ATM Network Services for Workstations

Document identification OSI95/B3/Book/v1
DJ Greaves*& D McAuley
Olivetti Research Ltd.

1993-01-21 (Jan 93)

Abstract

Workpackage B of OSI 95 was titled 'New Communications Techniques' and was an evaluation of how to make use of the new communications techniques which offer services with a guaranteed quality of service, including ATM and B ISDN. This document discusses the provision of the ATM networking services to application programs running on general purpose computing equipment which is connected to an ATM network.

Section 1 of this document asserts that ATM may either be integerated into end systems or else be used as an encapsulation technology. Section 2 describes how and why ATM technology should be used in the local area. Section 3 describes the essential component of service common to all ATM systems. Section 4 describes the Multi Service Network Architecture (MSNA) approach to ATM integration. Section 5 describes a set of logical link control primitives for access to ATM. Section 6 describes the design space for ATM host interface hardware. Greater detail for each topic may be found in the related OSI 95 'deliverable' documents.

1 How may we use ATM ?

A new transport protocol may either use existing network layer services or may be defined in terms of new network layer services. The work within Work

*Email: djg@cam-orl.co.uk

package A of OSI 95 which developed the TPX transport protocol specification has adopted the former approach, using the standardised OSI connectionless network service [ULG 4].

The MSNA (multi service network architecture) approach, described herein and in [MAC] takes the alternative route of offering a new network layer service. Within MSNA, ATM virtual circuits are brought into the end systems and terminated at or above the network layer service access point. This opens the way for a transport protocol which treats the ATM layer as a connection oriented network layer service.

Recently, within the Internet Engineering Task Force and also within the highly regarded discussion group *atm@sun.com* there have been proposals for using network layer addresses as the termination points for ATM virtual circuits. This is in keeping with MSNA.

1.1 Encapsulation versus Integration

ATM may be used as the bearer service for encapsulation of existing protocol stacks. For example, the Bellcore SMDS service offers encapsulation and, in the future, will operate over ATM. In addition, a draft RFC 'Multiprotocol over ATM Adaptation Layer 5' by Juha Heinanen specifies encapsulation formats for virtually all existing protocol stacks.

Task B3 of project OSI 95 has produced interfaces for ATM end systems and therefore has had the chance to introduce ATM specific lower layer protocols into the end systems. The MSNA approach to this has been introduced in OSI 95/ORL/Deliverable 1.

Encapsulation of older protocols, such as IP, cannot offer new services. For example, IP does not support a large set of quality of service options and those which are specified are not associated with guarantees or always implemented. This situation has been described by Bob Metcalfe as 'the worst of both worlds' because the overheads of an ATM subsystem are present but the advantages of ATM are not being realised. In order to support new applications, such as real time traffic, it is sensible to concentrate on integration of ATM with the existing protocol stacks. We leave the field of encapsulation to those who are unable to modify their end systems. This is addressed within OSI 95 in [BELL 1].

2 ATM in the local area.

There is no point supporting multiple classes of quality of service within the international ATM network or local ATM backbones unless these separate qual ities of service can be presented to the individual application programs loaded into an end workstation. This consideration is addressed in an approach where the ATM virtual circuit is terminated inside individual applications and there is no multiplexing of multiple applications onto a common virtual circuit.

The arguments which convinced the CCITT to recommend ATM as the solution for Broadband ISDN also operate in the local area for private multiservice net works. Research work to establish this point has been carried out within OSI 95 at Olivetti Research Ltd. Increasingly, computer manufacturers are promoting the use of ATM techniques in privately owned and operated digital networks.

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 adds two further, consequential advantages:

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

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, in terms of the service offered to the end systems, since in ATM, headers can be manipulated by each switching entity, while the payloads are passed unaltered from one point to another. This implies that network manufacturers can use an optimum cell header for the number of virtual circuits and particular media characteristics that their equipment supports. An example is a current Olivetti Research ATM radio project, where sequence numbers and a MAC layer response field are put in the cell header.

[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.

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 has superior real time performance 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.

3 The ATM service.

The definition of 'ATM' from the CCITT recommendations is: 'the use of a fixed length cell as the primary means of information transfer where the peri odicity of cells is not known by the receiver in advance, but it is indicated by a circuit identifier in the cell header.'

More specifically, any equipment which can offer the CCITT ATM layer service is a suitable component for a private ATM network. The ATM service is the ability to transparently ship ATM cell payloads along a virtual circuit from one ATM layer SAP to another while preserving order. A result of our work is the realisation that the implementor of an ATM interface for a host is free to use any techniques which meet the ATM layer specification. Naturally he will also benefit if the lower and higher layers of his implementation are a CCITT or ATM Forum standard, but he may have good reason for using alternative techniques.

Private ATM networks may be interconnected over the public B ISDN network without an in band processing overhead provided the B ISDN service is access able at the ATM layer. The cost of a virtual circuit is likely to be greater within the global public network than in a private network which may only span a sin gle site. Therefore translation between signalling and addressing formats, along with other management functions, will probably have to be implemented at the gateway.

