Technical Report

UCAM-CL-TR-258
ISSN 1476-2986

Number 258

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Two papers on ATM networks

David J. Greaves, Derek McAuley,
Leslie J. French

May 1992

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Private ATM Networks.

David J Greaves - Olivetti Research Ltd, UK.
Derek McAuley - University of Cambridge, UK.

Presented at the 3rd International IFIP WG 6.1/6.4 Workshop
'Protocols for High Speed Networks'
Stockholm May 13-15, 1992.

Abstract.

This paper advocates the use of local area networks which use 48 byte ATM cells. Hosts connected to the network are fitted with ATM interfaces and run a new protocol stack, up to the network level, which avoids multiplexing and efficiently handles the out-of-band signalling used by ATM.

The private network may be of WAN, MAN or LAN dimensions and contain several different network technologies, provided each is able to perform the basic function of carrying ATM cells from one point to another. The private network may be connected to the B-ISDN at one or more points.

1 Introduction.

Our research is directed at the provision of a distributed multimedia application environment. To attain this we are concerned with both communication architectures that can provide the desired services and how such services are supported within end-systems.

In this paper we discuss why we consider that an end-to-end Asynchronous Transfer Mode (ATM) network provides a useful basis for the communications architecture for such systems and hence leads to the desire for private ATM networks for local working and, via B-ISDN, ATM wide area interconnection.

We believe that besides cost reduction aims, functional requirements will lead to different solutions for local and wide area ATM networks and that these solutions will change with time and technology. To address this we have developed the Multi-Service Network Architecture (MSNA). We present a brief description of MSNA, with both reasons for its internetworking approach and examples of its use.

While we advocate the use of ATM networks on an end-to-end basis, we also realise their usefulness in the interconnection of packet switched networks; the paper also includes a brief description of our experience with the use of MSNA.

2 Motivation for using ATM in the private local area.

The arguments which convinced the CCITT to recommend ATM as the solution for Broadband-ISDN also operate in the local area for private multiservice networks. Research work to establish this point has been carried out at the University of Cambridge Computer Laboratory and at Olivetti Research Ltd. Increasingly computer manufacturers are promoting the use of ATM techniques in privately owned and operated digital networks. A specific example is the 'Emerald' ATM switch recently announced by the BBN company [1].

By 'ATM techniques' we mean the use of a fixed-length cell as the primary means of information transfer, where the periodicity of cells is not known by the receiver in advance, but is indicated by a circuit identifier in the cell header. (This definition is taken from the CCITT recommendations.)

The attractive properties of ATM itself include:

- Support of a mixture of traffic types, including fixed, variable rate, and bursty traffic.
- Low jitter owing to short cell size and reduced switching delay due to cut through of multi-cell blocks.[1]

The fact that B-ISDN has adopted ATM as its transfer mode add two further, consequential advantages:

- Increased availability of special purpose VLSI devices.
- Opportunity for ease of interoperability between private and public networks.

These advantages can only be fully realised if the private networks use the same size cell *payload* as the public networks, namely 48 bytes. However, the header size and *format* is not particularly important, since in ATM, headers can be manipulated by each switching entity, while the payloads are passed unaltered from one point to another.[2]

2.1 Comments on cell payload size.

It is not universally agreed that 48 bytes is the optimum payload size for a general purpose network. On the other hand, if one agrees that a fixed-size packet (or cell-based) network is preferable to one which supports variable length packets, then it is clear that, at least, all components of the network infrastructure should support the same cell size. This implies that private ATM networks should employ the same 48 byte payload size as the CCITT's B-ISDN.

[1] *Cut-through* switches are a class where the start of a message may already have left the switch on the appropriate output port, before the end of the message has been received at the input.

[2] Our definition of MSDL in Section 3.1 will imply the control of other header bits, such as cell loss priority and payload type; we treat the coding of in-band indications as data-link specific so that their setting and interpretation is performed by MSDL, on a per-VCI basis, and according to the QoS of that VCI.

Before returning to issues of fixed size versus variable size in Section 2.2, we comment on technical issues of cell size.

It has often been complained that a 48 byte payload is ridiculously small for computer data applications. The principle objections are that even the smallest messages sent by the current generation of distributed applications are much larger (e.g. a keystroke message using the X protocol over TCP/IP is 72 bytes in size) and that no computer would like to take interrupts for every 48 bytes received. These statements are generally correct, and the answer that this paper offers is that sensible computer network interfaces to ATM networks ameliorate these objections.

A payload of the order of 48 bytes was chosen by the CCITT so that the packetisation delay of 64 kilobit voice, when filling cells with 44 to 48 samples, was acceptably low: i.e. 6 milliseconds. Computer network users, who do not see speech as the predominant network traffic and who expect only low average utilisations of their networks, have argued that a larger cell size could easily be used, provided the cell is only partly filled when used for speech. If compact disc quality stereo sound becomes the norm, then the larger cell is filled already with the same packetisation delay. (E.g. a CD stream at 1.4 MBit/second would require a cell of 1050 bytes for 6 ms duration.)

These application and host interface arguments may indicate that a larger cell size is appropriate. Some multiple of a video-RAM shift register length has been suggested as suitable for easy implementation [2]. However, no single cell size will suit all applications; most existing data-oriented applications require variable length messages and will continue to do so. Clearly adaptation of these variable length messages to a sequence of fixed sized cells is required. ATM implies this approach, and once the need for such adaptation is recognised, whether implemented in hardware or software, the argument of what size cell to use becomes masked from the application, in all respects except delay and jitter performance.

2.2 Switching characteristics of short, fixed-length cells.

The basic principles of ATM switching is that fixed length cells are good for switching with low 99-percentile of jitter, while 'short' cells reduce the jitter and delay (and hence buffering) nearly in proportion to their length[3]. It is well known that in a variable length message switching system the 99-percentile of jitter is proportional to the packet size found in the tail of the packet size distribution [3]. Therefore if all messages are to be kept short, it is sensible to make them all the same size. This eases hardware design and in particular the buffer management. The reduction of delay with cell size is seen from a simple dimensional analysis, since both the duration of a cell (for a given transmission rate) and the queue sizes in switches are proportional to the cell length.

Short cells also simplify priority mechanisms, since there is no need to preempt transmission of a long, low priority message, in order to send higher priority messages; there are no such long messages. Finally, short cells imply that the speed penalty from not providing cut-through switches is minimised, and indeed, cut-through designs for ATM switches are rare, if not non-existent.

