This is the third part of Chapter 10 from the second edition of :

Networks, Routers and Transputers:

Function, Performance and applications

Edited by: M.D. May, P.W. Thompson, and P.H. Welch

© INMOS Limited 1993

This edition has been made available electronically so that it may be freely copied and distributed. Permission to modify the text or to use excerpts must be obtained from INMOS Limited. Copies of this edition may not be sold. A hardbound book edition may be obtained from IOS Press:

IOS Press
Van Diemenstraat 94
1013 CN Amsterdam
Netherlands

IOS Press, Inc.
P.O. Box 10558
Burke, VA 22009–0558
U.S.A.

IOS Press/Lavis Marketing
73 Lime Walk
Headington
Oxford OX3 7AD
England

Kaigai Publications, Ltd.
21 Kanda Tsukasa–Cho 2–Chome
Chiyoda–Ka
Tokyo 101
Japan

This chapter was written by C. Barnaby and N. Richards.

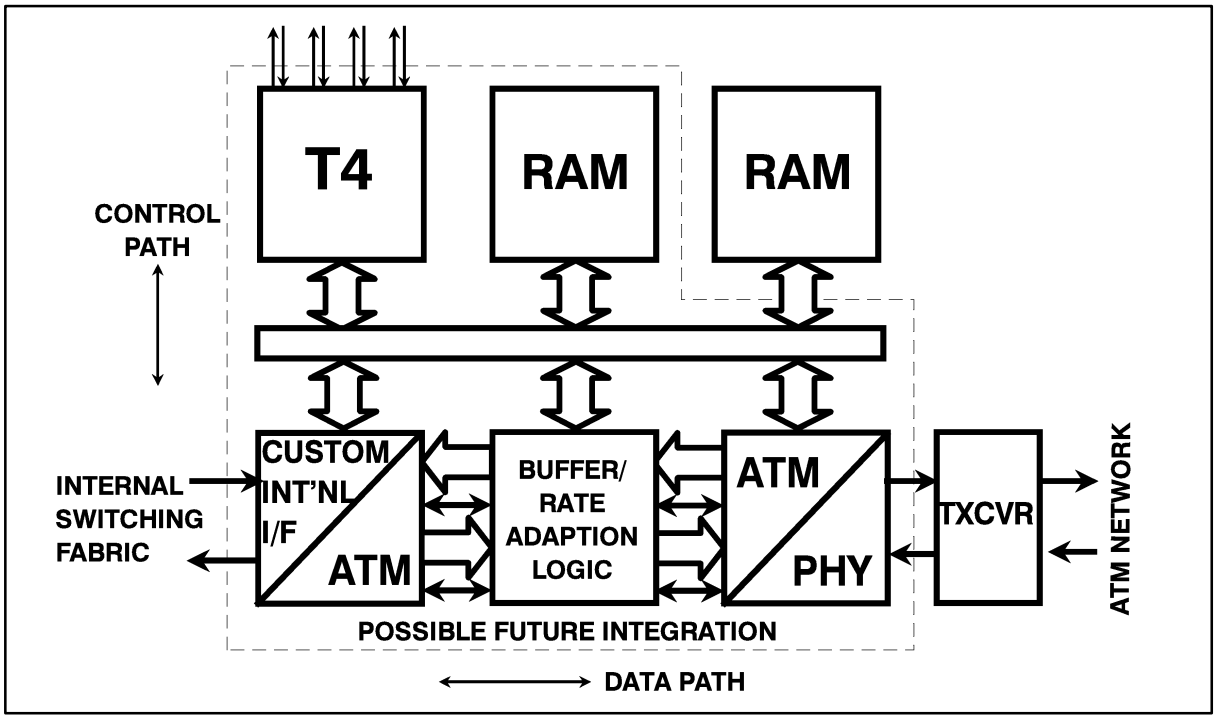

Figure 10.18 Possible ATM 'Line Card'

Such a line card is essentially a uniprocessor application, so the use of the transputer serial links for multiprocessing is not required. However, the serial links are very useful in other ways; for program download and debugging, test and diagnostics.

Putting software in ROM on the line card is undesirable from an upgrade and maintenance point of view. It would be better to be able to download code from some central point within the exchange. This could be achieved either by sending code via the switch fabric (possibly using a small boot ROM for cold–starts only) or by sending it down the transputer serial links (performing cold starts via the ***boot–from–link*** capability).

If the serial links are brought to the edge of the line card they can be used for testing in one of two ways. First, they can be used as part of the production test of the card by integrating them with an ATE system. Test code can be downloaded into the transputer (via the links) which runs entirely in the internal RAM. This code can exercise, at full speed, the external interfaces of the transputer as part of the test functions of the ATE system. Secondly, if the serial links are accessible while the card is in service in the exchange, it is a useful 'entry point' for a test engineer to interrogate the system. Better yet, if the serial links are internally interconnected, the switch control computer itself can use them to interrogate the system.

Switching Fabric

In a large public switch the data rates and requirements of the switching fabric are such that it is most likely to be built out of dedicated hardware and will in itself be a very complex subsystem. It is **not** appropriate to consider the use of the C104 for this fabric directly, nor to consider that the (non–maintenance) ATM traffic could be carried via transputers. However, like the network interfaces, there is considerable benefit in embedding processors within the hardware to provide intelligent control of the fabric. Maintenance and statistical measures can be provided, routing tables updated (if applicable) and the fabric monitored and reconfigured under fault or congestion conditions.