4 The MSNA approach to multiplexing.

The Multi Service Network Architecture (MSNA) protocols are used at Olivetti Research Ltd and at the University of Cambridge Computer Laboratory. Al though the two sites internally employ and are interconnected by a variety of networks, the MSNA platform offers a homogeneous 'ATM Internet'. MSNA also provides transparent inter operation between the ATM and non ATM style

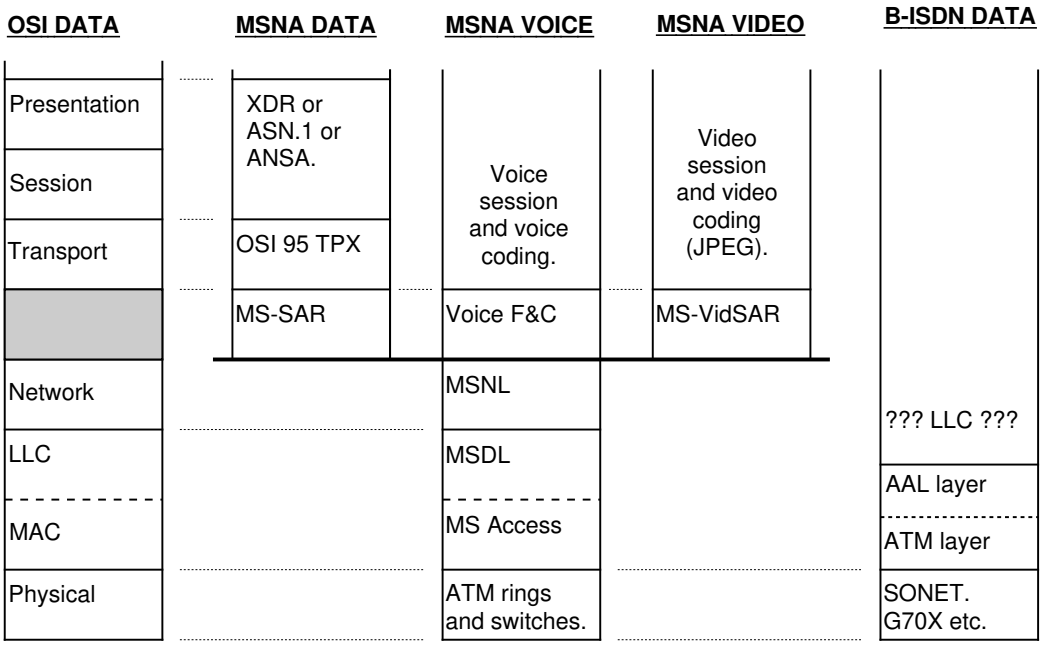

Figure 1: Functional mapping of MSNA to the OSI reference model and CCITT B ISDN reference model.

networks to support access to devices and machines situated on Ethernets and other non ATM subnetworks. MSNA is designed for efficient implementation in software, with or without hardware support. It makes efficient use of transmis sion bandwidth, allowing many types of transaction to fit into a single ATM cell. Management and connection procedures are conducted 'out of band', allowing implementation of data paths in hardware. Most importantly, despite providing interoperation between different classes of computer network, it attempts never to compromise the capabilities of the underlying network technologies.

Current CCITT work, regarding the integration of ATM with the OSI 7 layer model, effectively hides ATM specific features below the LLC service interface. The current 802.6 MAN protocol stack takes a similar approach. As will be shown, the MSNA approach is radically different, especially with regard to the semantics of its link layer connection and its elevated position of the segmenta tion and reassembly function. See Figure 1.

The most basic MSNA function is the interconnection of MSNA service access points (MSAPs). These are identified by a 64 bit MSNL (Multi service network layer) address. They are basically a concatenation of a conventional network layer address and an application port number. Owing to the separation of con trol and data within ATM, these addresses need never appear in the header field of a protocol data unit. They are communicated in the data fields of man agement PDUs. The motivation for MSNA stems from the desire to eliminate multiplexing outside the physical layer.

4.1 'Layered Multiplexing Considered Harmful.'

Many conventional protocol implementations, including OSI and TCP/IP, of fer provision for multiplexing at multiple points in their layered architecture. By *multiplexing*, the process of tagging component SDUs from several streams with a different stream identifier and then merging the resulting streams is implied. One aim of MSNA was to avoid unnecessary multiplexing. Tennen house summarised the motivation for this approach in his short paper, *'Layered Multiplexing Considered Harmful.'* [TENNENHOUSE]. The unnecessary and harmful effects of layered mutliplexing in a connection oriented ATM environ ment are reiterated in this Section. However, multiplexing and demultiplexing are vital functions, if only to enable multiple streams to share one network, and so multiplexing and demultiplexing are inevitably required at certain points in the architecture. MSNA attempts to confine this function to a single point, such as the ATM switching layer.