[3]The 'nearly' is due to the bulk size increasing as messages from a fixed distribution are fragmented into a greater number of cells as the cell size is decreased.

When we measure delay and jitter in seconds, rather than bits, cell duration, rather than length, is clearly the critical factor. In particular, cell size is only important in terms of multiplexing performance when the lowest rate link(s) of an ATM system are considered. The lowest rate links are likely to be between 100 and 150 Mbit/second. This is the limit of inexpensive multi-mode LED fibre technology and 10 K series ECL. For example, the basic SONET-based access to B-ISDN operates at 155 Mbit/second. The Advanced Micro Devices TAXI/FOXI chip set uses this technology at rates including 100 Mbit/second.

At rates of 100Mbit/second, the 48 byte payload cell duration is about 4 microseconds. When ten or so streams are statistically multiplexed and utilisation of the link is relatively high, the mean waiting delay for a cell will be about 20 microseconds and the 99-percentile between five and ten times greater, giving at worst, 200 microseconds. Although these example figures are quite specific, values in practice depend on many factors, particularly on arrival discipline and any priority mechanisms. The importance of the example lies in that the 200 microsecond 99-percentile would be multiplied by ten to 2 milliseconds if a 480 byte cell was used. Clearly, 2 milliseconds of jitter from a single switch or link may have a significant effect on certain real-time applications, especially voice. With more than one multiplexing point in the system, the situation becomes worse, since jitter increases as switches are connected in tandem (although at a power which is less than linear with the number of stages). The conclusion of this is that if a network of cheap 100 to 150 Mbit links is to be used, increasing the cell size significantly beyond 48 bytes, or 4 microseconds is unattractive.

3 Protocol Architecture for private ATM networks.

In this section we present a protocol architecture for an *ATM internet.* We assume each end host or other network user is fitted with an ATM interface and runs the protocol stack. Issues related to the support and integration of existing network architectures with new ATM and B-ISDN networks are discussed in Section 3.2.

Our protocol architecture takes an internetworking approach in an attempt to address the problem of heterogeneous cell-based networks. While this could be said to be an historical problem at Cambridge due to a surfeit of different types of ATM networks, standardization activities have already defined two different implementations of the ATM service in B-ISDN and DQDB[4]; it would seem that as with packet networks, this heterogeneity will only increase with time.

One reason for this heterogeneity will be related to cost reduction for local area networks. Already several computer manufacturers have suggested the use of TAXI transmission systems instead of SONET due to the cost saving. Similar cost reduction exercises are bound to happen in switching systems; e.g. the B-ISDN virtual path service may not provide the management benefits in a LAN to make it worthwhile supporting.

Another reason for heterogeneity is a change in the required functionality. When we constrain ourselves to consider all ATM networks as composed of switches interconnected by point-to-point links, an argument can be made for following B-ISDN standards; however this is a naive approach. Consider an ATM based low power radio network for hand held

[4]Note here we refer to the DQDB access protocol not the higher level functions of 802.6 or SMDS.

computers: firstly, the requirements for radio media access require significantly different ATM headers from B-ISDN, while the requirement to deal with multiple simultaneous receptions of cells by different base stations and their consequent resolution requires that there be information in the header to identify duplicated and out of sequence cells.

Both these arguments come together when considering rings and dual bus solutions for ATM switching; these solutions can lead to cost reduction while their implementation requires a different cell header from B-ISDN to accommodate media access.

3.1 MSNA.

The protocol architecture presented is the Multi-Service Network Architecture [4]. An operational MSNA internet exists in Cambridge UK and spans the sites of Olivetti Research Ltd and the University Computer Laboratory. The physical network components consist of Fairisle ATM switches [5], slotted rings [6] and [7], and, in the near future, radio based ATM systems.

ATM networks allow users to specify differing qualities of communication service. MSNA is designed so that its implementation can be integrated closely with the end system processor scheduler to allow the provision of quality of service to the actual application. The desire is to be able to identify and schedule the relevant active entity in the end system with the minimum of protocol processing. One particular mechanism used in MSNA is to minimise, if not eradicate, the layered multiplexing often found in communications systems.

The multi-service network layer (MSNL) is an ATM internetworking service which is based on the idea of a lightweight virtual circuit providing raw ATM access, that is an end-to-end stream of ATM cells. We choose a connection oriented approach at the MSNL level as we see the virtual circuit as the obvious unit to which to attach a quality of service; for applications not requiring traffic guarantees we use a pipelined circuit set up mechanism to achieve a more rapid start up.

We use the term 'lightweight' to refer to the fact that, at the ATM level, no flow or error control is *required* on a hop by hop basis[5], and that the resources allocated to the connection are neither to be thought of as valuable or permanent. One aspect of this lightweight nature is that MSNL connections may be unilaterally de-allocated by any of the MSNL entities involved, in particular due to garbage collection to recover VCI space from idle or dead connections. Applications using MSNL directly are responsible for re-establishing the circuit, although for many 'datagram' oriented applications, a layer on top of MSNL provides re-establishment *on demand* without explicit interaction with the higher-layer software; e.g. such a service is used in the implementation of IP over MSNL.

To support MSNL, each different ATM network type is required to provide a generic virtual circuit service interface, called MSDL. An MSNL circuit (called a *liaison*) is formed from a concatenation of MSDL circuits (called *associations*).

---

[5]Of course, flow and error control may be implemented to increase system performance – the CFR [6] implements both – however, MSNL does not rely on all hops being totally reliable.

[6]It is not possible to give an exact match between OSI and MSNA layers; the diagram indicates where the functions of the different MSNA and OSI layers have greatest overlap.

