The Desk Area Network

Mark Hayter and Derek McAuley
University of Cambridge Computer Laboratory

May 1991

To appear in ACM Operating Systems Review

Abstract

A novel architecture for use within an end computing system is described. This attempts to extend the concepts used in modern high speed networks into computer system design. A multimedia workstation is being built based on this concept to evaluate the approach.

1 Introduction

High speed networks (100Mbit/s per access point) are currently being designed and coming into use as local area networks (LANs). The desire to support traffic with real time requirements (e.g. multimedia traffic) has lead many groups to propose the use of Asynchronous Transfer Mode (ATM[1]) for such networks. In the longer term the adoption of ATM by Broadband ISDN (B-ISDN) will give wide area ATM coverage. The effective use of these networks depends on the design of end systems which can utilise the bandwidth provided. The real producer or consumer of the data will be a particular processor or device within the end system, so besides considering the design of a network interface, consideration must be given to the method used to transfer the data between this interface and the real end points of communication.

LANs are normally thought of as connecting a set of autonomous machines, where there is a significant delay, compared to processor execution time, in data transfer. LANs also use a set of communication protocols to deal with the unreliability of data transfer between nodes and the consequences of node failure. On the other hand, interconnection networks have been used to link processors and memory systems in high performance multiprocessor machines, using simple protocols, typically transferring small data units (frequently single words or cache lines). In such a system, the communication is comparatively reliable leading to simpler control mechanisms, with failures usually treated as terminal.

We propose an architecture for the internal interconnection of a computer system, which combines various aspects of ATM LANs and traditional interconnection networks. An ATM switch is used to provide interconnections between the components

[1]We use ATM in the general sense defined in CCITT I.113 [2], *'A transfer mode in which information is organized in to cells; ...'*, rather then to refer to specific properties of the ATM Layer in B-ISDN

of the machine and from these components directly to a high speed ATM network. However, all components are treated as would be the components of a workstation. We call this type of interconnection the Desk Area Network (DAN).

In recent years the categories of network have expanded. For example the term MAN has now become accepted, to describe networks joining buildings relatively close to each other. Table 1 gives a guide to existing terms. However, in categorising a network, often it is the protocols and management techniques that differ rather than the underlying networks themselves. For example although the Cambridge Fast Ring (CFR) was designed as a LAN, it has been used as an ATM switch in the Unison project [11], a LAN in the Pandora project [4] and an Ethernet bridging MAN in the Metrobridge [8]. A similar divergence of uses can be seen with FDDI and the Cambridge Backbone Ring [3] (MANs used as LANs) and SMDS (DQDB MAN used as access link to WAN). In this work we are taking a switch developed for the LAN and changing various control and protocol mechanisms to those appropriate for use in a single machine, while attempting to maintain the data transfer characteristics.

Type	Size	Data rate
Wide Area Network (WAN)	>15km	64kb - 50Mb
Metropolitan Area Network (MAN)	1 - 50km	1 - 500Mb
Local Area Network (LAN)	<5km	1 - 500Mb
Interconnection Network	< 4 m	1Mb - 2Gb

Table 1: Network nomenclature

The experimental DAN system we are building is based on a space division ATM switch designed and built by the Fairisle project [5]. Around this switch, we are building a multiprocessor multimedia workstation to gain further understanding of the hardware and software architecture of such a machine, and to enable investigation of the systems issues raised.

2 The Desk Area Network

The Desk Area Network is intended to bridge the gap between multiprocessor interconnection networks and LANs by the use of ATM as a common data transfer mechanism. A combination of mechanisms from both interconnection networks and LANs are used to realise the DAN.

Consider a networked computer system, traditionally based on some bus; incoming data is demultiplexed based on information within the received packets, and then routed over the bus to the memory or peripheral for which it is destined. Similarly outgoing data is transferred over the bus to the network interface for transmission. Basically the network interface is required to translate between the multiplexing mechanism used on the bus and that used on the network.

