Architecture of a Multimedia Desktop Workstation

G.J. Armitage & K. M. Adams

Dept. of Electrical & Electronic engineering
University of Melbourne
Parkville 3052

Phone : (03) 3447976
(03) 3447974
Fax: (03) 3446678
Acsnet: gja@mullian.ee.mu.oz.au
kma@mullian.ee.mu.oz.au

Abstract

This paper reports on the development of a workstation oriented towards BISDN. The workstation transmits and receives telephone, computer data, and video information using ATM packets carried by an ethernet link. The implementation of an experimental terminal using SUN SPARCStations is described. From this study we conclude that video data needs to be split off from other data as close to the ATM layer as possible. A mix of software and hardware used to achieve the splitting function is outlined. The broad architectural details of the hardware splitting functions are described. In addition some theoretical studies of ATM cell throughput on the ethernet link are described.

1. Introduction

An overview of this project and the constraints on its design have already been given [1].

Three 'types' of ATM cells can be distinguished, namely **V**ideo cells (carrying bitstreams between codecs), **D**ata cells (carrying segmented data between higher layer User processes on every workstation), and **S**ignalling cells (carrying data between higher layer Network management processes on every workstation). An ethernet connection between each workstation and a Metropolitan Area Network (MAN) carries a randomly multiplexed collection of these three cell types.

Our task was to develop ways of merging the three streams of ATM cells as they are generated and consumed within the workstation onto a single network access point.

2. How To Separate the Cell Streams

The first part of the task was to analyse the possible data paths within the Multimedia Workstation. Ethernet packets would "enter" the workstation carrying a mixture of V, D, and S type ATM cells. The destination of the V cells is the videocard developed by TRL, which expects to receive ethernet packets containing V cells only. Fig. 1 shows the format of these packets.

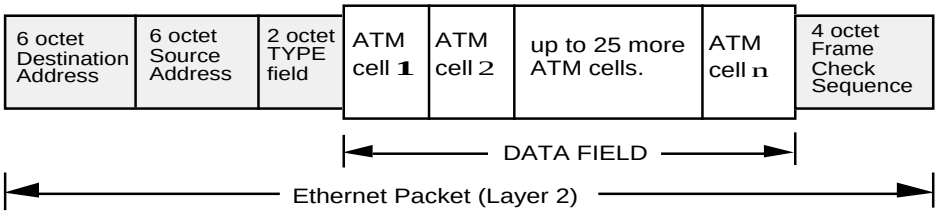

Fig. 1

The destination of D and S cells is the SUNOS kernel. (In this scenario management information for video connections is carried in S cells, management functions are handled by processes running under SUNOS and communicating with the video card via a dedicated RS232 link).

The first solution considered, involving minimal changes to the existing hardware, is shown in Fig. 2.