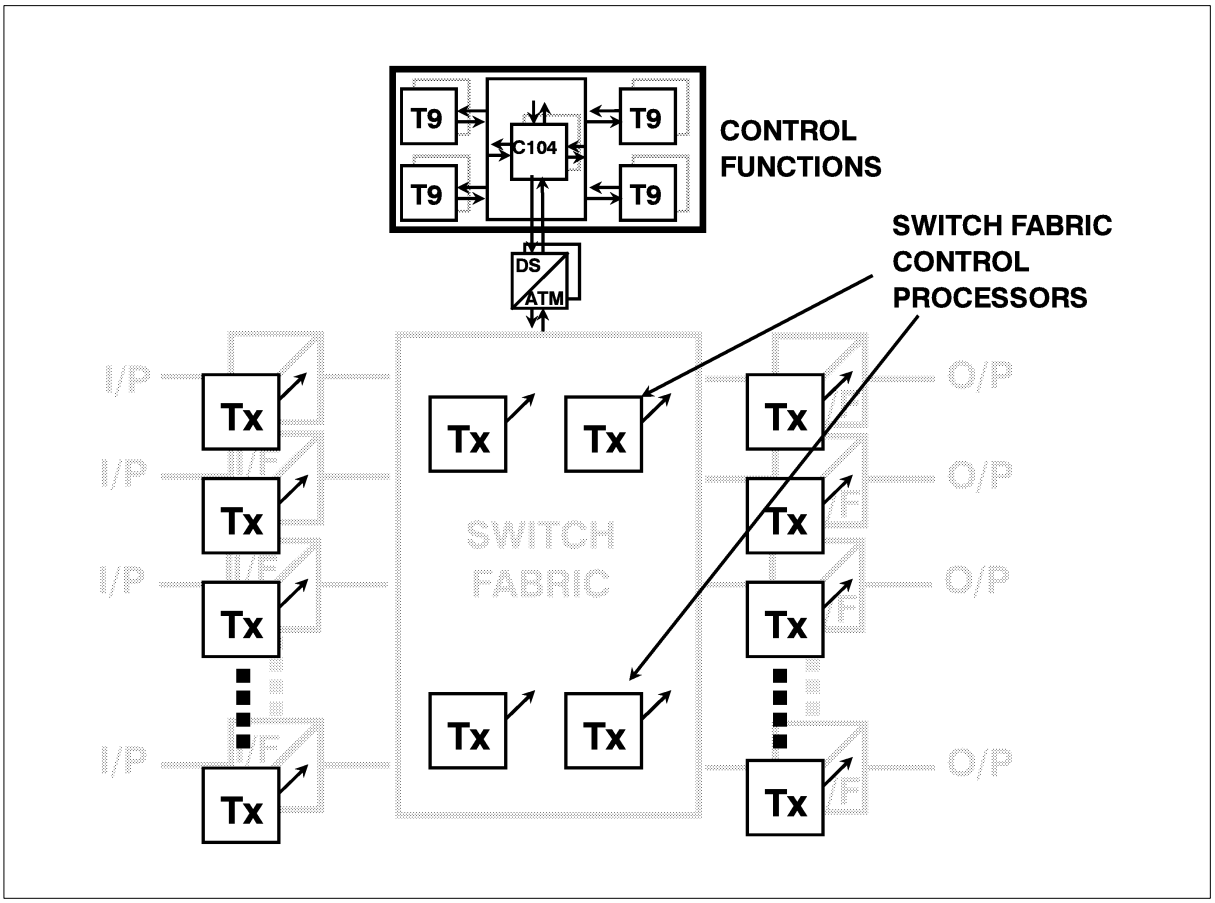

Figure 10.19 Embedded Switch Fabric Control

If desired, the links available from the control transputers can themselves be interconnected via a C104 network to provide a distributed control plane which is quite independent of the main ATM switch fabric, as illustrated in figure 10.20.

There are many other possibilities for mixed processor/hardware intelligent switching fabrics that remain to be investigated, and it is hoped that further ideas will be presented in future papers.