To give a feel for the argument, we quote from Tennenhouse's paper:

> The principal advantage of layered architectures is that they pro vide for the *step by step enhancement of communications services* [TEN 1]. In theory, each service boundary between adjacent layers identifies a stage in the enhancement process. In order to minimise the duplication of functionality across layers, the network architect should *collect similar functions into the same layer.*[2] In the case of the multiplexing function, this principle has largely been ignored. For example the OSI architecture presently provides for multiplexing within six of the seven layers of the protocol stack.[3]
>
> It is claimed that the extensive duplication of multiplexing func tionality across the middle and upper layers is harmful and should be avoided.

Let us consider, as an example, the layers of multiplexing present when con ventional TCP/IP is run over an Ethernet. Inevitably there is multiplexing of packets from different sources and destinations over the physical medium of the Ethernet. The corresponding de multiplexing function is performed on a MAC layer address basis by the interface hardware. Please note that there is only one MAC layer address for a station and that this is fixed for the lifetime of the station. (As will be shown, a different approach is possible in an ATM system.) Conventional LANs of this nature, the Ethernet being no exception, generally include an additional source address field in the MAC layer header. Note that the source address field is not used by the media access hardware and that the destination address field has no further role after the message had been received by the hardware. In an ATM architecture, these fields can be compressed into the virtual circuit identifier field.

Encapsulated in the MAC data field is the IP (Internet Protocol) header. This is usually 20 bytes in length and contains a 32 bit IP source address and a 32 bit IP destination address. Encapsulated in the IP data field there is a TCP header. This contains a 16 bit source port and a 16 bit destination port addresses (NSAPs). These identify the network layer service access points. Still further levels of layered addressing can often be found: for instance some RPC systems multiplex their own logical connections over a single NSAP. If we only consider mutliplexing in and below the transport service layer, by examining the TCP and IP headers, we find $2 \times (48 + 32 + 16) = 192$ bits (24 bytes) of addressing associated with every message. This is just to support a single end to end TCP byte stream.

As Tennenhouse stated, a justification for layered protocols is provided by the

[2] Principle P4 guiding OSI layer determination [TEN-2] (Tennenhouse's footnote).

[3] The presentation layer is the sole exception. However, presentation address selectors have been incorporated into the naming and addressing scheme on the grounds of *architectural consistency* (Tennenhouse's footnote).

step by step enhancement principle. However layers of *multiplexing* below a particular service layer boundary are intended to be completely hidden and therefore to offer no perceived enhancement of service. Indeed, hidden multi plexing can sometimes manifest itself as *crosstalk*, causing interference between the multiplexed streams. By 'crosstalk' we mean that the service within one logical instance of a protocol stack within a machine experiences interference, mainly in the form of processor service jitter, resulting for messages arriving for other instances of the protocol stack. With conventional protocol architectures, considerable processing of header fields is required before the recipient and the recipient's QoS priority can be identified. The effectivenes of process scheduling decisions is thereby diminished.

On the other hand, worthwhile enhancements of service is provided by certain of the algorithms at end and intermediate network points and their associated fields in the headers. An example is flow control. Equivalent functionality to these services must be found or preserved in a new protocol architecture which is avoiding multiplexing.

4.2 Layered protocols in a connection-oriented environment.

If a way can be found to delete the sub addressing fields in the protocol headers, the remaining information bearing fields can be divided into two catagories : those which are of constant value (the same value for every message sent) and those whose value varies each time. Protocol identifiers are examples of the first catagory and checksums are examples of the second. In a connection oriented environment, fixed circuit parameters, including quality of service requests and protocol identifiers can be properties of the connection and then need not be sent every time.

These connection attributes can be permanent properties of a permanent cir cuit, requested properties of a dynamically set up circuit, or negotiated proper ties, dependent on the networking resources available. MSNA uses connection oriented *lightweight virtual circuits*, and the MSNA architecture supports any of these approaches to circuit establishment. A connection oriented system has advanatages for delay sensitive traffic in a multi media environment, since the probability of adjacent messages between the same source and destination being routed along different paths is much reduced. Fixed routing is almost certainly required for jitter sensitive, multi media services, and therefore has to be pro vided in MSNA to achieve its multi service goal.

Returning to the Ethernet/IP/TCP protocol combination; a brief calculation shows that if it were possible to delete the addressing and constant value fields,

the header length would be reduced from 52 bytes to about 15 bytes.[4] This is of great interest considering the ATM cell size of 48 bytes.[5][6] As stated, at least one level of addressing is always required, and when MSNA is run over an ATM substrate, MSNA uses the cell VCI[7] in the cell header for this purpose.

Through the deletion of the fixed fields from the low level protocol headers, and by concentrating the multiplexing function at a single point just above the media access layer, it is clear that the size of the smallest network layer service PDU, including a few bytes of actual data, can be reduced to a single cell. When RPC and transport layer services are overlaid, including optional segmentation and reassembly protocols, many types of useful application traffic still map into single cell messages.

To summarise, the principle advantages of the points outlined so far are: the reduction of protocol header overhead on the network, the reduction of process ing time needed to generate and check headers, and the confinement of crosstalk between multiplexed traffic streams.