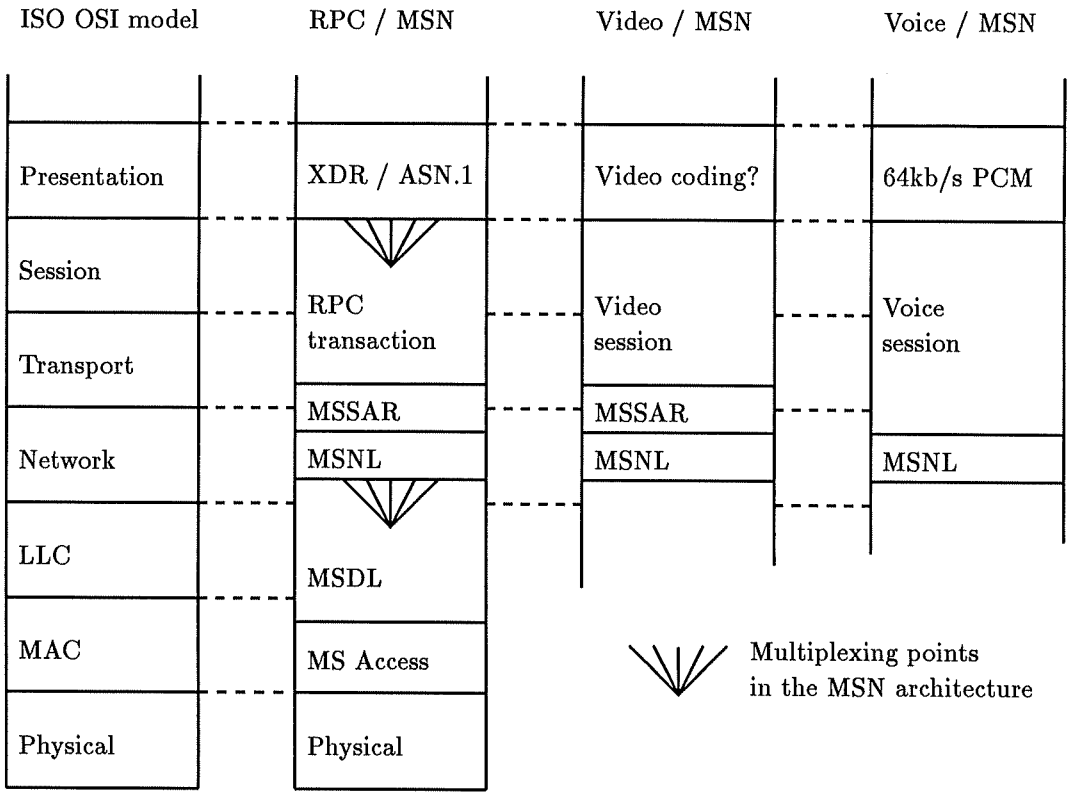

Figure 1: Relationship of functions within MSNA to the OSI Reference Model.[6]

There are three important aspects of MSNL:

- it defines the MSNL address (MSAP),
- it defines an out-of-band liaison set-up procedure,
- it does not multiplex its liaisons over the MSDL associations.

Since MSNL liaisons are not multiplexed over MSDL associations, MSNL does not require in-band protocol headers in the service units. This means that MSNL introduces no processing overhead in the data path, and provides the same basic data interface as an MSDL association. The function of forwarding cells between heterogeneous networks requires that the VCI mapping function be capable of dealing with mapping between the two different header formats involved; this is straight forward in both hardware and software.

An MSNL liaison is established between two MSNL SAPs (MSAPs). These are unique and are allocated from a 64 bit global address space[7]. A host computer may have many MSAPs

[7]The currently adopted approach at ORL and the University of Cambridge Computer Laboratory is to base 32 of the 64 bit address on IP addresses. This simply provides a convenient unique identifier space.

(loosely corresponding to the conventional idea of multiple ports), but on the other hand, there may be many computers sharing a single MSAP, such as individual controllers on the ports of a fast-packet switch. In general, for ease of routing decisions when a connection is set up, it is beneficial if the structure of the 64 bit numbers is actually hierarchical. In [4], the division into separate, 32 bit, *identifier* and *port* port fields is suggested. This optimises the typical case where multiple MSNL clients are situated at a single location (host). However, the individual client streams do not become multiplexed, owing to the separate liaisons for each client.

Setting up an MSNL liaison involves establishing a concatenation of MSDL association hops, with the MSNL address providing the addressing mechanism. During liaison establishment, routing is performed at each MSNL entity to select the appropriate MSDL instance for the next hop. The MSDL definition for each network type then maps the MSNL circuit establishment mechanisms to those appropriate for its underlying network type; this can involve both address mapping and, in some cases, protocol mapping. Once the liaison is established the routing of cells from one MSDL instance to another is normally performed by hardware; where the two instances are identical, this is the normal VCI mapping function seen in switches, where they are different, the hardware also translates the other fields of the header.

3.2 Interconnecting MSNA sites using B-ISDN.

The MSNA concept of providing an end-to-end service for cell payloads enables any adaptation layer to be used over MSNA. In theory, the same is true in B-ISDN, since the 'Empty' adaptation layer is available as a specified service, along with the four prescribed adaptation protocols. However it is unclear to the authors whether B-ISDN providers will allow users access to the network without one of the non-empty CCITT adaptation layers being imposed. Another doubt is whether VCI/VPI space will be available as cheaply and freely as in MSNA. With unfortunate pricing policies, customers may find it attractive to use an AAL which provides a circuit multiplexing function in order to save on active VCIs or VPIs.

Assuming these problems do not arise, or arise in a form which is easily overcome, local private MSNA domains can be interconnected over B-ISDN without impact on the MSNA protocol architecture. Since MSNA was designed to cope with heterogeneous networks with various signalling architectures and cell header formats, alternative control mechanisms for the virtual circuits used on the public network offer no new management problems; B-ISDN simply presents itself as an instance of MSDL.

4 Adaptation layer protocols

Our aim of producing an ATM internet, such that ATM is delivered into the real end system, means that our consideration of adaptation is concerned primarily with efficient implementation in end systems rather than as a means of interconnection of current packet or STM switched traffic types. However, as previously mentioned, by providing end-to-end service for cell payloads, MSNL can support any adaptation layers, in particular those being proposed by CCITT.

Within MSNA, and in keeping with the desire to minimise multiplexing, we have concentrated on adaptation layers which do not perform cell level interleaving of different segmented blocks over an single virtual circuit. If cell level interleaving is required, the VCI which is designed for the purpose is used, i.e. we use multiple MSNL streams. In B-ISDN terms this is equivalent to using a virtual path service where the end-to-end VCI bits distinguish cell interleaved blocks, rather than using a single VCI and sorting cells into blocks by another layer of multiplexing defined by the MID. In the MSNA architecture this block level service is known as MSSAR.

We have been using a particular adaptation protocol for a number of years, while investigating issues in adaptation [8]. In investigating the minimal information required to implement sufficiently secure yet efficient and simple to implement adaptation layers we have arrived, together with other researchers, at a proposal which requries that the ATM cell header now include a logical end-to-end user data bit[8].