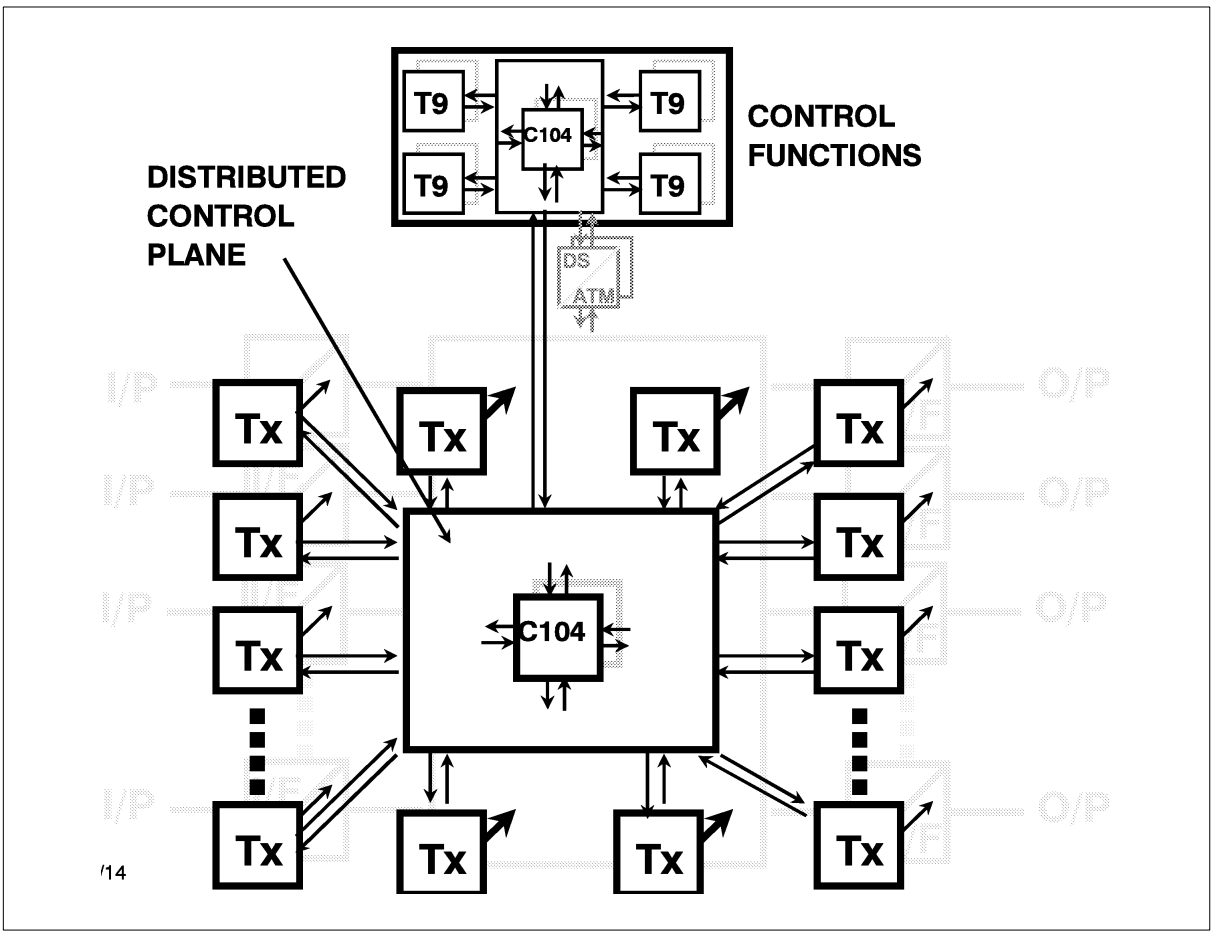

Figure 10.20 Distributed Control Plane

10.3.2 Private Switching Systems

All of the preceding discussion on public ATM switches also applies to private systems. However, there are some important differences:–

- the machines are not as large
- the bandwidth requirements are likely to be lower
- they are far more cost sensitive.

The nature of the ***Customer Premises Equipment (CPE)*** market is also likely to require much faster design cycles for the equipment, probably 1–2 years as the technology becomes established. The dynamics of the market are likely to place manufacturers under pressure to provide modular, flexible designs which can be upgraded, either in terms of performance, services or number of connections. Greater emphasis than in the past will be placed on network reliability, so the fault–tolerance aspects of the equipment will come under closer and closer scrutiny.

A Generic Private ATM Switch

The main difference from the point of view of applying the transputer architecture is that in private systems it is now possible to consider to use of the C104/DS-Link as the basis for an inexpensive switching fabric. Many current ‘campus’ ATM switches have been derived from existing bridge/router technology and are based on shared bus interconnect schemes. These do not provide scalable performance, as the common bus quickly becomes a bandwidth bottleneck. However, using the communications architecture of the transputer we can construct a scalable ***Generic ATM Switch*** for private applications [6].

