ATM on Linux

Werner Almesberger
werner.almesberger@lrc.di.epfl.ch

May 4, 1995

The Laboratoire de Reseaux de Communication of EPFL is developing ATM support for Linux, with the goal of creating a flexible and powerful ATM platform for research and education. This is a quick introduction to ATM and an overview of our first results.

1 ATM characteristics

The Asynchronous Transfer Mode (ATM, also known as Broadband-ISDN) is a highly flexible network technology offering a large range of available bandwidths (e.g. 25 Mbps, 155 Mbps, 622 Mbps), support for real-time traffic, and seamless interconnection between WANs and LANs.

ATM endsystems can request a guaranteed quality of service (QOS), which is essential for applications like video on demand (VOD), teleconferencing, etc. Because ATM receives support from both the telecommunication and the data networks communities, we can expect ATM to become the most important network technology in the near future.

ATM is connection-oriented and uses cell switching. Data is transmitted in tiny cells of 53 bytes (5 bytes header, 48 bytes payload), which can easily be switched in hardware. Since datagrams are usually larger than 48 bytes, they have to be segmented into cells at the sender side and reassembled at the receiver side. The protocols used for segmentation and reassembly are called ATM Adaption Layers (AALs), of which the simple, data-oriented AAL5 is the most popular one.

2 Adapter and driver design

Modern ATM adapters perform AAL5 processing in hardware, they usually support hundreds if not thousands of simultaneous connections, and they also perform traffic shaping. ATM adapters for PCs are typically EISA or PCI devices, use speeds of 100 or 155 Mbps, and support fiber or UTP5 cabling.

One common adapter design buffers PDUs in adapter memory and performs transfers to or from host memory with bus-master DMA. Another common adapter design only buffers a few cells and directly transmits from, or receives to, buffer chains in host memory.

A typical ATM device driver primarily has to manage memory used for sending and receiving, control ATM-specific resources (e.g. connection identifiers, bandwidth), and, of course, enqueue and dequeue packets coming from or going to upper protocol stack layers.

The protocol layer on top of the ATM device driver can be an ATM-specific transport protocol or it can implement some multiplexing for higher layer protocols (e.g. IP, ARP, IPX). The current implementation provides a rudimentary unreliable ATM-specific protocol and support for RFC1483-compliant multiprotocol encapsulation (e.g. for IP over ATM).

3 Performance hacks

The current Linux networking code copies data at least twice: between the network adapter and kernel buffers, and between kernel buffers and user memory. These two copy operations limit throughput to about 24 MB/sec on today's fastest PCs, 13 MB/sec on what used to be the fastest PCs three months ago.

Considering that 155 Mbps ATM transfers data at more than 15 MB/sec in each direction, it is clear that, at such speeds, better throughput is needed in order to be able to feed the network or to process incoming data.

A possible solution for this problem is to avoid copying between kernel and user space. Instead, when sending, memory pages are locked in physical memory and physical memory addresses are passed to the device driver, which then sends directly from user memory. When receiving, kernel receive buffers are aligned to page boundaries and the read system call directly maps kernel pages into user space, thereby avoiding to touch their contents.

With this approach, throughput can be increased to about 50 MB/sec.

4 Status

A first pre-alpha version of ATM support for Linux has been released at the beginning of May 95. It includes a "driver" for ATM over TCP, a driver for the Efficient Networks ENI155P-MF board, a simple raw ATM API, and some experimental support for IP over ATM. Here are a few performance measurement results:

Transport	Throughput
TCP	38 Mbps
checksummed UDP	51 Mbps
unchecksummed UDP	72 Mbps
raw AAL5, no MMU hacks	97 Mbps
raw AAL5, with MMU hacks	129 Mbps

All these tests were done on directly connected Pentium 90 MHz PCs, with Neptune PCI mainboards, 32 MB RAM, and ENI155P-MF-C adapters. Datagram size was always 8192 bytes. The theoretical throughput limit is 135 Mbps.

Future work will include further performance enhancements, design and implementation of a Linux ATM API, and fully RFC1577-compliant IP over ATM support.

5 Resources

There is a mailing list for ATM on Linux. You can subscribe by sending a mail with the body

```
subscribe linux-atm
```

to `majordomo@vger.rutgers.edu`. The mailing list is archived in `lrcftp.epfl.ch:/pub/atm/linux-atm/`

The current ATM implementation for Linux can be found in `lrcftp.epfl.ch:/pub/linux/atm/`

The API draft is in `lrcftp.epfl.ch:/pub/people/almesber/atm/`