The Bit can be used to implement a range of different adaptation layers. For example, for data services, the bit indicates last cell of a fragmented block; the last cell includes a 32 bit CRC over all the cells forming the block and a length field. The CRC detects the usual bit and burst errors, as well as most cell reordering[9], while the length is required to detect certain cell drop outs where the content and position of the cell contrive to provide no contribution to the CRC. This mechanism, now known as AAL-5 [9], is undergoing standardization activities in ANSI and CCITT.

We use this adaptation layer without further multiplexing in higher level protocols. For example a single VCI is used between a two threads to implement a reliable transport, while for RPC, a client acquires a VCI for the duration of a call to a server. This manner of working means that the VCI associated with received blocks can be used to identify the final recipient of the data (e.g. some thread). At this point the standard resource allocation mechanisms of the operating system associated with buffering and scheduling can be invoked without further protocol processing due to more layers of multiplexing.

4.1 Interconnection of existing networks.

Whilst aiming for a network infrastructure in both the local, metropolitan and wide areas based on interconnected ATM networks to provide integrated services to end-systems, we also appreciate the role ATM has to play as a integrated transport mechanism for current packet and synchronous traffic.

The MSSAR service provides many of the same facilities as Frame Relay or B-ISDN AAL-5, that is, a connection oriented data service without guarantees. Hence we have used this service to provide interconnection of packet networks between Olivetti and the Computer Laboratory. Using MSSAR implemented in the Wanda operating systems we have implemented bridging of 802.3 networks; this is in everyday use bridging XNS traffic between the Olivetti and Computer Laboratory Xerox systems using MSNA over the Cambridge Backbone Ring.

---

[8]The December 1991 CCITT draft I.361 extends the payload type to three bits and defines this end-to-end user data bit.

[9]To detect whether two cells i apart have been swapped we need to select a polynomial such that the CRC of $x^{i*384} + 1$ is non-zero.

We have also implemented MSNA in a number of 4.3 BSD derived systems, and as well as providing both MSNL and MSSAR to user processes as a new protocol family, MSSAR can be configured to present itself as an IP network interface, hence providing IP connectivity over our ATM internet. Circuits are established on demand to the next IP hop as indicated by IP routing; these circuits are cached to be used for future IP packets routed to the same hop and are deleted when that are observed to be idle for several minutes. Currently, about 150 machines at Olivetti and the University are able to communicate between each other using IP/MSNL; widescale use for everyday IP service awaits either our ATM network to become as reliable as Ethernet or we acquire some host based dynamic IP routing software able to perform automatic fail-over.

The most stringent synchronous requirement so far is within the Pandora system [10], where the audio component of the system uses standard 64 kbit/second sampling. This is transported over the network in 2 millisecond units consisting of 16 samples with a source time stamp in each ATM cell. The time stamps are used for resynchronisation at the receiver rather than any mechanism based on recovery of a 8kHz timing signal from a transmission system [11].

5 Conclusion.

We consider that the merits of ATM for wide area public service also apply in the local area and for private networks. Starting from this, the aim of our work can be described as 'ATM everywhere'. That is we are primarily concerned with the delivery of integrated services based on ATM to end-systems, even where such end-systems are hand held mobiles. This leads to our internetworking approach and in to a different set of considerations from B-ISDN in the adaptation protocols we support. We consider that the B-ISDN supporting an empty adaptation layer plays a key role in this approach by the provision of wide area coverage for ATM.

References

[1] 'BBN unveils broadband strategy with Emerald switch.' 'Communications Networks' October 1991.

[2] 'Micron Mos Data Book' from Micron Technology Inc, 2805 East Columbia Road, Boise, Idaho 83706.

[3] 'A Fast Packet Switch for Integrated Services.' P Newman. IEEE JSAC 6(9) December 1988.

[4] 'Protocol Design For High Speed Networks.' DR McAuley. University of Cambridge technical report 186. December 1989.

[5] 'Fairisle: An ATM Network for the Local Area.' Ian Leslie and Derek McAuley. Proceedings of SIGCOMM '91, Zurich, September 1991.

[6] 'The Cambridge Fast Ring Networking System.' A Hopper and RM Needham. IEEE transactions on computers, Vol 37 no 10. October 1988.

[7] 'The Cambridge Backbone Ring.' David J. Greaves, Andy Hopper and Dimitris Lioupis. Proceedings of IEEE Infocom 90, San Francisco 1990.

[8] 'Cambridge HSLAN protocol review.' DJ Greaves, ID Wilson. Proceedings of IFIP WG6 International Workshop on 'Protocols for high-speed networks' edited by H Rudin and R Williamson, held at IBM Ruschlikon 1989. Elsevier 1989.

[9] 'AAL-5 – A New High Speed Data Transfer AAL.' IBM et al, ANSI Committee T1 Contribution T1S1.5/91-449, November 1991, Dallas Texas

[10] 'Pandora - An experimental distributed system for multimedia applications.' Andy Hopper. ACM Operating Systems Review, April 1990.

[11] 'Network Compatible ATM for Local Network Applications.' Phase 1, version 1.0, Apple et al, April 1992.

Protocol and interface for ATM LANs.

David J Greaves - Olivetti Research Ltd, UK.
Derek McAuley - University of Cambridge, UK.
Leslie J French - Olivetti Research Ltd, UK.

Presented at the
'5th IEEE Workshop on Metropolitan Area Networks'
Taormina, Italy, May 1992.

Abstract.

This paper advocates local area networks using the Asynchronous Transfer Mode, where data is carried in the payloads of 48-byte cells. We describe the design and performance of a simple ATM host interface for the DEC Turbochannel together with the MSNA protocol architecture. We describe how MSNA creates a homogeneous internet for ATM hosts and devices. We discuss the implementation of an adaptation layer for computer data which is able to take full advantage of MSNA semantics, and which makes use of the end-to-end ATM layer header bit which has recently been accepted.

1 Introduction.

Recently, ATM has been advocated as an important technology for the interconnection of heterogeneous wide area network types including traditional packet and synchronous switched networks. We consider that the advantages of ATM for the wide area also extend into the local area. The interconnection of these networks to provide an end-to-end ATM service is an important component in providing a flexible service to applications.

Computer manufacturers are beginning to promote the use of ATM techniques in privately owned and operated digital networks. Examples include the 'Emerald' private area ATM switch recently announced by the BBN company [BBN], a range of switches and host interfaces from Fore Systems [FORE] and the use of DQDB as a customer premises access network for B-ISDN.