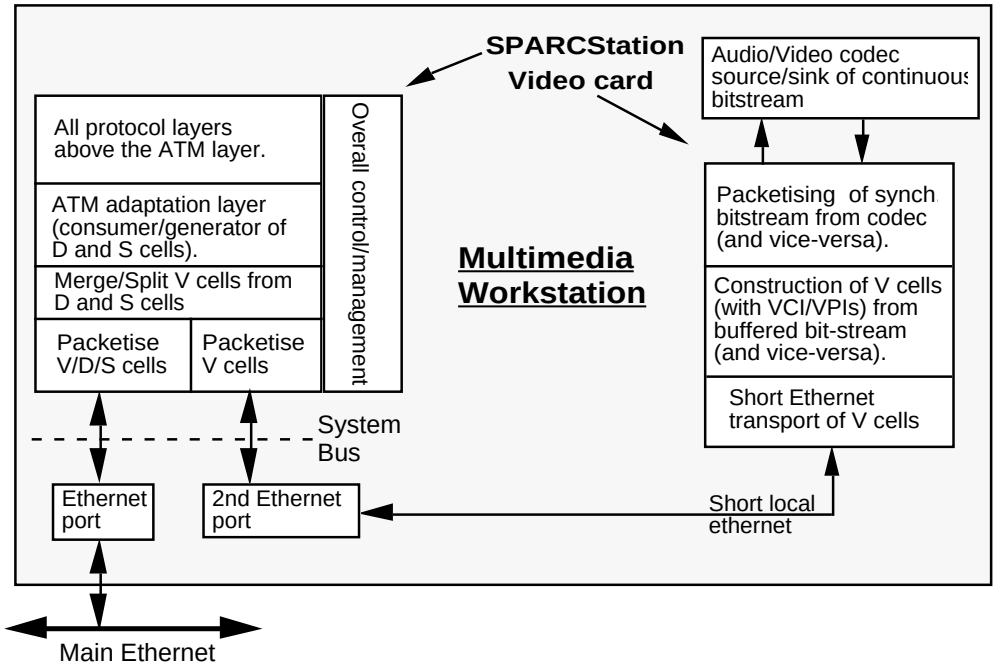

Fig. 2

In this solution the extraction of ATM cells (of all 3 types) would be a task for the SPARCStation1 processor. The ATM protocol stack would be implemented fully in software. At a suitable point in the protocol stack V cells would be re-directed to a second ethernet port. V cells arriving over the main ethernet link would be buffered and sent to the video card in full ethernet packets. V cells arriving from the videocard would be buffered in the kernel and multiplexed with the stream of D and S cells destined for the main ethernet link.

In this scheme it was evident that excessive load would be placed on the system bus (SBus) of the SPARCStation1. Video data was expected to be generated at 2048Kbits/sec. This traffic (including overhead of ATM cells and ethernet packets) would have to travel twice over the SBus, and be processed with requisite speed by the SPARCStation1 processor. It was not our

belief that the SPARCStation1 could simultaneously provide reasonable computing power to its users and necessary real-time processing of V cell traffic.

The solution adopted was not to simply find a machine with a faster bus, nor simply lower the acceptable video data rate. We decided, instead, to remove V cell traffic from the SBus completely. This is being achieved by doing the ATM cell to Ethernet packet conversions in a customised piece of hardware called the Splitter. The revised data paths are shown in Fig. 3.