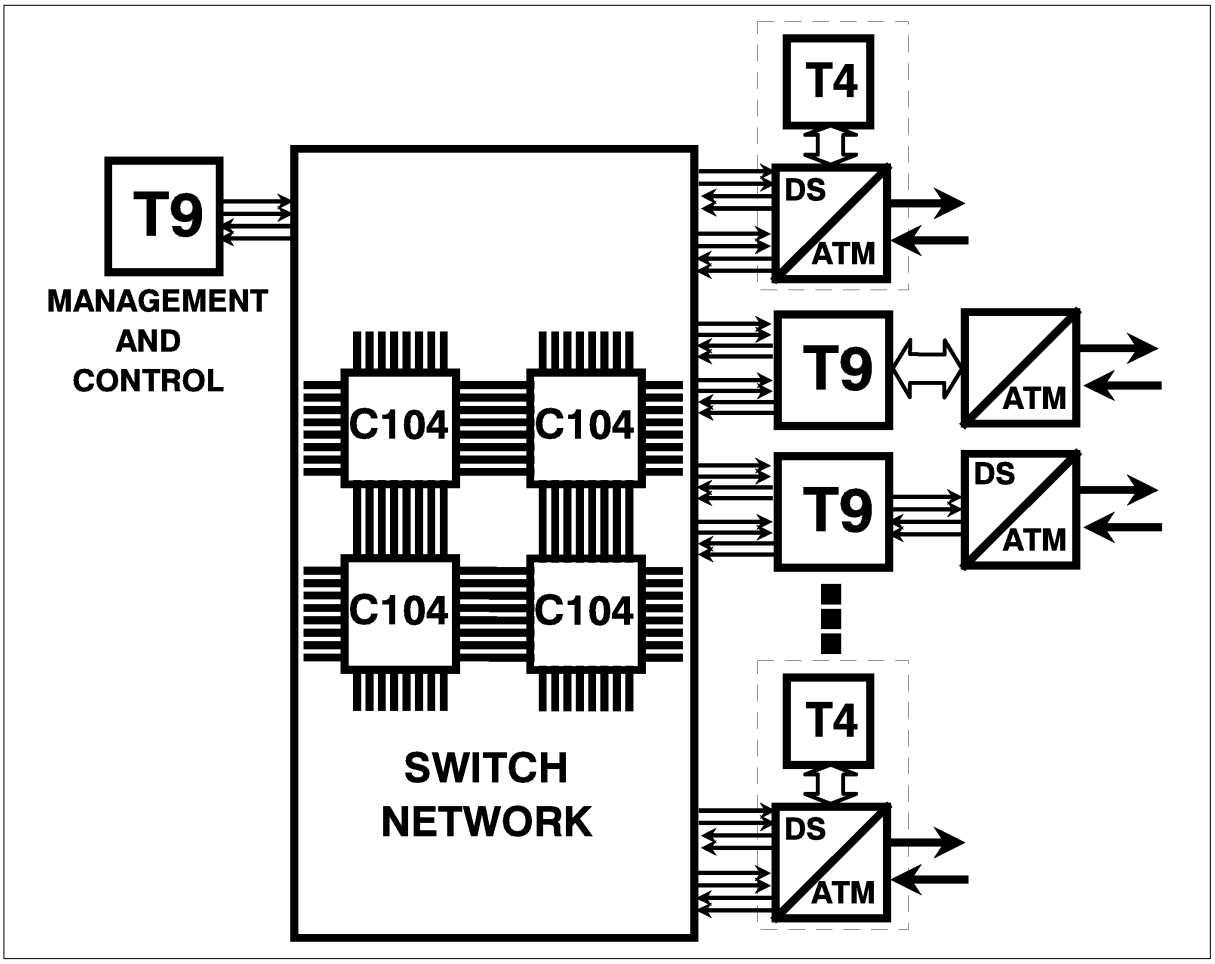

Figure 10.21 Generic Private ATM Switch

At its simplest, this switch may be considered to be a no more than a 'black box' multiprocessor computer running an ATM program. It has interfaces around the periphery to allow it to talk to the transmission network outside, but in essence it exploits the architectural similarity of message–passing/fast–packet–switching machines discussed earlier. Figure 10.21 shows illustrates several ways in which ATM interfaces can be built for such a switch, depending on cost/performance trade–offs required.

There are some important features of the DS-Link/C104 communications architecture which apply in its use as a fast packet switch:–

- DS-Links are ***cheap***
- The C104 can be used to build ***Scalable*** networks
- The in–built ***Flow Control*** mechanisms at the Token layer of the DS-Link protocol mean that the fabric is ***Lossless***, that is, no data packets/cells are ever lost internally due to buffer overflow within the fabric itself. Buffer dimensioning/overflow issues are moved outside the switch fabric to the network interfaces at the edge.
- DS-Links may be ***Grouped*** to provide high bandwidth connections within the fabric. This can be used to:–
  - minimize congestion for a given desired bandwidth
  - carry high–bandwidth traffic (for example, to 622 Mbit/s ATM).
  - provide redundant paths in the fabric for fault–tolerance reasons

- Link grouping on input can be used to avoid ***Head–Of–Line Blocking*** (congestion at the input to the switch fabric) by statistically increasing the chances of accessing the fabric (this is illustrated in the previous diagram)

- Universal (randomized) Routing can be used to avoid the 'hot–spot' congestion which can sometimes occur with certain systematic traffic patterns (for a full examination of this see Chapter 7).

- Traffic of ***any packet length*** may be carried by the C104 fabric. Only traffic intended directly for the (current) T9000 needs to be segmented into 32–byte packets, although longer packets may affect the congestion characteristics of the fabric.

Since the C104 fabric is simply an interconnect mechanism for a multi–processor computer, it is trivial to add further processors to this architecture to perform the Management and Control functions. As many as necessary can be attached to the switching fabric and they can communicate with the 'line cards' directly using the same fabric.

Multiple switching planes could also be used to provide either:–

- Separate control/data traffic planes

- Different planes to handle different traffic priorities

- Redundant Fault–tolerance within the overall switching fabric

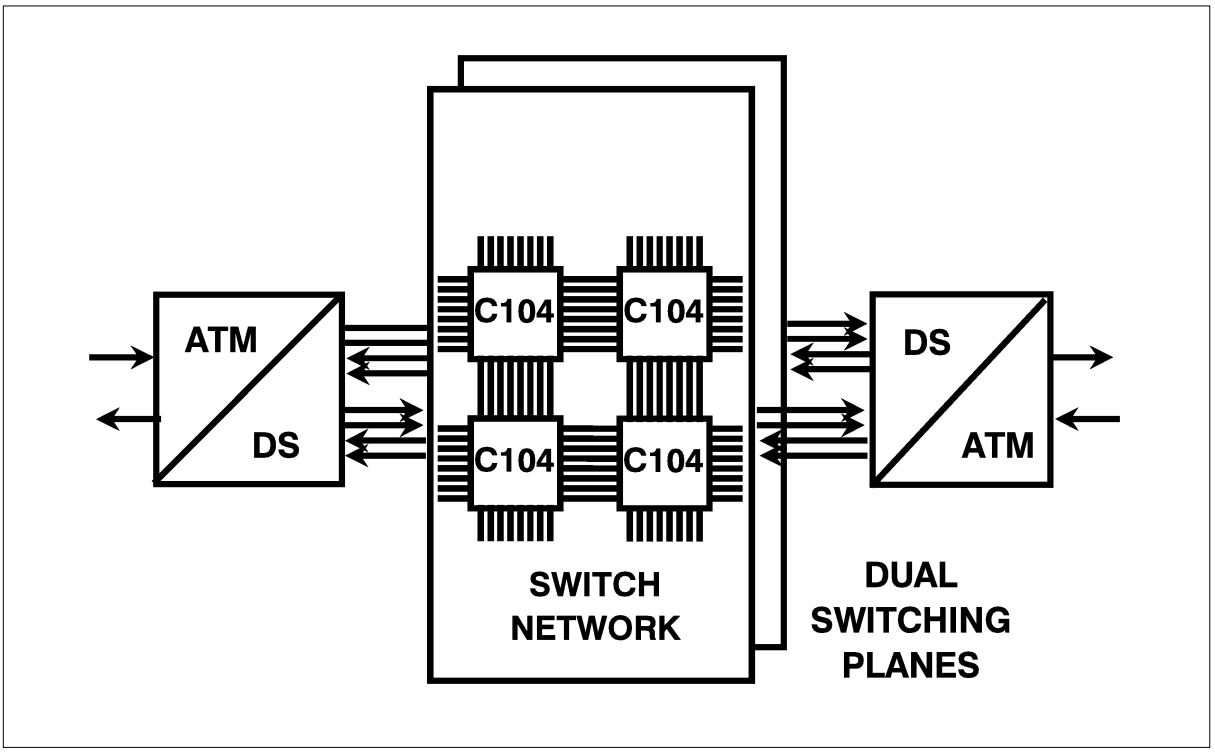

Figure 10.22 Multiple Switching Planes

Generic Internetworking Unit

One of the attractive aspects of this architecture is that interfaces to other networks, for example ethernet, token ring, FDDI, frame relay, etc., can be added very easily and so provide a ***Generic Internetworking*** architecture:–