The DAN does away with this translation step in the data path by using the same multiplexing mechanism within the machine as is used in the network; the multiplexing and protocol techniques used in an ATM LAN are extended to the links between devices and processors within a machine. The use of ATM within the end system preserves the advantages of using ATM all the way to the real end points of the communication. When moving data, the network interface is now a straight forward cell router, while from a control and security point of view, it is the point at which the different protocols and policies used in the DAN and LAN meet.

As in an interconnection network, the small size of a DAN system (both physically and in terms of nodes), enables it to provide highly reliable communication between its nodes and hence allows the internal control architecture and protocols to be greatly simplified.

Within a DAN, the benefits of statistical multiplexing at the ATM cell level will be minimal due to the comparatively small number of active channels within a system at a particular time. While this would be a significant problem in a LAN, the scale of the DAN allows the controlling operating system to be aware of the state of all the attached devices and their bandwidth requirements; this knowledge allows the operating system to alter the queuing and scheduling algorithms used in the switch to optimise performance for particular traffic patterns in the system.

One of the objectives of using a DAN is to transfer data between the LAN and the real end points of communication in the machine with the minimum of overhead and without introducing a bottleneck. This can be achieved by using a switch technology in a DAN implementation which is similar to that used in the network – we intend using a space division ATM switch in our prototype DAN machine, but other techniques, such as an ATM bus, could be used.

We first consider a DAN used in a symmetric multiprocessor.

2.1 A multiprocessor

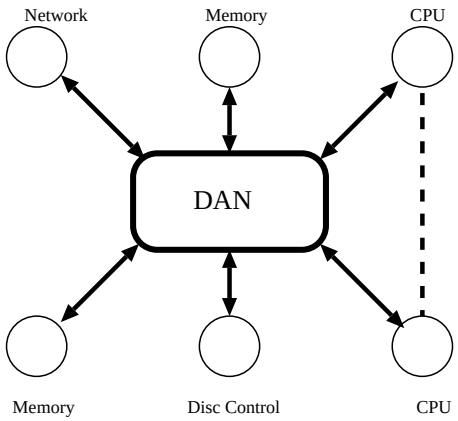

Figure 1: An example DAN system.

Figure 1 shows an example multiprocessor DAN system with an attached local disc. There are some number of processors, some memory, a disc interface and a network interface all connected via the DAN. When using a space division switch, multiple memory units can be used to spread the load across different links in the DAN. The processors would have some amount of local cache memory – the relative speeds of the DAN and the processors, and the number of processors and memory units would determine the size of cache required. Of particular interest is the average latency which can be achieved in the envisaged situation where the DAN interconnects the main (DRAM) memory to a processor's cache (or secondary cache for high performance processors). This is discussed below (section 3.1).

Of interest for a DAN based multiprocessor, are some of the techniques developed for distributed virtual memory [6]. In these systems, a collection of machines on a LAN are used as a pseudo-multiprocessor by arranging that some memory is shared between all the processors. The high bandwidth and low delay between the nodes of the DAN allow a much finer grain of memory sharing than is possible in the LAN environment. Such methods would not be required in the case of a DAN implemented as a TDM bus, where memory coherency could be maintained by *snooping*. However, this is not appropriate when using space division switches (such as Fairisle) to achieve greater system bandwidth and scalability.

In such a system, it is straight forward for a processor to access the ATM LAN. The processors compete for DAN bandwidth into the LAN interface node, where the ATM cells are forwarded onto the LAN connection. Incoming cells are similarly demultiplexed to the appropriate processor or memory unit.

The Axon project proposes a similar system for a host interface to a high speed LAN [10]. The main difference is that in the Axon design the network interface will plug into an existing interconnection network. Thus the point at which the ATM multiplex is lost is the network interface rather than the communication end points.

Similarly, the DataMesh project at HPlabs [12] proposes using a high speed interconnect to link a collection of storage nodes within a single storage server on a LAN. As in a DAN machine, the DataMesh represents a single security domain, so that nodes within the server do not need to authenticate themselves to each other, nor do they need to encrypt internal communications. Another similarity is the ability to use physical connection mechanisms which do not scale beyond a few metres, e.g. ribbon instead of fibre. However, as with Axon, the major difference is that the protocols used inside the DataMesh are different from those running on the LAN.