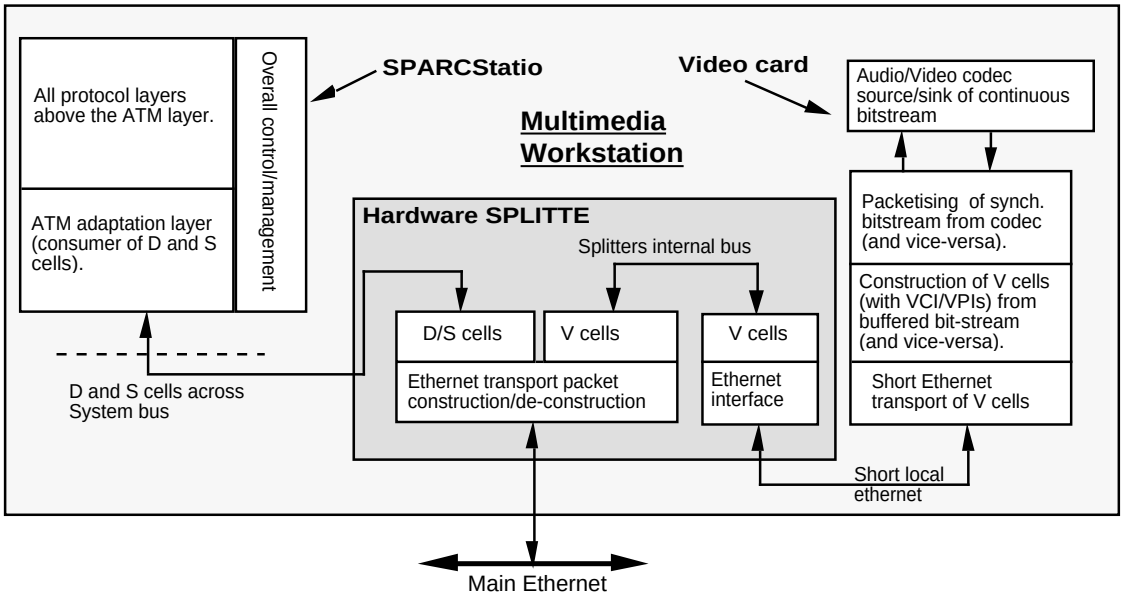

Fig. 3

The Splitter contains 3 interfaces to the rest of the system. One is the SBus interface across which D and S cells will be transferred. The other two are ethernet ports providing access to the main ethernet link and the sub-link to the videocard. The Splitter is initialised, and controlled, by device drivers running on the SPARCStation1. At this stage the Splitter contains only sufficient facilities to separate V cells from D or S cells and to merge V cells on the one hand with D and S cells on the other. The separation of D and S cells occurs after the cells cross the SBus, and is effected by the software in the SPARCStation1.

3. ATM embedded in Ethernet Packets

The method of transferring ATM cells over the Ethernets has been deliberately kept very simple. The philosophy is that error recovery for ATM cells lost through packet loss should be left to the protocols running on top of the ATM layer (since the layers above ATM presume ATM cells to be an unreliable datagram style medium anyway). The underlying Ethernet is intended to provide a transparant Workstation-MAN link.

The ATM cells themselves are carried in each Ethernet packet, without any further identification of bytes, etc. Fig. 1 shows the general format of these packets and Fig. 4 shows a typical ATM cell [1]. The octet order within ATM cells in the ethernet packets is the same as their order on the MAN, that is 5 header octets followed by 48 data octets.

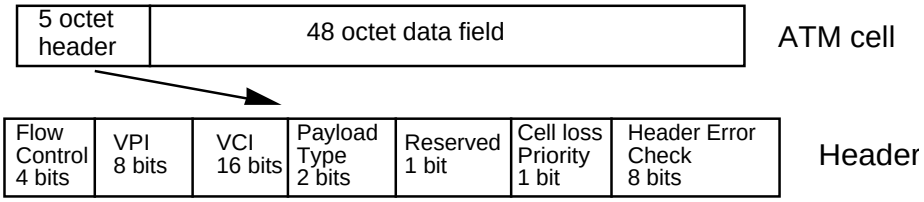

Fig. 4

Across the main ethernet link the sequence of cells in each packet is unspecified and may be a random mixture of V, D, and S cells. The destination of packets from the workstation is always the same - the ethernet/MAN node. Therefore there is no special algorithm relating VCI/VPI pairs to ethernet addresses, at least as far as the workstation is concerned. Across the link to the videocard only V cells are carried. Fig. 5 shows how the codec bitstream might relate to the final contents of each packet (assuming no intermediate 'adaptation' layers, etc, between the codec and ATM layer). In this scenario again only 2 constant ethernet addresses are used.