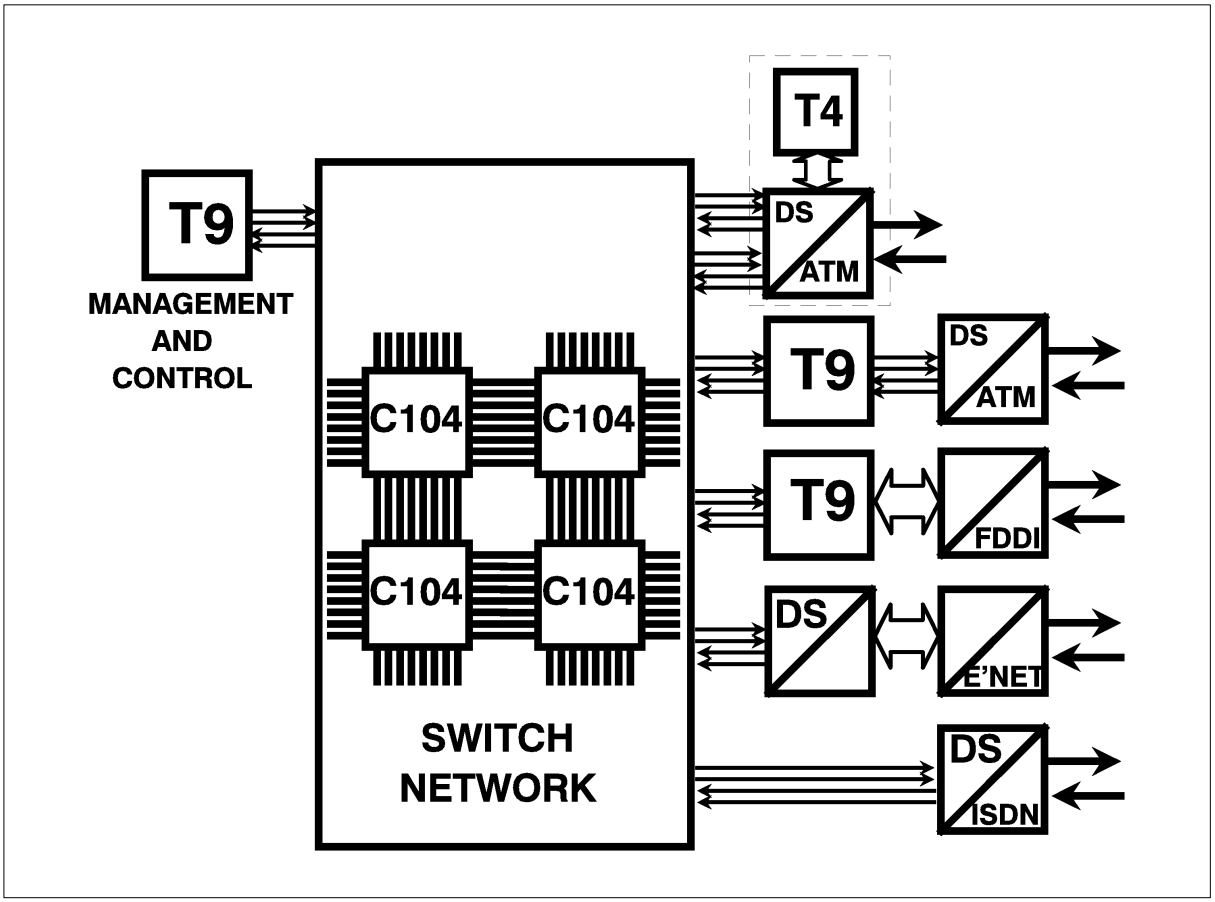

Figure 10.23 Generic Internetworking Architecture

Since we have simply built a computer (and one which is scalable in performance at that) we can add additional computing performance where required. A "pool" of processors can be added to this system to provide high–performance protocol processing between the various networks. Indeed, "***Parallel Protocol Processing***" techniques may be applied. For example, a 'farm' of T9000 processors may be made available to perform frame–by–frame AAL conversion from ethernet to ATM.

ATM Concentrator

We can extend the internal serial interconnect beyond the confines of our 'black box' ATM computer to provide a low–cost, lower speed entry point from an ATM terminal into the network, a sort of broadband serial concentrator. By using appropriate physical drivers, we can use the DS-Links directly to carry ATM cells asynchronously over local distances into the switch. Apart from cost advantages (since the DS-Links are inexpensive and the complication of full STM framing is not required) the DS-Links also provide an in–built ***flow–control mechanism*** which would provide an automatic means of 'throttling' the traffic flow back to the source. This is something which is currently missing from the ATM standards (GFC bits notwithstanding) and which could be added without requiring any alterations to the ATM standards by using the DS-Links. The availability of flow control to the source would considerably ease the buffering/performance design issues within the local switch as well as reducing the hardware/software costs associated with header policing on input.