4.3 Further aspects of layered multiplexing.

When intermediate layers of multiplexing are eliminated, essentially we are left with multiple instances of low level protocol stacks, one for each virtual circuit. These, of course, do not need to be the same, but may be chosen to suite the traffic type in use. For high speed operation, the individual protocol stacks can be implemented in one monolithic section, rather than separate, layered software modules. All state variables of a connection can then be held in one activation record, rather than in several different places, reducing the number of time consuming look up operations per message. Monolithic software can usually be faster than modular software.

Further benefits accrue when we consider the implementation of a protocol stack

[4] This estimate is obtained by counting the following fields only. These are the IP time-to-live (1 byte) and length (2 bytes) fields and the TCP source and acknowledgment sequence number fields (4 bytes each), and the checksum and window fields (2 bytes each). The IP fragmentation facility has been ignored in this account since MSNA supports fragmentation using MSSAR above the MSNL layer.

[5] Cells of length 32 bytes are used by several of the networks constructed in Cambridge. When appropriate, the length of a cell to be used on a circuit is an MSNL liaison attribute.

[6] It may be argued that the 52 to 15 byte comparison is not strictly valid, since the figure of 52 bytes includes the two 48 bit Ethernet MAC addresses, where if all of the multiplexing is concentrated into the VCI, the cell header length should also be counted. The discrepancy arises since the Ethernet MAC layer address cannot be used as flexibly as a directly interpreted VCI on an ATM network.

[7] In this document, there is no distinction between VCI and VPI (virtual circuit and virtual path identifiers), and since a general principle of MSNA is that there should only be one multiplexing mechanism, a distinction is not desireable.

in a multi threaded environment. When a single processor is servicing mutiple client processes or threads, scheduling decisions must be performed efficiently, fairly and with regard to minimising context switching rate, and therefore over head. Inefficiencies arise if incoming messages have to be partially demulti plexed before their recipient process can be identified. This also precludes exact resource accounting for scheduling purposes. MSNA avoids this by eliminating intermediate multiplexing points, thereby enabling the correct recipient thread to be identified as soon as a message arrives and requiring no subsequent context swaps as 'port' sub addresses are uncovered.

As stated before, the use of separate instances of a protocol stack for each logical connection also enables heterogeneous protocol stacks to be used if desired. An example is that not every connection will require segmentation and reassembly information. An obvious case is where only single cell messages are to be used on a connection. This may be the case for acknowledgement cells going in the reverse direction to a foward circuit, or for multicast messages which are placed in a single cell to ensure atomic reception. Cutting out the SAR layer obviously improves performance, both in terms of processing overhead and payload usage. Voice is another example: a time stamp is probably more appropriate than SAR information.[8]

In a multi processor or multi media workstation, it is convenient for the hard ware to demultiplex incoming messages according to which processor, or frame store, or whatever, they are destined for. For instance, some streams of traffic may require to be routed through decompression or de encryption hardware, while others may not. Again, each of these types of traffic will generally treat the cell information field in a different way, and therefore require a separate protocol.

Within a uni processor system, it may be argued that a direct implementation of an entirely vertical protocol stack shifts multiplexing complexity from the networking software to the process scheduler. It is certainly true that the lower levels need to operate on multiple PDUs in parallel. For reception, this is necessary when multiple higher level blocks are being reassembled in parallel, and it is also required on the transmit side when there is a rate limitation on particular VCIs, or to avoid head of line blocking in general.

[8] The provision of a *set* of CCITT recommended adaptation layers is intended to address these varying requirements.

5 LLC Primitives for an ATM interface

In this Section we give a brief consideration to the formal primitives for an ATM interface which would approximate to the LLC level of the OSI reference model. The primitives offer service to both the in band network layer and the out of band protocol layer for operation and maintenance.

We consider the connection control plane's protocols to be implemented below the offered primitives. This is appropriate since the compexity of connection establishment lies in end point naming, which is above the LLC layer, and in bandwidth reservation databases, which (logically) lie outside the host: they lie in the ATM interconnection fabric.

For the in band data, multiplexing may be avoided by using a null connection oriented network layer on top of these LLC primitives (as in MSNA).

5.1 OSI LLC situation.

The ISO/IEC 10039 MAC service description defines an abstraction of the MAC definitions 802.3, 4, 5 and 7 [OSI MAC]. Only connectionless service is specified in the 1990 draft. The MAC services provided in the abstraction are:[9]

- Independence from the underlying MAC and physical layer, except of course, in terms of quality of service.
- Transparency of transfered information, except there may be a byte limit on PDU size.
- Priority selection the MAC service makes available to MAC service users a means to request the data at a specified priority.
- Addressing the MAC service allows the MAC service user to identify itself and to specify the MSAP (MAC layer service access point) to which data is to be transfered.

The PDU transferred is termed the *unitdata object*. The MAC service is allowed to

- discard objects, and
- change the order of the objects

[9] This list is slightly condensed from Section 6, 'Overview of MAC service', from [OSI MAC].

and exhibits a negligable rate of

- object duplciation
- reordering of objects of a given priority.