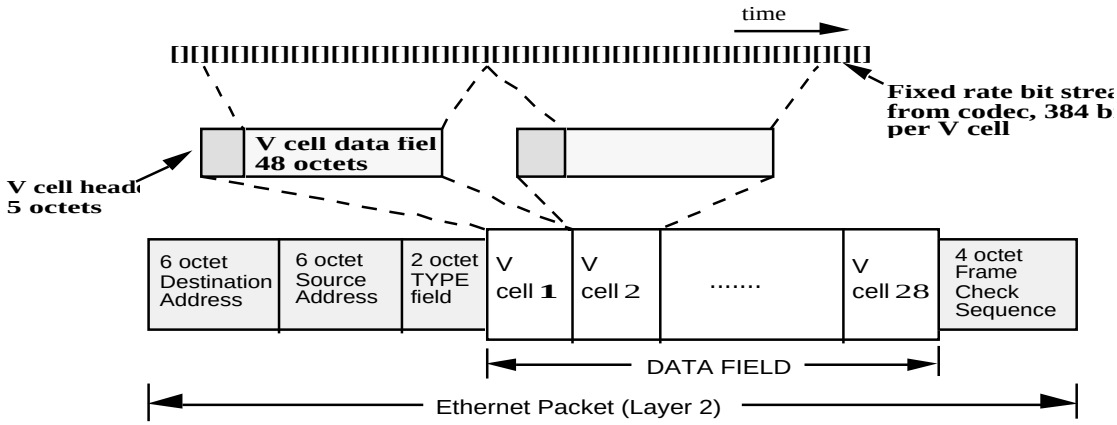

Fig. 5

4. Identification of V, D, and S cells

The concept of V, D, and S cells requires that each cell contain some information in its 5 octet header that identifies what type of cell it is (the intended analogy is with the different physical data streams [channels] offered by BISDN connections). Our work in this area has been tentative due to the current standardisation work being done by others in this area.

A minimum of 2 bits is needed to differentiate V, D, and S cells. Early discussions proposed using either the payload type bits or the reserved bits in the header for this purpose. The payload type bits appear now to have been given another role. We are wary of using of the reserved bits because their use in this case may not be easy to justify.

Our current solution is based on the analogy between cell types and the separate data channels of BISDN. The three cell types are providing separate virtual channels for Video, Data, and

Signalling information. Therefore it makes sense to use the VCI (Virtual Channel Identifier) to indicate the cell type.

There are two possible ways of using the VCI in this fashion.

(a) Dynamically assign and re-assign which VCI's correspond to V, D, and S cells on a call by call basis.

(b) Permanently allocate blocks of the VCI 'address space' to either V, D, and S cell use.

Option (a) results in the more complicated circuitry in the splitter and greater processing time for the kernel resident ATM layer. The VCI is contained in 16 bits of each ATM cell header. Both the splitter and kernel resident software would have to keep tables specifying what type of cells are being carried by each of the 2^{16} virtual channels.

Option (b) eases cell type differentiation by reducing the number of possibilities. Our decision is to use the 2 most significant bits of every VCI to specify the cell type. If, for example, the following patterns were used:

00xxxxxxxxxxxx	-	V cell
01xxxxxxxxxxxx	-	S cell
10xxxxxxxxxxxx	-	D cell
11xxxxxxxxxxxx	-	D cell

then one quarter of the available VCI's would be permanently allocated to V cells and S cells, and one half to D cells. (It is likely that more Data channels will be open at any one time than either Video or Signalling channels, hence it makes sense to allocate more VCI's to D cells). This sort of differentiation is much easier to implement in the hardware splitter, and will allow more efficient software differentiation too. Additionally if more VCI's were needed for D cells and less for S and V cells then 3 bits could be used to partition the VCI space into eighths.

We have not considered how the VPI field could be used. Its definition is currently not as clear and seems to be less analogous to the V, D, and S cell concept.

5. Overview of the Hardware Splitter

Fig. 6 shows basic block structure of the Hardware Splitter, with emphasis being on the data flow paths rather than actual circuit arrangement.