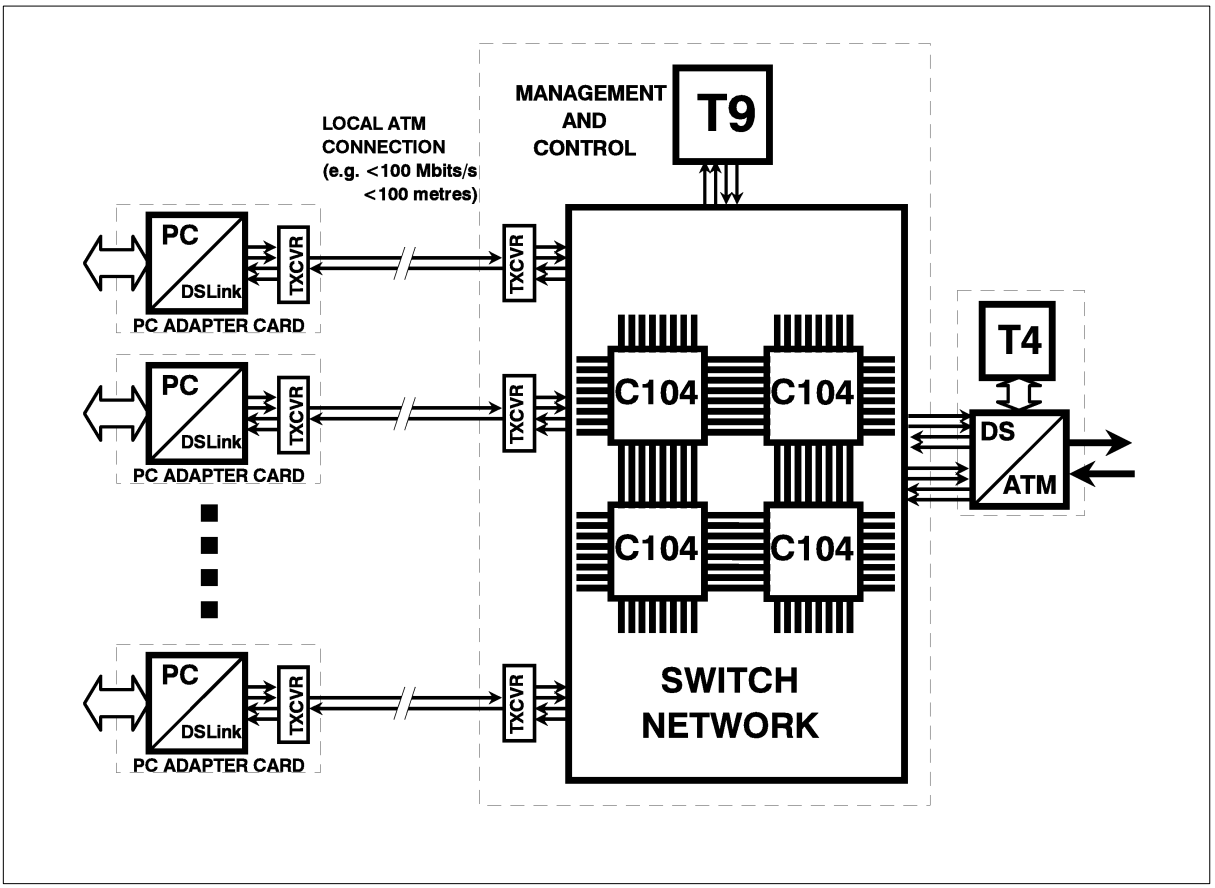

Figure 10.24 Low Cost ATM Concentrator

Issues and techniques for using DS-Links at a distance have been covered in Chapter 4 and such an interconnect could probably provide a very low cost entry–point into an ATM network for end user terminal equipment.

Private ATM Network Interface

The basic issue concerning the network interfaces for our private C104–based ATM switch is how to get ATM cells from the transmission system onto the DS-Links. Later in this Chapter a discussion is presented of the various ATM–DS-Link mappings that are possible and the performance issues that arise. Here, we consider the functional aspects of such interfacing for the moment.

The ATM line card must perform:–

1 **Rate adaption:**

- The need for rate adaption will vary depending on the speed and number of DS-Links provided at the line interface. In any case, some FIFO buffering will be needed to cope with slight rate mismatches caused by cell header processing, etc. More exotic methods may be added if the DS-Links are to run at a substantially different rate to the ATM line. Rate adaption between the DS-Link network and ATM can be provided by supporting one or more of the following:–
  - FIFO's to cope with traffic bursts
  - Inserting and deleting ATM 'Idle cells' (null cells for bandwidth padding) into a full–rate 155 Mbit/s ATM cell stream
  - Allowing the ATM clock rate to be varied (for example 1.5/2/34/45/155 Mbits/s. This may be allowable for private networks, but not on the public side).

2 ATM Cell Header Processing:

- HEC checking and generation for the ATM header
- Policing functions
- Header translation

3 Packetisation:

- Encapsulation of ATM cells into DS-Link packets for transmission via the DS-Links to the switching/processor network

4 STM/ATM Interfacing:

- Interfacing the ATM cell output stream to the synchronous, framed transmission system, where required on the public network. This will typically be done in hardware.

5 Management and Control:

- HEC error counts
- Policing parameters/algorithms
- Translation table updates, etc.

There is a hardware/software 'threshold' to be determined here which is the subject of further investigation. Some functions are obviously suited for hardware implementation, others for software. There is a grey area in between for functions such as policing and header translation, where the exact split between hardware and software could vary. A simple block diagram of a proposed network line card is given in the diagram below. The dotted line indicates where scope exists for a semi–custom integration of the card onto a single device in future.

