Into this directory, the folder "drivers/atm" only contains Kconfig and Makefile. You will be missing suni.h, to compile the module. If you cannot get the official kernel sources, do not hesitate to take this file from the vanilla kernel of same version (kernel.org), and copy it into drivers/atm.

Do not type "make modules && make modules\_install". You will just compile the nicstar module and install it into your current kernel. To do so, proceed as following:

Into the root of the directory of your kernel headers retrieve the .config file of your current kernel by typing:

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

To prepare the compilation, type:

```
# make prepare
```

```
# make modules_prepare
```

If some files are still missing, just add them from your kernel source or from the vanilla kernel source of same version.

Apply the prosum patch into drivers/atm as described above.

Make sure that ncurses-dev or libncurses5-dev package is installed, and type:

```
# make menuconfig
```

Into the ATM drivers configuration (Device drivers->Network device support->ATM drivers), remove all ATM device drivers except nicstar and suni in order to prevent any driver conflict at startup.

Into menuconfig, select M for nicstar and Y for suni.

Exit and save the new .config configuration file.

Then compile and install the nicstar and suni modules by typing:

```
# make modules M=drivers/atm
```

You should only see the compilation of nicstar and suni.

Then install the new modules by typing:

```
# make M=drivers/atm KERNELRELEASE=`uname -r` modules_install
```

```
# depmod -a
```

Check that nicstar and suni have been copied into `/lib/modules/your_kernel_name/extra` or into `/lib/modules/your_kernel_name/kernel/drivers/atm`. Then remove all other ATM modules from `/lib/modules/your_kernel_name/kernel/drivers/atm` to prevent start up conflicts, especially with `idt77252`.

Check that nicstar is correctly installed by typing:

```
# modprobe nicstar
```

```
# lsmod
```

You should see nicstar and suni used by nicstar into the list.

## 3. MODULE COMPILATION OPTIONS

**GENERAL\_DEBUG** in `nicstar.c` allows printing traces for debug purpose.

**EXTRA\_DEBUG** in `nicstar.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 `NS_VPI_BITS`:

<i>NS_VPIBITS</i>	<i>VPI</i>	<i>VCI (128KB)</i>	<i>VCI (512KB)</i>	<i>VCI (2MB)</i>
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.

### NS\_VPIBASE and NS\_VCIBASE

You can normally transmit and receive on VCs with VPI/VCI values depending on `NS_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 `NS_VCIBASE` and `NS_VPIBASE` according to your needs. For example setting `NS_VPIBASE` to 8 and `NS_VCIBASE` to 0 would allow you to play with the 8.35 VC.

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

`NS_VPIBASE` is smaller than 256. It must be a multiple (0 included) of the number of VPIs, which depends on `NS_VPIBITS`.

For example, if `NS_VPIBITS = 2`:

- the number of VPIs is limited to 4,
- `NS_VPIBASE` can be set to 0, 4, 8, 12 ... 254.

`NS_VCIBASE` plays for the VCs the same role as `NS_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 NS\_VPIBITS = 2, the number of VCI bits is 8 (10-2) and thus the number of VCIs is 256, then NS\_VPIBASE may be set to any value from among 0, 256, 512 ... 64280.

### **NS\_RCQ\_SUPPORT**

Enable RAW CELL receipt. You can implement and add your own raw-cell process into ns\_rsqe\_process ().

### **NS\_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.

### **NS\_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.

### **NS\_PRINT\_RAW\_CELL\_MESSAGE**

Enable raw cell warning message.

### **NS\_TST\_RESERVED**

Number of Schedule Table entries (over 2048) that are reserved for non-CBR VCs (UBR/VBR/ABR)

### **NS\_BOBUFSIZE and NS\_B1BUFSIZE**

Specify the size of small buffers and large buffers. Don't modify these options unless you know exactly why.

### **NS\_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.

### **NS\_OAM\_SUPPORT**

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

### **NS\_LLID**

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

## 4. MODULE 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 NS\_VPIBITS, NS\_VPIBASE, and NS\_VCIBASE.

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

SDH framing can be forced or cleared by using fullsdh. If used, this parameter overwrites the NS\_SUNI\_FULL\_SDH compilation option.

Example: modprobe nicstar fullsdh=1

## 5. ATM ON LINUX

Install the linux-ATM package from your distribution or from [linux-ATM.sourceforge.net](http://linux-ATM.sourceforge.net). Do not hesitate to install the lib-atm and atm-devel packages.

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

## 6. 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 nicstar
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 nicstar
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.

## 7. 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.

## 8. OAM CELLS

Nicstar2 can process automatically F4 and F5 ATM fault management, loopback, and continuity check cells. It behaves like an endpoint and automatically generates the appropriate response. It permits also 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 a loop, a test that allows evaluating the performance of the card on your machine.

There is a compilation option into "nicstar.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.

## 9. TECHNICAL AND SALES SUPPORT

Technical support: Send E-mail to [support@prosum.net](mailto:support@prosum.net)

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