The relative rates of occurrence are assumed to be known *a priori*. The service specification states that the receiver is not able to influence the speed of the transmitter, although it does not discuss rate control at the transmit side between the MAC and the MAC transmit side user. MSAP addresses are defined for broadcast and group addressing and it is stated that the received addresses must be the same as that transmitted.

5.2 An ATM LLC primitive set.

In an example ATM LLC, in order to open a virtual circuit, we might use the primitive

```
lid = vc_open(ADDR destsap, QOS qos);
```

where a negative value of lid indicates failure. We assume lid is a local identifier for a local virtual circuit control block (descriptor). The fields in this control block are established using signalling protocols. The most important value held is the VCI and VPI to be put in the header of all cells sent on this virtual channel.

To close a virtual circuit one may use

```
vc_close(lid);
```

The quality of service field includes the maximum rate to be used on the virtual circuit and other parameters such as the expected average rate of cells and whether cells should be dropped or delayed when unavoidable.

Outgoing rate control will be parameterised within the QoS values supplied in the open primitive and implemented below the LLC layer by the interface or device driver.

In order to send data we may either use LLC primitives with the AAL underneath or we may have LLC primitives which have direct access to the ATM layer. In this example set of primitives, both types of access are available according to which primitive is invoked.

Transmission of a cell might be done as follows

```
send_cell(lid, char *data);
```

where data is a pointer to a cell payload data unit (held in 48 bytes of consecutive memory), and receive might use

```
rc = receive_cell(lid, char *data);
```

which is a blocking call which returns when one cell has been read. The return code is available for error reporting.

Where the primitives automatically perform segmentation and reassembly using AAL 5 or AAL 3, we must extend the two primitives just presented with the addition of a length field

```
send_block(lid, int length, char *data);
rc = receive_block(lid, int length, char *data);
```

where length is the number of data bytes to be transmitted or the buffer length available for reception. The return code is available to report errors and the actual size of the data unit received.

5.2.1 F5 Operation and Management flow messages.

Cells of the F5 flow of a virtual channel are identified by a value in the cell header type field. Such cells may only be sent and received once their virtual circuit is established. Access to the F5 flow at the LLC boundary may be offered using primitives similar to those for normal data. The local managment plane entities may either use these primitives or the ATM management may be considered to lie below the LLC boundary. In any case, the device driver's interface to the hardware is the same as for in band data, except for the need to produce the alternative payload types.

5.2.2 Cell loss priority.

The CCITT standards have not finalised how congestion control and cell loss priority should be handled by the ATM adaptation layer. It is reasonable that our LLC primitives ignore the CLP flag of received cells and set the outgoing CLP flag according to the AAL standard. Therefore CLP does not appear in the primitives.

5.2.3 Congestion indications.

Received cells indicate congestion through values in the payload type field of the cell header. Congestion notification can be passed up to transport level entities (including VBR video compression entities etc) in order for them to modifiy their sending discipline. Alternatively, congestion may be considered an ATM management issue and the use of congestion indications can be restricted to modifying future connection acceptance and routing decisions.

Current CCITT working documents recommend that the congestion indications may be available at the AAL service layer but the mapping onto cells is not agreed. A sensible approach may be for the AAL congestion indication to simply reflect the congestion indication field of the end of message cell. A host interface may easily provide this and the the LLC primitive return code may be extended in range to include it.

5.2.4 Generic Flow Control.

When the interface is connected to media which employs generic flow control, the generic flow control protocol may be implemented entirely below the LLC layer, probably in the interface hardware. Lack of throughput for outgoing cells may be detected from an increase in the outgoing queue length and notification passed up through the congestion indication primitives.

5.3 MSNL and connection set up.

McAuley defined Multiservice Newtork Layer (MSNL) as follows [MAC]. The multi service network layer is an interworking service which is based on the idea of MSNL *liaisons*. An MSNL liaison is a *lightweight* concatenation of MSDL associations. There are three important aspects of MSNL:

- it defines the MSNL addresses,
- it defines association and liaison set up procedures,
- it does not multiplex its liaisons over the MSDL associations.

MSNL provides an out of band connection establishment mechanism. Since MSNL liaisons are not multiplexed over MSDL associations, MSNL does not require in band protocol headers in the service units, so MSNL introduces no processing overhead on the data path and it provides the same data interface as an MSDL association.

Defining the MSNL connection as *'lightweight'* means that the resources allo cated to the connection are neither to be thought of as valuable or permanent. As discussed in slightly more detail later (e.g. Section 5.5), MSNL connections may be unilaterally de allocated, or fail in other ways, in which case MSNL layer software may provide re establishment without explicit interaction with higher layer software. On the other hand, MSNL users cannot always be expected to explicitly close connections (for intance they might be unexpectedly re booted), and so the existence of garbage collection mechanisms is assumed.