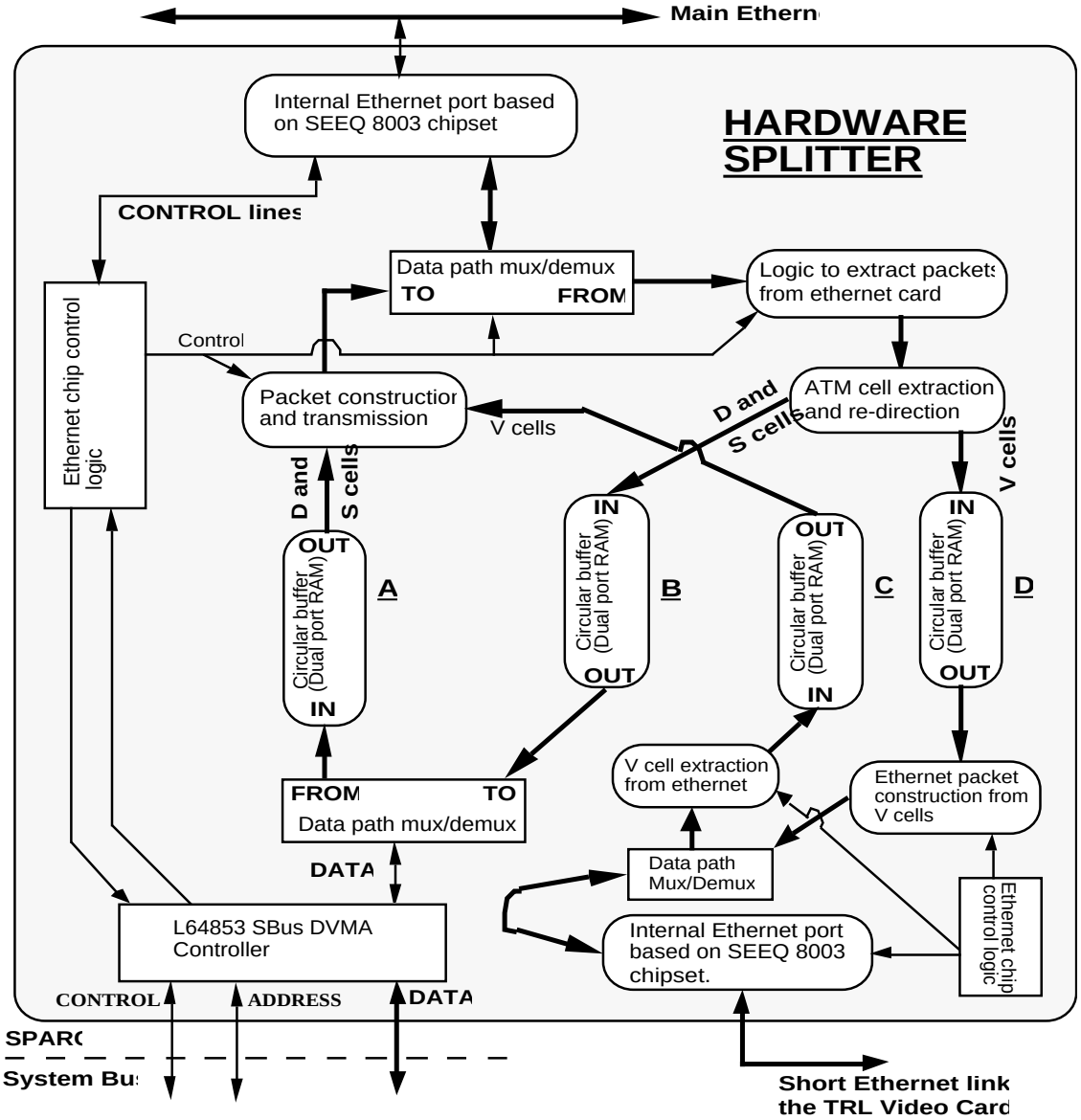

Fig. 6

Data flow is two-way at all three interfaces. On the incoming path from the main ethernet there are three major functions: ATM cell extraction and splitting, ATM cell buffering, and ATM cell transfers to either the SPARC or the video card. On the outgoing path the functions are: ATM cell collection (from SPARC or video card), ATM cell buffering, and the final merging/packetising of ATM cells into ethernet packets as shown in Fig. 1.

The core of the splitter is four circular buffers with independent input and output ports. These are designed to act as FIFOs from which data may be inserted and extracted asynchronously. There are two buffers each for the SPARC interface and the video card interface, one buffer for each direction of data flow. Buffers **A** and **B** are the **transmit** and **receive** FIFOs used by the SPARC ATM software. Buffers **C** and **D** provide buffering between the video card ethernet link and the main ethernet link.