2.2 The network interface

Previously we simply described the network interface node as forwarding cells between the DAN and LAN. This section describes the functions of the network interface in more detail, and in particular in terms of the in-band and out-of-band processing required.

We are primarily interested in the connection of the DAN to a virtual circuit based ATM LAN. The choice of ATM for the DAN was based on the desire to reduce the in-band processing of data passing through the network interface node and implement what in-band processing that is required in hardware. The functions required are header translation, routing, and queuing; the network interface node is almost identical to the input and output port controllers of the network switches. The differences arise due to the resource allocation mechanisms used within the DAN switch which are reflected in the cell queuing algorithms. For our experimental work, we are able to use the port controller produced for the Fairisle switch, as it is flexible enough to allow the different queuing algorithms to be implemented.

The network interface also represents a security boundary, and so, if secure communication is desired, data must be encrypted at (or before) this point. The placement of encryption hardware at this point allows it to be shared amongst the multiple components present on the DAN.

Although there are many similarities between the data handling in the DAN and LAN, the policies and protocols implementing the out-of-band control are very different in the two cases. Within the DAN, we can envisage the use of shared memory, or procedure call between mutually trusted entities, to implement the control and resource management. The policies implemented would also be radically different; in particular within a workstation, the user would expect to be allowed to dictate such policies. However, when establishing a connection through the network, these internal requests would need to be translated to the network signalling requests and protocol interactions. This is logically a function of the network interface.

2.3 A multimedia workstation

One of the main problems associated with the support of multimedia in a workstation is the management of the real time streams of data. In a workstation based on a DAN, the switch is used to deliver such real time data streams directly to the relevant devices, while their associated control streams are routed to a controlling processor.

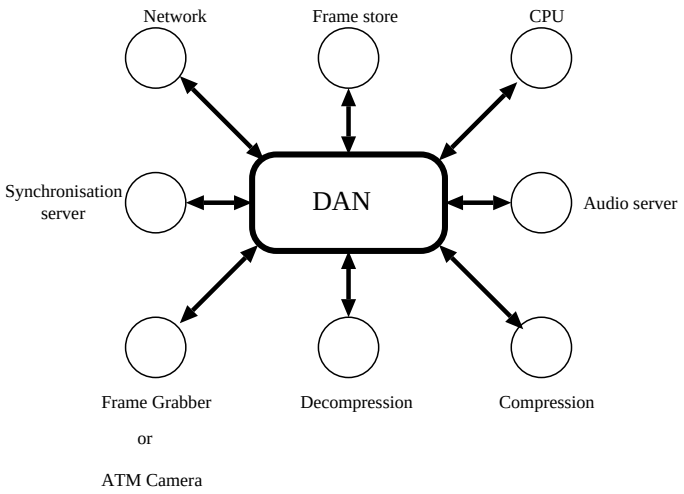

Figure 2: A multimedia workstation DAN system.

For example, consider a video/audio link between two such workstations. The frame capture section (or ATM camera, see later) will produce an ATM cell stream containing the video which is passed over the DAN to a compression unit. Similarly, an audio card generates an audio stream. The compressed video and audio are then passed through the network interface node onto the LAN. On reception the network interface node on the receiving machine will route the incoming cells of the data streams over the DAN to the audio device and decompression unit. In our case since the DAN is a space division switch, these communications can proceed in parallel. Control components can be extracted from the streams and passed on to a server responsible for synchronization; the processor implementing this service need never handle the real time data.

In fact, a multimedia workstation based on a DAN is a highly asymmetric multiprocessor machine. The DAN is used as the common data path between multimedia devices allowing for considerable flexibility in the choice of data paths. For example video may sometimes be transferred without compression (when LAN bandwidth is available), or could be generated by a processor on the DAN, or may come from a storage device, all of which require different data paths through the system. Several of the nodes on the DAN will be special purpose and dedicated to a single function,

others may be more general purpose processors and implement services *on demand*. Some typical components are:

- Control
  There will be a single distributed operating system running on the DAN based machine. There will be need for processor time to manage both the machine's resources and the DAN itself.