An MSNL liaison is established between two MSNL SAPs (MSAPs). These are unique and are allocated from the 64 bit global address space.[10]

A host computer may have many MSAPs (loosly corresponding to the con ventional 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 hierar chial. In [MAC], 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. MSDL does not have a mechanism for naming peers before an association is set up, so cannot directly perform this function. The MSNL address provides a naming mechanism, and so provide the basis for association and hence liaison set up.

5.4 Promiscuous MSNL.

For start of day reasons, MSNL must be able to contact at least one manage ment service before any connections can be established.[11] This management service is contacted using *meta signalling* consisting of idempotent MSDL mes sages on one of a set of *well known* VCIs. The receiver of these messages must be prepared to accept messages from any source. This is known as *promiscuous MSNL*. Messages sent on such a VCI are not part of an association.

This initial management entity may either be a fixed machine, offering the start

[10] The initially adopted approach was to base 32 of the 64 bit address on IP addresses. This provided a convenient unique identifier space. Increasingly we have MSNL entities, such as ATM cameras, which do not have an IP address, so MSNL addresses have now become less tightly coupled to IP addresses.

[11] Permanently established associations do not require this, but then they do not have a 'start-of-day' case.

of day services, or if the VCI is in fact a multicast address, this basic service may be provided in a distributed manner by a number of machines. No state is retained for incoming messages on a promiscous association. Such messages must be idempotent and fit into a single MSDL PDU (cell).[12]

Once a mechanism for the start of day problem is provided, arbitrary levels of indirection can be inserted before the service actually wanted is reached, although there is no advantage to excessive complexity. Of course, the most important service is connection establishment.

At least one level of indirection is sensible for reliable connection establishment. A connection involves allocation of VCIs from VCI space, if not the reserva tion of other resources, such as bandwidth. Allocation of resources cannot be done with idempotent semantics, hence the requirement for the establishment of a signalling connection. The last level of indirection in the chain generally becomes the only one frequently used, since sensible implementations are able to cache earlier results. In practice, this means that a station has signalling MSNL connections to management entities which are able to establish further connections to related MSNL addresses, or else provide further indirections to further management entities. In the current implementation, the hierarchial structure of the MSNL address is used along with mask fields to help with this. The concept of 'caching' the earlier enquiries, rather than retaining hard state relating to connection management entities, provides intrinsic adaption to net work reconfigurations. If a cached value is found inadequate, the concept of an 'authorative' inquiry can be used, which will propagate backwards through the network, ignoring cached responses, until the source is found.

5.5 Connection closure and time-out.

Liaison closure can either be performed explicitly by higher layers at the end points, or autonomously, when any intermediate entity wishes to reclaim re sources from an old or under used association. Entities performing a close have the opportunity to inform other participating entities of impending closure. There is also the possibility that connections can be re routed or re established by the MSNL layer, while minimising interruptions to the provided service. Further aspects of liaison set up, routing amd closure are presented in [MAC], however, in this report it is only neccessary to concentrate on the ATM and LLC specific features of MSNA.

[12]CCITT draft recommendation I.311, 'B-ISDN General Network Aspects' defines the *meta-signalling channel*. This provides a start-of-day service which is available before signalling channels are allocated. As in MSNA, this channel is situated on a well known VCI (VCI=0x00001) and uses idempotent, single cell messages.

6 Design Space for ATM host interfaces

This Section considers the implementation space for an interface, concentrating on the trade off between protocol implementation in the host against within the interface.

An ATM host interface receives ATM cells from the physical layer and, in con junction with the device driver, allows processing of their contents. It also transmits cells onto the physical layer.

This Section limits itself to data oriented applications and to systems with a conventional host bus architecture. Multi media traffic carried over data adap tation layers, which is a likely situation for workstation videoconferencing, is covered implicitly.

6.1 Four major parameters

Four major parameters of a host interface are briefly discussed in this Section; more subtle parameters are examined in subsequent Sections.

1. **How fast does it go?** Three speeds are of primary interest:
   (a) The cell rate on the physical medium (a function of the signalling rate and coding and framing efficiency).
   (b) The data rate sustainable over the host bus. This relates to the percentage of cell slots on the physical medium that the host interface can fill or empty.
   (c) The data rate delivered to host processes after processing by the operating system drivers, protocols and sockets.

   Other speed issues include: the maximum rate at which full cell slots may be handled, performance degredation when transmitting and receiving simultaneously and performance degredation when incoming VCI's are highly interleaved.
2. **Which physical layer?** The physical layer may be optical or copper and may use one of several line codes and cell framing mechanisms. Three framing mechanisms are of primary interest:
   (a) Pure ATM (i.e. non SDH) using TAXI block coded data according to the T1.S1 specification as described in the Appendix to this doc ument.

(b) Synchronous Digital Hierarchy (SDH) encapsulated ATM according to the CCITT packing functions and using the SDH scrambler.

(c) The ATM Forum block coded, fibrechannel like encapsulation.

All of these can operate at various line speeds on different physical media with various connectors.

3. **Which host bus?** The interface normally connects into an expansion slot of the workstation. What type of bus and/or workstation is supported?