Consider packets coming in from the main ethernet link. The SEEQ 8003 chip passes the entire layer 2 packet to the cell extraction/re-direction block if the packet was correctly addressed. The cell extraction/re-direction block reads the ATM cells (in the data field of each packet) octet by octet, sending D and S cells to buffer B and V cells to buffer D.

When sufficient cells accumulate in buffer D they are extracted and sent across the local link to the Videocard. Buffer B is read when the SPARC initiates a DMA transfer of its contents to main memory.

Incoming packets from the Video Card have their V cells extracted and temporarily stored in buffer C. When the SPARC device driver wishes to send D and S cells they are DMAed from main memory into buffer A. The contents of buffers A and C are used to form composite packets for transmission over the main ethernet.

6. A brief look at the potential ethernet traffic

The use of an ethernet link between the workstation and the MAN is dependent on its ability to cope with both videophone traffic and more traditional data transfer traffic. We have, therefore, done some simple modelling of traffic flow across the main ethernet link to get some idea ofthe link's capabilities. (The link between the Hardware Splitter and the Videocard is not analysed here because it is a sub-optimal solution with no intended longterm use). Fig. 7 shows a simplified model of the sources and sinks of ATM cells in the workstation. The question arises: what D and S cell traffic can we hope for, given certain V cell traffic?

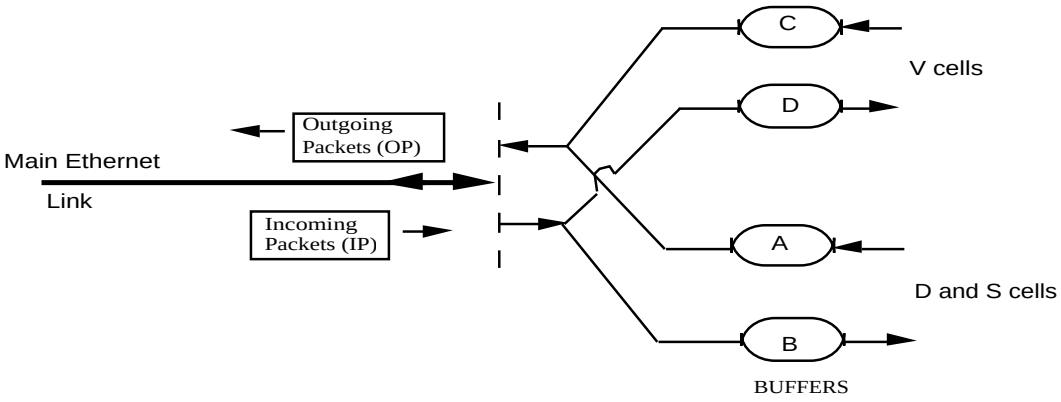

Fig. 7

IP's and OP's are Incoming and Outgoing Packets respectively. The format of IP's and OP's has been shown in Fig. 1. Outgoing V cell traffic (through buffer C) will be regular and bursty, whilst D and S cell traffic (through buffer A) will be more random. However for simplicity we will assume that both buffers fill at a regular rate.

The limitation on maximum throughput is the rate at which packets can be delivered in one piece. Once this is calculated the actual ratio of V cells to D/S cells in each packet can be derived, and hence the actual throughput of D/S cells. The two boundary conditions are the required minimum V cell rate to be supported and the attainable transfer rate of packets.

To get some idea of the absolute maximum theoretical throughput consider the ideal situation where both ethernet stations

(a) are alternating the transmission of packets (there are no collisions);
(b) have always got something to transmit;
(c) and never suffer receiver over-runs.

Add to this a variable delay, D, to cover some non-ideal conditions (e.g. each end cannot switch from transmit to receive instantaneously, transmission line delays, etc). Fig. 8 illustrates this.