- Camera
  To enable sourcing of real time video a camera must be connected. This would most cost effectively be an ATM camera (see discussion later). However, a frame grabber could be used, in which case the captured frame needs to be segmented into cells.

- Audio
  Audio codecs are now common, many offering "CD quality" sound. The audio processing will mainly be done using specialized devices but there is still the need to generate a cell based representation.

- Compression
  For both storage and network bandwidth reasons real time data may need to be compressed (and decompressed).

- Clipping
  If the video is going to a windowed display, a simple optimization at the source is possible by clipping before transmission onto the LAN.

- Display
  In the case of a multimedia workstation the display will need to support a windowing system as well as real time streams[2].

- Synchronisation
  Streams coming over the network will suffer from skew and jitter, and will need resynchronising (the lip-sync problem being a classic example). [9] has proposed that a synchronisation service for multimedia needs to be placed in the end system, or on a network node with a low jitter path to the end system. A processor on the DAN would be suitable for this service.

- LAN interface
  The interface to ATM LAN is straight forward, and allows the interaction of DAN machines across such a network.

- Storage
  A workstation may need a disc for local storage of multimedia documents that are being edited. However, with a comparable speed LAN, dedicated file servers may be a better approach.

[2] An initial experimental system may use separate displays, but the Pandora [4] project has shown the benefits of an integrated approach

In the workstation there will be two main types of communication. One due to the real time streams, and the other control information. Both are implemented on top of the ATM substrate.

For real time streams the existing LAN protocols are used, allowing the LAN interface to be kept simple. For these streams, some guarantees on performance are required (cell loss, delay, etc.). Within ATM based WANs (e.g. B-ISDN) it is believed that such guarantees can be based on the statistical properties of the streams, made possible by the large amounts of statistical multiplexing present. Such multiplexing is less evident in a LAN; work within the Fairisle project is investigating how this affects the nature of these guarantees in a LAN. In a DAN machine, the small number of streams and nodes involved reduces the benefits of statistical multiplexing to a minimum. However, the small size of the system allows less subtle techniques to be used (e.g. based on searching for feasible queue and switch schedules).

For control purposes, both distributed shared memory (across the DAN) and RPC are being investigated.

3 The experimental workstation

The initial workstation will be based around the Fairisle ATM space division switch [5]. The switch fabric will be of either 4 or 8 ports. In its use as a LAN there is a combined input and output port controller for each switch fabric port. These units perform the management, buffering, queuing and routing functions for the switch, and interface to the transmission system[3]. A separate loopback path also exists between the output and input.

The port controllers run a microkernel called Wanda, a locally developed multiprocessor operating system. This supports the ANSA Testbench[4] [1] over the Multi-Service Network Architecture (MSNA) [7]. ANSA provides a distributed computing environment, while MSNA is a network architecture designed to support multimedia by providing ATM internetworking.

When used for DAN experiments the port controllers are used as the processor elements - each has an ARM3 processor[5], up to 4Mb of DRAM, an IO bus, and 128K of 35ns SRAM used for the cell buffer memory. The IO bus allows the addition of frame capture and display devices, audio devices, and connections to existing networks (e.g. Ethernet and CFR).

In our DAN workstation, the port controllers are used in a variety of ways. A number are used as general processing elements, in particular to execute the controlling

[3]The transmission interface supports a data rate of up to 160Mbit/s; for LAN work, TAXI devices are used

[4]The ANSA Testbench is an example implementation of the ANSA Architecture

[5]A 15MIPS RISC processor designed by Acorn Computers Ltd.

software. At least one is equiped with a transmission system and is used to provide the link to the Fairisle network. The rest are used in conjunction with devices on the IO bus to emulate true ATM devices; this mechanism allows the use of existing devices (e.g. frame capture cards) as ATM controlled devices for development, while work on directly connected devices is in progress.

3.1 ATM devices

One of the features of the DAN architecture is that everything uses ATM cells for both data and control communication. Some of the preliminary designs for ATM based devices are presented below.