We take our definition of 'ATM techniques' from the CCITT recommendations to be: 'the use of a fixed-length cell as the primary means of information transfer where the periodicity of cells is not known by the receiver in advance, but it is indicated by a circuit identifier in the cell header.'

Colaborative research work to investigate this approach is being carried out at the Univer-

sity of Cambridge Computer Laboratory and at Olivetti Research Ltd. We are developing ATM host interfaces for standard workstations and have implemented the MSNA protocol suite in UNIX and in an experimental microkernel called Wanda. In this paper we describe the interface for the DEC Turbochannel.

2 ATM Motivation.

The motivation to use ATM techniques stems from two different points of view:

- as a technique which lends itself to the implementation of cost effective high performance switches,
- as a flexible communication mechanism.

The technological and performance arguments for networks based on a general topology interconnection of ATM switches have been, and continue to be, fully represented in the literature. Switches with 160Mbps line rates and aggregate capacities of several Gbps are now feasible with current technology, offering a packet switched service at rates normally only associated with STM networks. Our consideration of other ATM based networks, rather than simply B-ISDN, also leads to the consideration of networks based on different technology, cost and management tradeoffs, which serve to extend the spectrum of solutions offered by ATM networks. Examples include DQDB, the 500Mbps CBN which is a network in everyday use between our two laboratories [CBN], and other future ATM LANs.

The flexibility of ATM comes from the fine grain of multiplexing present in the ATM layer, which allows a large delay-insensitive packet to be pre-empted by higher priority or time-sensitive traffic. While it is true that this allows ATM to provide an integrated mechanism to carry traditonal STM and PTM services, the ATM multiplexing can also offer a flexible interfacing mechanism to end-systems and hence a much richer range of possible services can be provided to applications.

3 MSNA.

The Multi-Service Network Architecture (MSNA) was developed to address both the naming and signalling required for the establishment of end-to-end ATM cell streams, and the integration of the in-band processing of cells with the systems present at the end points of communication links.

Our protocol architecture takes an internetworking approach as we believe that varieties of ATM networks will appear, depending on the functionality and data rate required, and the geographic area covered. An operational MSNA internet exists in Cambridge UK and spans the sites of Olivetti Research Ltd and the University Computer Laboratory. The physical network components consist of Fairisle ATM switches [FAIRISLE], slotted rings [CFR] and [CBN], and, in the near future, radio based ATM systems.

The Multi-Service Network Layer (MSNL) is the basic ATM service, giving an end-to-end stream of cells. It defines the MSNL address and circuit establishment techniques, including the mechanism for requesting various qualities of service (we are currently working on call acceptance and control algorithms to enforce the QoS guarantees).

Different ATM networks must present a common interface (MSDL) to the MSNL layer. During MSNL circuit establishment, routing is performed at each MSNL entity to select the appropriate MSDL instance for the next hop. The MSDL implementation for each network type then maps the MSNL circuit establishment mechanisms to those appropriate for its underlying network type; this can involve both address mapping and, in some cases, protocol mapping. Once the circuit is established, the routing of cells from one MSDL instance to another is normally performed by hardware. Where the two MSDL instances are identical, this is the normal VCI mapping function seen in ATM switches. Where they are different, the hardware is also required to translate the other fields of the header.

A particular design goal in MSNA is to minimize, if not eradicate, the layered multiplexing often found in communications systems. We wish to identify the end-system application entity with the minimum of overhead and hence invoke the normal resource allocation mechanisms of the end system (e.g. processor scheduling and buffer management) before higher-layer protocol processing. So, starting from the end-to-end ATM cell service of MSNL, where incoming cells are demultiplexed by use of the VCI, we try, as far as possible, to map a single application association onto a single MSNL circuit. Although we build adaptation, transport, session and presentation layers on top of MSNL for particular services, these provide no further multiplexing. This also fits naturally with some services, such as video, which implement all of the in-band processing in hardware.

4 Adaptation layer protocols

Our aim of producing an ATM internet, such that ATM is delivered into the real end system, means that our consideration of adaptation is concerned primarily with efficient implementation in end systems rather than as a means of interconnection of current packet or STM switched traffic types. However, by providing end-to-end service for cell payloads, MSNL can support any adaptation layer, in particular those being proposed by CCITT.

Within MSNA, and in keeping with the desire to eradicate unnecessary multiplexing, we have implemented only adaptation layers which do not perform cell level interleaving of different segmented blocks over a single virtual circuit.[1] When cell level interleaving is required, distinct VCIs are used. That is, we use multiple MSNL streams and allow the VCI to perform its first intended role of distinguishing them. In B-ISDN terms, this is equivalent to using a virtual path service where the end-to-end VCI bits distinguish cell interleaved blocks, rather than using a single VCI and sorting cells into blocks by yet another layer of multiplexing defined by the MID. In the MSNA architecture, the block level service is known as MSSAR (multi-service segmentation and reassembly) and is built on top of MSNL.

[1]Since MSNL is a service below the adaptation layer, MSNL can also support multiplexing adaptation layers, including the CCITT defined adaptation layers, or any other.

We have been using a particular adaptation protocol for a number of years, while investigating issues in adaptation [UDL].[2] We investigated the minimal information required to implement sufficiently secure yet efficient and simple to implement adaptation layers, and have now arrived, together with other researchers, at a proposal which requires that the ATM cell header now include a logical end-to-end user data bit[3]. The bit can be used to implement a range of different adaptation layers, but in this paper, we concentrate on adaptation layer protocols for data services.

For data services, as a starting point we may take the definition:

> *The fundamental role of the data adaptation layer is, at the transmitter, to take a PDU and provide segmentation into cells and, at the receiver, correctly reassemble the cells.*

The requirement for *correct* reassembly is generally necessary if the adaptation layer is to fulfil its nominal role of adapting ATM cells to the variable-length PDUs of conventional protocol stacks, but we question the requirement for accurate reassembly more closely in Section 5.1. The issue is a hardware versus performance tradeoff predicated on the desireable complexity of an ATM interface which performs or helps with adaptation in hardware.