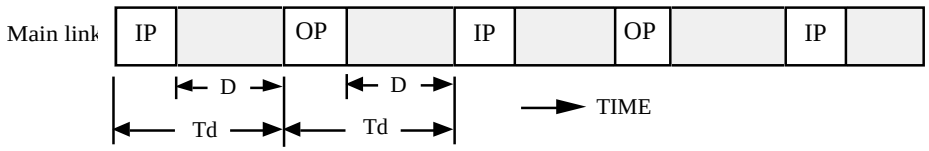

Fig. 8

The maximum packet rate occurs when D=0. The parameter Td is defined such that 2Td is the time between the start of consecutive packets from the same transmitter. Taking one bit time on the ethernet to be 0.1 μs it can be shown (using ethernet specifications [2]) that:

$$Td = D + 30.4 + 42.4*Nt \quad \mu s$$

where D is interpacket delay described above (in μs) and Nt is the total number of cells in each packet (Nt <= 28). The most efficient use of each packet is when Nt = 28, and so:

$$Td = D + 1217.6 \quad \mu s$$

The total one way cell transfer rate is given by:

$$Rt = 28*10^6 / (2*Td) \quad \text{cells/second}$$

The second boundary condition is the need to ensure a minimum V cell transfer rate. A maximum codec rate of 2048Kbit/sec, and packing 48 octets (384 bits) into each V cell (Fig. 5), gives us a required V cell rate of:

$$Rv = 5333.3 \quad \text{cells/second}$$

The minimum number of V cells needed in each packet to ensure this rate is given by:

$$Nv = \text{trunc}(1 + 28*Rv/Rt) = \text{trunc}(1 + Rv*2*Td) \text{ cells. [rounded UP to nearest integer]}$$

Then the actual one way cell transfer rate will be:

$$Rta = 28*Rv / Nv$$

The one way D/S cell transfer rate is given by:

$$Rds = Rta - Rv$$

and the bit rate of ATM cell Data field transfer is

Bds = 48*8*(Rta - Rv)

Thus, with D=0, the maximum bit transfer rate for D/S cell data fields is:

Bds = 2.363 Mbit/s [Nv = trunc(1 + 5333.3*2*1217.6*10^{-6}) = 13]

With Rv held constant at 5333.3 the worst case D/S transfer rate is:

Bds = 75.85 Kbit/s [when 1220 µs < D <= 1313 µs and Nv=27]

For D > 1313 µs, D/S cell transfer effectively stops. Fig. 9 shows a plot of throughput versus delay D for the simple case outlined above.

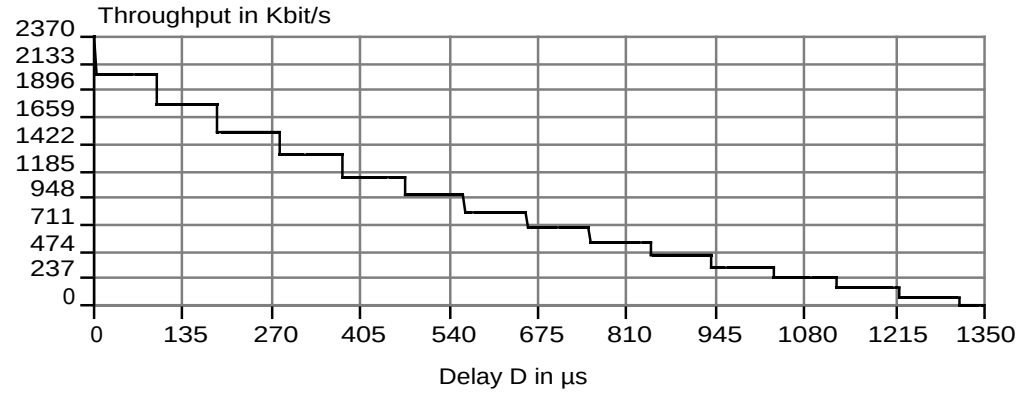

Fig. 9

The step nature of the graph is from quantisation due to having only integer ratios of V cells to D/S cells. Fig. 10 shows D/S throughput for the same conditions except that the required V cell rate is only 1000 cells/second (corresponding to a video bit rate of 385Kbit/s). The maximum throughput is then 3.2 Mbit/s; at D=1315 µs it is still 1.408 Mbit/s.