4. **Which device drivers and operating systems are supported?** A separate device driver is required for each operating system and each work station.

Given these basic parameters, the next distinction between designs is whether the interface maintains any state between handling one cell and the next.

6.2 Interfaces without inter-cell state

If the interface does not maintain state information between cells, then the interface requires processing support from the host CPU for each cell received or transmitted. The functions of such an interface are thus restricted to:

- generation, checking and correction of the per cell header check (HEC),
- generation and checking of payload per cell CRCs, as used in AAL 3 and AAL 4,
- outgoing peak rate control, determined by an interface timer control reg ister
- possibly calculating multi cell CRCs, as used in AAL 5, but with the host being required to load and save the running CRC residue register at the start and end of each cell.

Although host processing is required for each cell received and transmitted, this does not necessarily require a host context switch per cell, since the interface will buffer cells. An example of this approach is the ORL 'Yes V2' interface presented in OSI 95/ORL/Deliverable 2. Received cells are buffered until a cell with the 'push bit' (i.e. the ATM layer user indication in the header) is set. The host then takes an interrupt and is able to process all received cells. Similarly for transmission, the host is able to write cells into the network interface at a higher rate than the actual line interface, allowing the processor to perform

other operations while the interface is transmitting at full (or rate controlled) speed.

Such an interface can operate using either programmed IO only or DMA as follows:

- Programmed IO is used for transferring both ATM payloads and ATM headers. No DMA is used.
- DMA of single complete cells, including a header word (32 bits) and 12 payload words.
- DMA to or from a host memory area formatted into a cell pattern, where every 13th word is an ATM header.
- A hybrid where programmed IO is used for cell headers and DMA is used for the payloads. Payload data is therefore transferred, by DMA, directly to or from the appropriate system buffer in the host.

6.3 Minor improvements to a simple ATM host interface

Minor improvements can be applied to an interface of the type described in the last Section. These add some inter cell state, but without implementing the *sorting* function described in the next Section. However, they might result a several times improvement in performance for many applications These minor improvements include:

- **An outgoing VCI/VPI register.** The interface contains a 32 bit reg ister which the CPU loads with a prototype cell header (4 bytes) to be prepended onto each transmitted payload. Payloads are taken by DMA from consecutive groups of twelve (32 bit) word memory blocks in host memory and transmitted without per cell host processing. The interface sets flag fields in the cell header (such as 'push bit') as required.
- **A reception VCI/VPI register.** The host reads a cell header using programmed IO and uses the VCI/VPI to determine a buffer address in host memory. The VCI/VPI from the header is copied into an interface reception VCI/VPI register and the host writes the buffer address and length into interface registers. The interface then copies data from con secutive received cells held in its receive buffers, by DMA, into the host buffer, but only while the cell headers match the value in the register.

Together, these two improvements enable the host to send and receive blocks of data with low overhead, provided cells of separate blocks are not heavily

interleaved on reception, or are required to be interleaved on transmission (for switch reasons etc.). Support for adaptation can be provided through a logical extension of this method, where separate additional registers are added for each adaptation function which runs from one cell to another within a block. Example functions are sequence numbers, total length, and per block CRC.

6.4 Interfaces which sort

A sorting interface contains state for each active association and is able to receive interleaved cells without host intervention. It sorts received cells into separate queues, one for each VCI/VPI. When an end of block cell arrives, the completed block is available in the queue, ready for host processor attention. Important design parameters of a sorting interface are:

- Which fields within the cell are available as sorting tags? A full implementation would support the VCI, the VPI, and for AAL 3 and AAL 4, the MID (multiplexing id).
- How many different queues (distinct sorting tags) are supported? This determines the number of active incoming virtual circuits for which sorting is supported. Other virtual circuits (e.g. for broadcast VCIs) may not need sorting.
- Are there restrictions on the mapping between the fields which generate the sorting tag? If associative look up of the cell header (and MID field) is used, then there are unlikely to be any restrictions. If directly indexed lookup using a hash function of the VCI/VPI field is used then there will be certain restrictions, but the practical impact of these should have been minimised by the hardware designer.

6.4.1 DMA for a sorting interface

A sorting interface may keep the queues of cells either in its own local RAM or in the RAM of the host CPU.

If cells are reassembled in the interface RAM, then a complete block can be copied in a single DMA transfer to the appropriate host buffer. This results in block sized bus transfers unless they are artificially broken up into smaller bursts by additional hardware to achieve a finer grain of bus sharing. Fine grain sharing is desirable to prevent cache starvation, etc., and is not uncommon in high performance bus interfaces.

Another consequence of reassembling cells in the interface RAM is that dynamic memory allocation is required within the interface, since it is too expensive to supply sufficient local RAM to support a maximal block for each sorting tag.

Alternatively, cells may be copied into host RAM as soon as they are received. This implies that the interface requires only sufficient buffering RAM to absorb variations in bus access latencies. It also now requires descriptor RAM to con tain buffer pointers in host space. The memory management is provided by the host, where it can be integrated with operating system IPC and network IO buffers.