A suitable data adaptation layer uses the bit to indicate last cell of a block, where the last cell includes a length field (or cell count) and a 32-bit CRC over all the cells forming the block [SEAL]. The length field is used by the adaptation layer to determine that the correct number of cells have been received.[4] The CRC detects the usual bit and burst errors, as well as most cell reordering while the length is required to detect certain cell drop outs where the content and position of the cell contrive to provide no contribution to the CRC. The CRC polynomial used in an ATM adaptation layer needs to be chosen with care, since if the CRC of $x^{i*384} + 1$ is zero, swapping two cells distance i apart will not be detected.

In addition, the length field in the last cell needs redundant protection, either with an explicit length check field, also in the last cell, or with some other sparse encoding. This is because lost cells may be comparatively frequent and a single bit error in the length field of the 1 to 0 type may compensate, resulting in the delivery of an incorrect PDU. Such a double error is not guaranteed to be detected by the CRC check since the lost cell may have contributed to the CRC residue in a way to counter the bit error in the length field. The chance of this particular event is 1 in 2^{32}, but there are other cases where more than one cell is dropped or cells are both repeated and dropped. In consequence, without a guard on the length field, the CRC would be reduced in strength to a level close to that of a 32 bit checksum. As explained in the next section, considerable effort is required to calculate a CRC-32 and it would be pointless to lose the potential benefit just to avoid the modest complexity of a sparse encoding of the PDU length.

[2]This adaptation layer, called UDL, places 16 bits of sequencing information in each cell, and is the one used in the performance example presented in Section 6.2.

[3]The December 1991 CCITT draft I.361 extends the payload type to three bits and defines this end-to-end user data bit. It is termed the 'ATM layer user indication'.

[4]A further length field, indicating the valid data section of the PDU, etc., can be placed in the PDU, say at the front, by transport or RPC higher level protocols.

We have measured the rate at which a workstation can calculate a CRC in software. Calculation of a 32 bit CRC on blocks of data on a 25 MHz Decstation 5000 Model 200 can be done at only 13 Mbit/second. This is using an 8 bit at a time method. (Using 16 bits at a time requires two undesireable 64K by 32 look up tables.) This suggests that assistance with the calculation of AAL CRCs is required in the interface.

5 Host and workstation ATM interface design.

A workstation interface for an ATM LAN should not require a host processor interrupt for every cell transmitted or received. Such interfaces thus differ from conventional LAN controllers (e.g. Ethernet) in that they must provide internal cell buffering for multiple received PDUs (cells). Simple FIFO buffering of the received and transmitted cell streams is sufficient to enable the host to operate asynchronously with respect to the network and also to reduce the interrupt rate below the one-per-cell level. This allows host interactions to occur at the 'packet' rate. An interface which restricts itself to this minimum hardware complexity is the current Olivetti Research Turbochannel ATM interface, described in Section 6.2.

Two ATM-specific functions may be readily incorporated into a more sophisticated ATM workstation interface. On the receive side, these are (1) the *sorting* of received cells according to their incoming VCI/VPI[5]

[6] and (2) the extraction and checking of per-cell adaptation layer check functions and protocol headers.[7] The inverse of these functions applies to the transmit side of the interface.

The following question arises: what applications require interfaces which should have, or need to have, this level of complexity? A high performance system may wish to have these functions implemented in hardware to relieve load on the main processor. On the other hand, for a low power mobile ATM station, several megabits-per-second can easily be handled by the CPU, avoiding hardware complexity. The advent of a low power, high performance ATM interface ASIC would probably be suitable for a range of interfaces, but restricted in the supported adaptation layer protocols. Equally, a workstation with a complex internal bus structure may not wish adaptation to be done in the network interface, instead it would route cells to various dedicated processing units. e.g. the framestore, a compression engine, disk controller or main store. Then the network attachment basically becomes an ATM multiplexer/demultiplexer.[8] An extension to this approach is to treat the set of internal interconnections within the end-system as an ATM switch [DAN].

[5]With certain protocol stacks, for instance when using AAL-4 or DQDB it is necessary to additionally sort on the MID field. In MSNA, MIDs, are ignored and there is no division of the header into separate VCI and VPI fields (c.f. Section 4). MIDs would only be present if non-MSNA ATM cells were encapsulated over MSNL.

[6]This sorting function is sometimes called 'fragmented reassembly'.

[7]Adaptation protocols based on a header flag, such as the one described in Section4 do not require adaptation layer headers in each cell payload.

[8]This approach is simplified when the cell header is used exclusively to route the cells to the appropriate entity, as in MSNA.

5.1 Cost and role of end-to-end check functions.

Returning to data applications, a necessary protocol function is guaranteed end-to-end integrity of PDUs. In OSI and TCP/IP, this function is implemented in the transport layer using the TP4 and TCP check functions. The cost of computing the check function in software has been recently widely debated. One view is that data generally has to be copied at least once by the processor, whether by the kernel or user-space code, or whether from a shared IPC buffer pool or from dedicated interface buffers. It is then of negligible cost to introduce simple checksum or CXOR [XTP] functions to the copying code. In addition, through suitable implementation, data which is not ever processed, need not be checked.[9] The extreme alternative view is that PDU integrity might be ensured by the network level, when connections with appropriate QoS are requested. The network level has knowledge of the error performance of each individual data-link along the route and so may be able to offer a *nothing-corrupt-if-delivered* QoS. This is the approach implicit in the CCITT AAL type 4, where per-cell checks are used.[10]

Several observations are easily made.

1. Within a reliable end-system, it is pointless implementing a transport level integrity check of the same strength (or weaker strength) than an already implemented end-to-end check at a lower protocol level.[11]
2. A CRC algorithm is the best type of data check, given that we are considering heterogeneous ATM networks, including radio and optical mobiles, where the error rates will typically be higher than in B-ISDN.
3. CRC calculation in software is too slow and CPU intensive for most applications (which is why weaker checksums have normally been used).
4. The network interface is a potentially sensible place to put CRC hardware.
5. If hardware in the interface is helping with the adaptation layer requirement that PDUs which are wrong owing to cell removal, repositioning[12] or replication are not (eventually) discarded, it must perform a check on the cells. This is either with per-cell sequence numbers or whole PDU check functions (e.g. CRC).

These observations lead to a number of possible solutions:

1. Do not check cells at the adaptation layer. Instead, pass up to the transport layer all cells received on a virtual circuit until the end-of-PDU indication is set, then check

[9]The MSNA approach of fully demultiplexing on the VCI and therfore avoiding layered multiplexing minimises the protocol processing which needs be performed at the same kernel priority as the process scheduler. This is of special benefit in real-time systems [TENNENHOUSE].