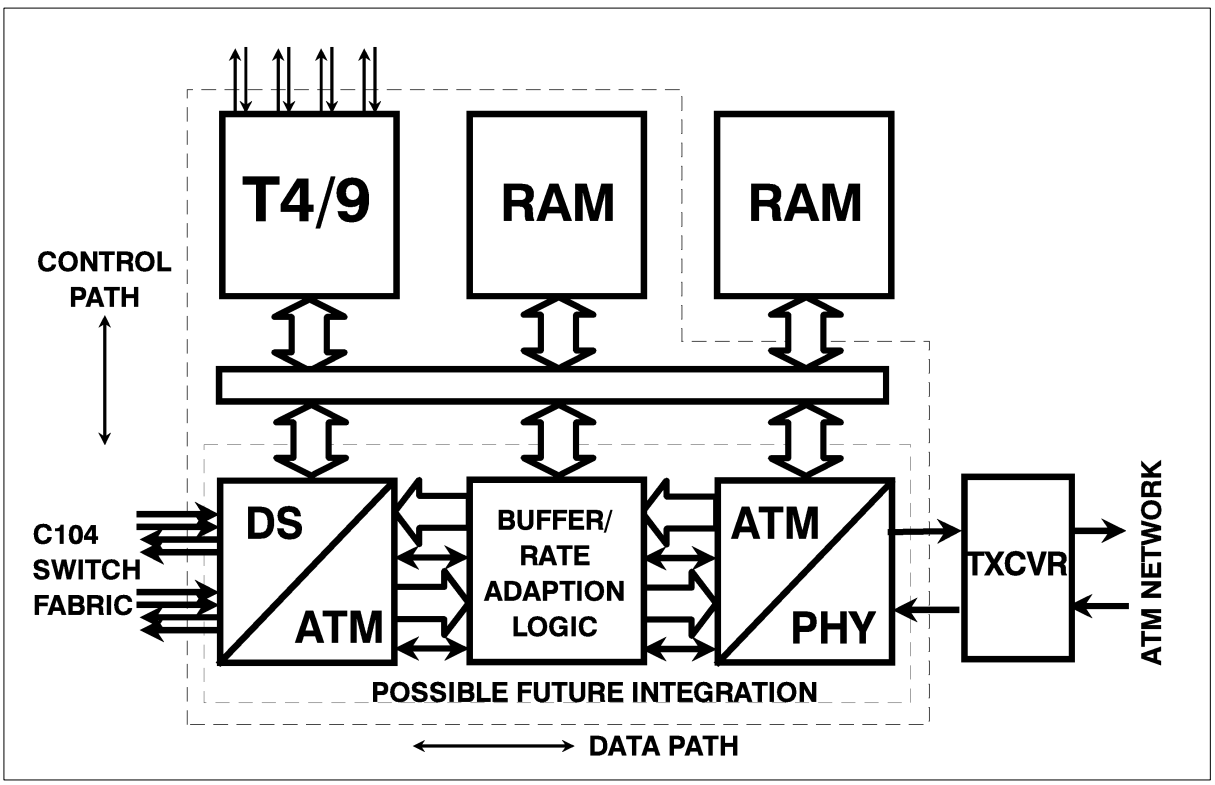

Figure 10.25 Simple ATM–DS-Link Network Interface Card.

10.3.3 ATM Terminal Adapters

Current PC's and workstations typically provide a fairly 'dumb' interface to a network in the form of a simple card to memory map an ethernet or token ring chip set into the hosts address space.

All interface control and higher layer protocol processing then falls on the host machine. It is becoming increasingly attractive to add a fairly powerful processor directly onto the network adapter cards in order to offload more of the protocol processing overhead from the host machine. As the bit–rate of the physical layer has increased in recent years, so the performance bottleneck in network access has moved to the higher layers of the protocol stack, which are more software/processor performance bound than the lower layers.

As 32–bit micro costs fall, we can apply many of the arguments for intelligent ATM line cards to an ATM Terminal Adapter and it becomes sensible to consider 'smart' rather than 'dumb' adapters. However, instead of providing an interface to a switching fabric (proprietary or DS-Link) we need a shared memory interface to one of the standard PC/workstation buses. A terminal adapter will also have to run one or more of the AAL standards and this is another reason for having a fast micro on the card – the AAL layer can be quite complex, the standards are changing and it may be necessary to run multiple AAL's to support, say, multimedia applications. This tends to mitigate against a hardware–only implementation and, like the line card, a hardware/software 'threshold' needs to be determined. Also, an ATM terminal adapter may not need to run at a sustained 155 Mbits/s rate, so it may be possible to sacrifice some performance in order to save cost by using software functions. In the end, the application requirements will decide.

A simple block diagram for a shared–memory PC Adapter card is shown below. A suitable ATM/PHY interface chip is assumed (these are now becoming available) and some appropriate system interfacing logic to load and store ATM cells in memory. Again, the dotted line shows the integration possibilities.