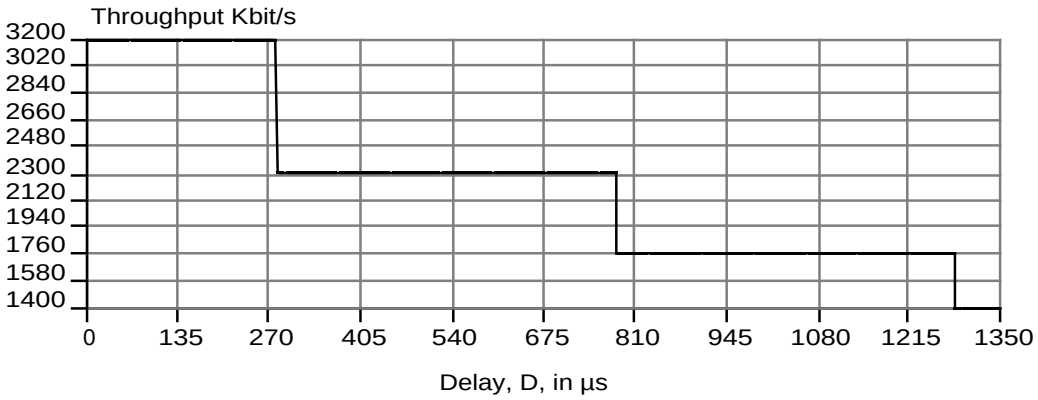

Fig. 10

This analysis has implications for the design of the Hardware Splitter, since it is here that decisions are made as to how the V, D, and S cells are packed into packets. It would seem obvious that the Splitter needs to be configurable by the operating system (while running) to provide optimum performance under varying load and line conditions. It is expected that variable rate codecs will make the V cell load changeable at least from call to call, if not during each call.

The existence of collisions on the main ethernet link will also degrade ATM cell throughput. Given that the main ethernet link may only have 2 stations it becomes reasonable to consider ways of predictive collision avoidance. Collisions also increase local buffering requirements, so their avoidance is doubly beneficial. An example would be synchronisation through alternating packet transmission (as used in the example above). This does not obviate the need for buffering at each end, however. It merely regulates traffic on the physical ethernet so as to avoid collision/backoff/retry scenarios. Also alternating packet synchronisation fails for low cell rate or irregular cell frequency traffic. This is an area we intend to examine further.

There are many other considerations which cannot be included within the scope of this paper.

7. Conclusion

In this paper we have discussed the use of ATM cells to provide Video, Data, and Signalling "channels". We have looked at architectural constraints imposed when using current generation computer systems and separate video sub-systems to provide integrated services. The use of at least some customised hardware has been necessary to achieve this integration. Future developments seem destined to require independant data paths for video and other high speed traffic. However, even the nature of these data paths still needs to be refined. The throughput of an ethernet connection between workstation and MAN has been considered, and questions raised about the need for modified ethernet access algorithms. Clearly one would want to eliminate the two ethernet links in an ATM system as soon as is feasible.

8. Acknowledgements

This project is being funded by Telecom Australia under Contract No. 7064

9. References

[1] K.M. Adams and K.E. Forward. An Experimental Broadband Integrated Services Terminal - System Overview, Proc. 1990 Australian Video Communications Workshop

[2] Digital Equipment Corp., Intel Corp., & Xerox Corp., "The Ethernet: A Local Area Network Data Link Layer and Physical Layer Specifications", Version 1.0 September 1980.