[10]We recognise that the CCITT would not call any of their adaptation layer services a network layer service, but from the MSNA point of view, the empty adaptation layer may be viewed as such.

[11]To avoid this duplication in practice, the transport protocol must be able to accept a 'data-ok' indication from the lower layers.

[12]Although CCITT has defined that B-ISDN should not reorder cells, we are considering all types of ATM system.

only at the transport layer. This essentially results in a transport protocol which is able to operate directly on concatenated ATM cell payloads. There is no reason why it should not also be able to operate and interwork with instances of itself on non-ATM data-links.

2. Implement a specified level of checking at the adaptation layer, then pass up the partially checked PDU to the transport layer software, where the remainder of the checking is done. This may in fact be a sensible approach if, as mentioned, for efficiency the transport entity can only do weak checking, such as checksum. (Checksum does not spot out-of-order cells.)

3. Fully reassemble and check each received PDU at the adaptation layer and then pass up only correct PDUs with the implicit semantic that they are checked and correct.

4. Calculate the CRC (or other check) of each cell payload in the interface and pass up this partial CRC along with the payload. The benefit of this is that higher level software is able to combine the partial CRCs to obtain a full CRC for the PDU at twelve[13] times the rate it would achieve by performing the whole CRC in software.

Similar issues where hardware implementation and protocol design cannot easily be decoupled can arise when attempting to place other higher-layer functions in the physical layer interface; encryption and presentation engines are significant examples. We do not discuss these issues here.

6 Implementing host and workstation ATM Interfaces.

Existing hardware implementations of ATM interfaces are now presented in Sections 6.1, 6.2 while enhancements are presented 6.3.

6.1 Previous ATM interfaces for workstations.

ATM interfaces for Unix workstations have been reported by Davie [DAVIE] and Traw [TRAW]. These are both high functionality interfaces. Traw's controller adapts ATM cells on a 155 Mbit/second SONET STS-3 carrier to an IBM RS 6000 microchannel and performs sorting of received cells in microcoded hardware. However it does not implement reassembly check functions. Davie's controller is for the DEC Turbochannel and is equipped with two Intel 960 processors to offer a variety of processing options. The controller will connect to four STS-3 carriers or one STS-12 carrier.

Simpler controllers have been built by Olivetti Research and Fore Systems for Turbochannel and S-bus respectively. These do not perform protocol processing, except for the ATM header check (HEC-8), and in the case of the Fore Systems units, the payload AAL-4 CRC-10. Neither interface maintains state information at a granularity larger than a cell.

Design and performance of the ORL interface is now presented.

[13] Assuming a 32 bit CRC, there are twelve 32 bit words in each cell payload.

6.2 Olivetti Research simple Turbochannel ATM interface.

The ORL controller, known locally as the 'Yes V2' interface, employs a physical layer using the AMD TAXI transmission devices at 100 Mbit/second. The physical layer cell format accords with a *de facto* standard employed by the research divisions of several companies and universities. The controller has separate receive and transmit FIFOs, each able to hold 76 cells. A further 'token' FIFO which is used merely as an up-down counter to keep track of the number of cells completely received. Host data transfer is either through programmed IO or by DMA. DMA transmissions of up to 39 cells are supported by the current Turbochannel host systems, which are Decstation 5000s. DMA transfers may either be used to copy a block of memory which has already been formatted by the processor with the appropriate ATM cell header inserted every 13 words, or in conjunction with programmed IO, where the processor reads or writes the ATM header word directly to/from the FIFO and then initiates a DMA transfer of the cell payload.

A receive interrupt can be generated on each cell received, or alternatively, on the arrival of cells with the header indication flag set. Using this latter mode, a PDU may accumulate in the receive-side FIFO until its last cell has been received, and then the host interrupted. Of course, the host may find other cells in the FIFO from incompletely received PDU's on other associations, in which case it must sort and store these as usual. However, these cases will be rare for many applications [MOGUL] and, in all cases, the overall frequency of interrupts may be reduced. Transmit side interrupts are not yet required in the Unix device driver, since, as explained shortly, we have found that programmed IO is slower than the 100 Mbit/second available transmit bandwidth and so the transmit side does not go busy. However, transmit interrupts are available on TX FIFO less than half full and on end of DMA.

An example of the performance of this simple controller using MSNA under Ultrix on a 25 MHz Decstation 5000/200 is shown in Table 6.2. The experimental system consisted of a single workstation with the MSNL protocol in its kernel and a Turbochannel controller, looped back on itself using a short length of cable. Two Unix processes were involved, a sender and receiver process. The sender process sends a PDU of 70 cells through its MSNL socket and out through the controller. Since the controller cable is looped back, the receiving side of the interface directly receives the PDU. The receiver passes the cells up through another MSNL socket to the receiver process. The receiver process then sends a single cell ACK message in the reverse direction. The process then repeats. The repetition of the process gives a stable display on an oscilloscope for taking measurements.

The resulting throughput of 34 Mbit/second for transmit and 9 Mbit/second for loop-back show that the workstation delivers respectable performance, despite using only programmed IO and the most simple type of ATM interface.

Using simple cell DMA transfer the device driver transmit speed increases to only 61 Mbit/second, as shown in Table 2. This is for large blocks which use Ultrix 'cluster mbufs'. It is clear that the throughput is still limited by the interface and not the protocol code.

Although we have not yet measured a figure, receive side speed will benefit even less from per-cell DMA. This is because the protocol overhead for reception is greater; the processor

Component of test	Time	$70 \times 8 \times 48$/Time
Time to fragment a block into 70 cells and write into the controller using programmed IO.	792 μs	34 Mbit/s
Time to locate reception mbuf, check the headers of 70 cells and copy the data into the mbuf using programmed IO.	1485 μs	18 Mbit/s
Time for non-device driver software, including (1) MSNL and socket processing for both transmit and receive of 70 cells, (2) a user space context switch between sender and receiver process and back and (3) user space processing consisting of sending a single cell ACK MSNL message in the reverse direction.	850 μs	–
Totals (loopback throughput)	3125 μs	9 Mbit/s

Table 1: Example performance of the simple ORL ATM interface under Ultrix. This result is for user-space to user-space loopback on a single machine using programmed IO only.