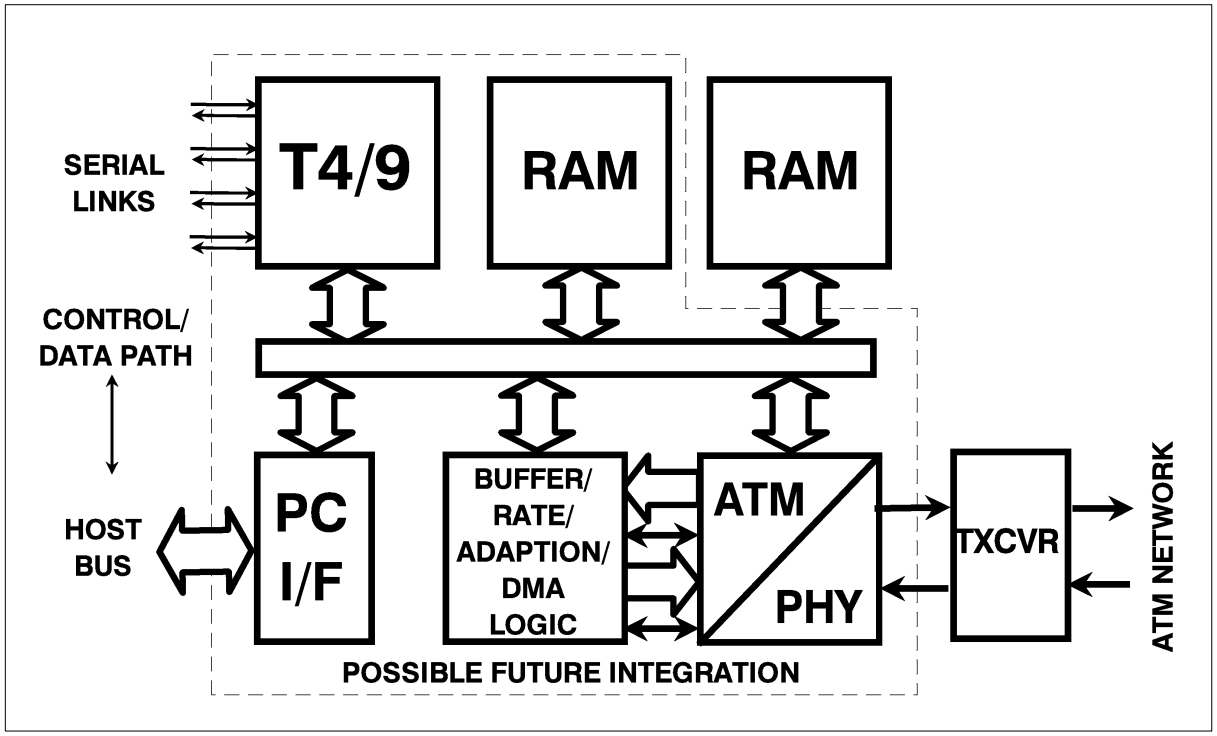

Figure 10.26 ATM–PC Terminal Adapter Card

In this example it is assumed that the AAL layer is handled in software by the transputer. A version of the AAL3 is currently being written for the transputer at INMOS in order to evaluate performance trade–offs and whether a software–only implementation is fast enough for modest applications. Details of this will form the basis of future papers.

An alternative form of Terminal Adapter can be envisaged for the control functions in a public or private switch. If a T9000 or multiple T9000's are being used for the control then it may be necessary to interface the DS-Links of the T9000 straight to ATM. A relatively simple ASIC

would be required in order to do this and which would perform the rate adaption, ATM cell timing, packetisation and HEC functions described above. All other functions could potentially be performed in software, since the Maintenance and Control cell rate is very low.

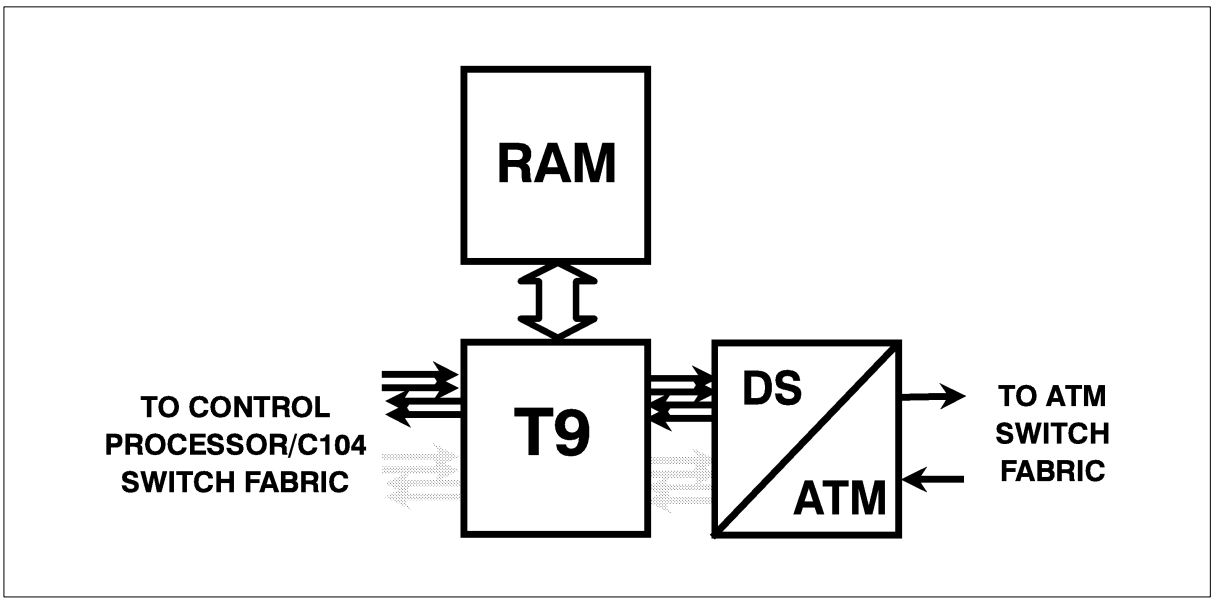

Figure 10.27 ATM–DS-Link Adapter Application

10.4 Mapping ATM onto DS-Links

In this section the issues associated with carrying ATM traffic over a DS-Link are considered. The DS-Link and the C104 do not require packets to be of a specified size, although the performance of the C104 chip has been optimized for use with small packets. This optimization is for parameters such as the amount of buffering on the chip and so variations in packet length will affect the blocking characteristics, although no packet data will ever be lost because the buffers cannot actually overflow. The current T9000 implementation, however, does place a constraint on packet length, presently of 32–bytes, and this means there are at least two ways of carrying ATM cells using DS–Links, depending on whether a T9000 is in the data path or not (this constraint could disappear in later T9000 versions if commercial issues justify a variant).

10.4.1 ATM on a DS–Link

In this section we consider the raw bandwidth the DS–Link can provide in order to carry ATM cells. We can consider 2 possible ways of using the DS–Links:–

- In a ‘T9000’ system with a full T9000 packet layer protocol implementation i.e. acknowledged packets of 32–byte maximum length
- In a ‘hardware’ system (built with no T9000’s in the data path) where the packet layer protocol implementation may be different, i.e. different packet length (and possibly without support for packet acknowledges).

A general performance model of the DS-Link is given in Chapter 6. This describes the data throughput of the DS-Link, given a specified message and packet size. It takes account of packet overheads, flow control and unidirectional and bidirectional use of the links. This basic model is extended here to show the throughput of the DS-Link carrying ATM cells, both with and without the full T9000 packet layer protocol. That is:–

- One ATM cell in single packet:–
  - One 53–byte packet on the DS–Link