A minimalist ATM camera would contain a CCD sensing element, an AtoD converter, a cell framer and some simple control logic. The output of the camera would simply be streams of cells containing the raw digital video, while incoming cells would provide commands to the camera. We hope it may be possible to use some of the "camera on a chip" devices becoming available and include the ATM control with the sensing array all on the same chip. More likely, major surgery will be performed on a Camcorder.

The design of an ATM based cache controller and memory system is not radically different from those seen in systems based on a TDM bus. Instead of the cache controller generating a bus transaction on a cache miss, an ATM cell is transmitted to the memory system containing the information that would normally be seen on a bus (cycle type, address, data). In the case of a write this is similar to write posting, in the case of a read, a reply cell would be generated from the memory to the processor. To be useful, the latency of this transaction must be comparable with a straight forward (DRAM) memory system.

Consider a simple cache controller (which can only have one outstanding cache miss at a time), a 32 bit wide DRAM memory system of 80ns components and no contention for the memory ; this could deliver a 256 bit cache line in 720ns. For an ATM based system, we note that it is possible to pipeline the cell reception, memory cycles and cell transmission in the memory unit, so that the transaction can be achieved in a single cell time. On Fairisle this gives a cycle time of 2800ns. However, although we are currently using the Fairisle switch for the prototype (because it exists), we would envisage the use of a purpose built DAN in a product. Using the same technology as Fairisle, a cycle time comparable to the raw DRAM speed is possible, by simply increasing the width of the data path through the switch; a 32 bit wide path (640Mbit/s per port) gives a cell cycle time of 700ns.

Within the confines of a single DAN, the cell transfers of such transactions can be made reliable; however, we shall be investigating the feasibility and desirability of extending this system to the LAN, where reliability cannot be assumed.

We have two ideas for an ATM based display, which place different amounts of com-

plexity in the device. One is a dumb frame buffer, of a similar design to the memory system described above; the other is smarter, and understands *blit* type requests. A project in the laboratory has produced an HDTV frame buffer (1952x1192x32) which can be attached in a straight forward manner to the port controller. With this we will be building a display of the smarter variety.

4 Conclusion

A novel workstation architecture, based on a Desk Area Network, has been described. The aim is to integrate the data transfer mechanisms used in the end system and in the local and wide area networks to build a multimedia workstation which can effectively utilise high speed networks. An experimental multimedia workstation is currently being built around the DAN ideas.

References

[1] Architecture Projects Management. *The ANSA Reference Manual*, release 01.00 edition, March 1989.

[2] CCITT. Vocabulary of terms for broadband aspects of isdn. CCITT Draft Recommendation I.113, May 1990.

[3] A Hopper D Greaves, D Lioupis. The cambridge backbone ring. Technical Report 90/2, Olivetti Research Limited, February 1990. Also in IEEE Infocom 90.

[4] A Hopper. Pandora - an experimental system for multimedia applications. *ACM Operating Systems Review*, 1990.

[5] IM Leslie and DR McAuley. Fairisle: An atm network for the local area. *To appear: Proc. ACM SIGCOMM*, September 1991.

[6] Kai Li and Paul Hudak. Memory coherence in shared virtual memory systems. In *Proceedinga of the 5th ACM Symposium on Principles of Distributed Computing*, pages 229 – 239, August 1986.

[7] DR McAuley. Protocol design for high speed networks. Technical Report 186, Cambridge University Computer Laboratory, 1990. Ph.D. dissertation.

[8] Olivetti Research Ltd. *The O.R.L. Metrobridge User Manual*, 1.08 edition, March 1990.

[9] C Sreenan. Synchronisation of real time streams. Private communication, March 1991.

[10] J Strebenz. Axon : Host - network interface design. Technical Report WUSC-90-7, Washington University in St Louis, March 1990. This is one of a series of papers on the Axon project.

[11] DL Tennehouse and IM Leslie. A testbed for wide area atm research. In *Proc. ACM SIGCOMM*, September 1989.

[12] John Wilkes. Datamesh - scope and objectives: a commentary. Technical Report HPL-DSD-89-44, Hewlett-Packard Company, July 1989.