Component of test	Time	4096×8/Time
Time to allocate a cluster mbuf of 4K bytes containing 86 cells worth of data.	39 μs	420 Mbit/s
Time to 'bcopy' 86 cells from process space to the mbuf	238 μs	276 Mbit/s
Device driver time, including a progio header write and a DMA payload transfer.	3 μs	128 Mbit/s
Totals (transmit throughput)	535 μs	61 Mbit/s

Table 2: MSNL kernel and device driver single cell DMA transmit throughput. This result shows the improvement in transmit performance when using programmed IO for the cell headers and DMA to transfer just the payload of each cell.

must always look-up the context of every received cell, regardless of the mechanism used to copy the data. We are measuring receive performance at the time of writing.

6.3 Worthwhile functionality in the ATM interface.

We have shown that an interface which simply avoids interrupts on a one-per-cell basis is not sufficient for a 100 Mbit/second ATM host interface. Using DMA, for each physical layer PDU (cell) increases the performance of the interface over programmed IO, but does not free up the processor for other activities, contrary to the typical case when using DMA with lower speed and non-ATM interfaces.

The approach of advance formatting an mbuf in cells enables longer DMA transfers to take place.[14] An enhancement is the use of a VCI register in the interface, where the VCI and other header fields are inserted automatically after the DMA transfer of every 12 words. DMA transfer of received cells should be the reverse, and continue until a cell whose VCI[15] does not match the register's contents is encountered. The processor must then re-program the interface's registers (DMA address, length limit and VCI register) with the context of the new cell. An interface for the DEC Station, using this approach, should easily be able to transmit at the 100 Mbit/second rate. Reception will only be significantly slower if cells from separate associations are highly interleaved.

The incorporation of CRC generating and checking hardware is vital for high performance interfaces. Since we intend to use the new adaptation layers which employ CRCs, such hardware must be deployed in all of our future interface designs. It is envisaged that the method of computing partial CRC's for each received cell, with its twelve times speed up, will not impact on the 100 Mbit/second target.

It is not clear that any further worthwhile features can be added to an interface unless it implements the sorting function. A sorting interface may either accumulate cells in its own local memory, or by direct insertion into the main host memory. An advantage of the former approach is that a CRC may be calculated without interruption when data is copied once from the controller's RAM to host RAM. Advantages of the second approach are intrinsically shorter bus holding times and 'cut-through' where data is ready in host RAM as soon as possible.

Sorting interfaces may only be able to support a subset of the available VCI space since it is desirable to use directly-mapped RAM to look-up VCIs. In all of our current MSNA data-link services (our ATM switches and rings), this problem is ameliorated since receivers are able to allocate their own incoming VCIs. If this is not possible when directly connected to B-ISDN, then it should at least be possible when using a local ATM LAN. The ATM LAN may be connected to B-ISDN at one or more points. The entity at the B-ISDN to ATM LAN boundary can then use associative techniques if necessary.[16] An interface which can support 64000 active VCIs is deemed sufficient by [DAVIE], and we would agree.

[14]The numerical result for this approach should be incorporated in the final version of this paper.

[15]The VCI, VPI and MID must be compared in non-MSNA systems.

[16]The B-ISDN switch manufacturers are faced with the same VCI/VPI look-up problem, so are probably able easily to control VCI space. The question is whether they will make VCI non-sparseness guarantees to the customer.

Custom and semi-custom VLSI can be used to implement an ATM LAN controller. The device would connect to OEM host bus controllers, such as those available for EISA, Turbochannel, VME etc. and to inexpensive fibre-optic physical layer devices, such as the AMD FOXI parts. A full ATM interface ASIC will perform sorting, adaptation, and probably buffer management of local RAM and scatter-DMA in host RAM. An example specification is available in [AIA].

7 Future directions.

The ever falling cost of VLSI implies that ATM specific host interface controllers will be available in the near future. The uncertainty over the specification of such ASICs is decreasing through increased cooperation among computer manufacturers. Users of private ATM LANs can employ an MSNA approach where the simple semantics of an MSNL connection enable users to insulate themselves from variations in B-ISDN specifications. Yet they may still achieve wide-area interconnection using the empty adaptation layer offered by B-ISDN. [DJGDM].

References

[AIA] 'Specification of a Workstation ATM interface ASIC' by David Greaves. Document OSI95/ORL/TN/B3/7/1 from Esprit project OSI 95.

[BBN] 'BBN unveils broadband strategy with Emerald switch' in 'Communications Networks' October 1991. Also BBN Communications press release, available from the company.

[CFR] 'The Cambridge Fast Ring Networking System.' A Hopper and RM Needham. IEEE transactions on computers, Vol 37 no 10. October 1988.

[CBN] 'The Cambridge Backbone Ring.' David J. Greaves, Andy Hopper and Dimitris Lioupis. In proceedings of IEEE Infocom 90, San Francisco 1990.

[DAVIE] 'A Host-Network INTerface Architecture for ATM' Bruce S Davie. Bellcore. Proceedings of SIGCOMM 91 Zurich, September 91.

[DJGDM] 'Private ATM networks' DJ Greaves and D McAuley. Presented at IFIP 3rd International Workshop on Protocols for high speed networks, Stockholm, May 1992.

[FAIRISLE] 'Fairisle: An ATM network for the local area' Ian Leslie and Derek McAuley. Proceedings of SIGCOMM 91 Zurich, September 91.

[MOGUL] 'Network Locailty at the Scale of Processes' Jeff Mogul. Proceedings of SIGCOMM 91 Zurich, September 91.

[SEAL] 'Simple and efficient adaptation layer (SEAL)' Tom Lyon, Sun Microsystems. Document T1S1.5/91 292.

[TENNENHOUSE] 'Layered multiplexing considered harmful.' David Tennenhouse. Proceedings of IFIP WG6 International Workshop on 'Protocols for high-speed networks' ibid.

[TRAW] 'A high performance host interface for ATM networks.' Brendan Traw and Jon Smith,

U. Penn. Proceedings of SIGCOMM 91 Zurich, September 91.

[UDL] 'Cambridge HSLAN protocol review.' DJ Greaves, ID Wilson. Proceedings of IFIP WG6 International Workshop on 'Protocols for high-speed networks' edited by H Rudin and R Williamson, held at IBM Ruschlikon 1989. Elsevier 1989.

[XTP] 'XTP protocol definition revision 3.6' Protocol Engines Inc. 11th January 1992.

This work was partly funded by Esprit project OSI 95: High performance OSI protocols with multimedia support on HSLANS and B-ISDN.