6.4.2 Cells on new circuits

With either of the above approaches, cells either arrive on circuits for which memory has been allocated, or else must be handled otherwise. Cells on new circuits may simply be togged (thrown on ground) until the circuit has been properly set up by management software, or else held while an interrupt is generated. The host may then allocate buffers. It may be necessary to suspend the receive side of the interface while the interrupt is serviced. The new cells can be copied after the handler has re enabled the interface, but other circuits may have been held up while the interface is suspended. Without the suspend option, such cells are lost, but other circuits are uninterrupted. A further possiblity is for the interface to reserve one descriptor for processing all unexpected cells.

Interfaces which copy data into host memory require that consistency is enforced for any host CPU data caches. This has an execution cost for processors which do not bus snoop. If the data is held in uncached interface RAM and copied using processor block moves, then snooping is not required.

6.4.3 Support of buffers not a multiple of 48 bytes

Operating system IO buffers may not be a multiple of 48 bytes long (or 44 as required when using AAL 3 or AAL 4), so if the interface is handling a list of buffer descriptors, it may need to arrange for the first section of a cell to go at the end of one buffer and the remainder to go at the start of the next.

Certain interfaces may restrict cells in host memory to be aligned only on 32 bit word boundaries. This is not likely for interfaces which support AAL 3 or AAL 4, but these might restrict alignment to 16 bit boundaries.

6.5 Choice of adaptation layers supported

An interface may provide full or partial support for a number of standard and non standard adaptation layers. The AAL 5 standard connection oriented data adaptation layer is possibly the most important, but others may also be avail able.

Partial support implies that only certain CPU intensive functions are performed by the interface hardware, whereas full support implies that the host access to the data is fully above the adaptation layer with little or no visability of the underlying cell boundaries.

Implementations of an AAL may vary greatly in their ability to recover cleanly after a lost cell or other AAL fault.

6.6 Transmission scheduling

An interface may be given a chain of buffers which describes more than one block of data to be transmitted on different virtual circuits. It must then decide in which order to send cells. ATM allows cells of separate vitrual circuits to be freely mixed on transmission. This alters the quality of service received by each virtual circuit and the peak rate of a given virtual circuit at the first switch.

A related question is whether outgoing rate control can be applied individually to each virtual circuit. This should be combined with interleaving of outgoing cells where possible.

A full functionality interface should be able to pre empt low priority transmis sion work, already scheduled, with newly arrived higher priority work.

6.7 Other options available in an ATM host interface

ATM host interfaces have to chose how to pack bytes received from the network into a 32 bit word. This gives either a big endian or a little endian interface. Big endian implies that the first byte of a cell is put in bits 24 to 31 of a 32 bit word. The packing order may be an association specific function or a per interface control option. (The ORL OSI 95 ATM interface boards support both endians on a per interface basis.)

Encryption is sometimes combined with a network interface.

Transport layer functions are sometimes combined with an interface. In partic

ular, support for the TP 4, TCP or XTP checksums may be included.

Under certain adaptation layers, interfaces could use AAL information to re order cells which are out of sequence.

6.7.1 Operation and Maintenance

The interface will offer certain loopback options to enable remote testing of the physical link or interface testing without a physical link.

The interface will also terminate certain O & M flows, such as the SDH BIP parity, and the interface may chose to autonomously reply to certain O & M cells (i.e. send a reply cell without consulting the host).

7 Remarks

ATM is a promising technology for both local and wide area networking. ATM must be an integrated part of the protocol suite if its full benefits are to be realised. The MSNA approach for this has been presented. Figure 1 showed a simple mapping between MSNA layers and corresponding OSI layers.

The basic in band operations required for a host interface, inlcuding cell fram ing, buffering, header generation, CRC computation etc. are well defined within ATM. Olivetti Research has constructed a set of ATM interfaces for a vari ety of machines and has run several applications over them. Many of these applications would benefit from a transport protocol layer implemented as an application layer or user space library We look forward to experimenting with such approaches within the ORL Medusa project and the new Esprit project HIPPARCH.

References

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

[ULG 4] 'Specification of TPX' A Danthine. OSI 95 Deliverable ULG 4. Avail able from Prof A Danthine at the University of Liege, Belgium.

[BELL 1] 'ATM LLC Assessment' J Boerjan, R Peschi. Alactel Bell, Antwerpen. OSI 95 Deliverable Bell 1. Available from authors, telephone number +32 3 240

40 11.

[TENNENHOUSE] 'Layered Multiplexing Considered Harmful.' DL Tennen house. In 'Protocols for High Speed Networks' H Rudin and R.Williamson (editors). Elsevier Science Publishers. IFIP WG 6.4 workshop 1989.

[TEN 1] 'Open Systems Interconnection Basic Reference Model.' International Standards Organisation Infomration Processing, International Standard 7498 1.

[TEN 2] 'OSI Reference Model Part 3: Naming and Addressing.' International Standards Organisation Infomration Processing, International Standard 7498 3.