

ATM as a memory interconnect in a Desk Area Network

P. Gunningberg* and Ø. Kure**

*Uppsala University/SICS, Sweden, per.gunningberg@docs.uu.se

**Telenor Research, Norway, oivind.kure@tf.telenor.no

Abstract

ATM has been successfully used in Wide Area Networks (WAN) and Local Area Networks (LAN). A possible next step in this evolution is to use ATM for memory interconnect. This paper discusses how ATM can fulfill the functional and performance requirements in a memory interconnect, and the implications this will have for external ATM communication. We conclude that ATM can be used as a memory interconnect. It can meet the throughput requirements but will have problems with the latency requirements for the transfer of small cache lines. The fixed 48 byte payload in ATM results in inefficient use of bandwidth and increased latency. The ATM and the ATM Adaptation Layers do not have sufficient functionality and must be extended to meet memory interconnect requirements. Using ATM internally in the memory interconnect has a limited synergy effect when the system is connected to an ATM based LAN or WAN.

Keywords

Data Communication, ATM, Desk Area Network, memory, multiprocessors.

1 INTRODUCTION

The term Desk Area Network (DAN) has been introduced for networks that interconnect devices typically found on a desktop, such as displays, cameras, speakers and microphones (Hayter,1991)(Katevenis,1994)(Finn,1991)(Leslie,1991)(Adam,1993). ATM was originally developed to carry integrated communication streams in a Wide Area Network (WAN). It has also been shown as a viable Local Area technology. The primary advantages of ATM are high bandwidth, ability to carry integrated communication streams, potential for re-

source reservation, ability to scale, and independence of a given physical layer. ATM is therefore a prime candidate for a DAN.

There is an interest in utilizing switching technology, such as ATM for memory and multi-processor interconnect (Adam,1993)(Hayter,1991), since the traditional memory buses do not scale (Dutton,1992). For memory interconnect ATM has the following potential advantages: 1) cost effectiveness since ATM chips and modules are proliferating, and 2) potential synergy between external ATM based communication and the interconnect.

At least two demonstrators have shown the feasibility of using a modified ATM in a DAN, the Vunet which is a part of the Viewnet distributed multimedia project at MIT and the DAN project at Cambridge University(Adam,1993)(Hayter,1993). These projects emphasize the interconnection of multimedia devices, and the asynchronous transmission of fixed length cells.

Vunet is a network architecture for handling multimedia. It uses ATM to interconnect three types of devices; workstations, image processing systems and network interfaces. A modified ATM standard with 56 byte cells, 700 Mb/sec links, and a pseudo AAL5 protocol is used.

The DAN project at Cambridge University utilizes a cell based interconnect, the Fairisle switch, for a multimedia workstation (Leslie, 1991)(Hayter, 1993)(Leslie, 1993). Specialized port controllers are used to interconnect a camera, a frame store, and a Digital Signalling Processing (DSP) node. The interconnect is assumed to be reliable with a feedback channel in case of unsuccessful delivery of cells. By using cells with 53 bytes they intend to move external ATM cells over the network interface without data transformation. In (Hayter, 1993) it is shown how a CPU/memory could be interconnected with 54 bytes cells, a specialized adaptation layer, and a specialized cache design.

Both demonstrators use limited ATM functionality and modified ATM standards. Our intention is to go one step further and discuss the advantages and disadvantages of using standard ATM as a memory interconnect. A potential advantage is the synergy between the memory interconnect and external ATM based LAN/WAN. The analysis of ATM for memory interconnect is therefore based on ATM as used in LAN/WAN

We identify the requirements of a memory interconnect and discuss how ATM can meet them. Our contribution is the evaluation of ATM as a memory interconnect and a discussion on alternative architectures to attach an external WAN ATM to an ATM based memory interconnect.

We believe that ATM can be used as a memory interconnect, but it has no inherent advantages besides the possible cost effectiveness; the ATM technology must be modified or streamlined for carrying memory data units to such an extent that it provides few advantages when such a system is connected to a WAN ATM.

The next chapter describes how ATM could be used as a memory interconnect. The third chapter discusses the external communication for an ATM based interconnect. The last chapter summarizes our results.

2 ATM AS A MEMORY INTERCONNECT

A typical high performance workstation has several buses, a memory bus reserved for fast memory access and an I/O bus for network interfaces, disks and other slower interfaces. A memory bus may also implement a cache coherence protocol for multiple caches. In this sec-

tion we discuss whether ATM can replace both types of buses. We identify functional and performance requirements and discuss how ATM can meet them.

2.1 Architecture

The reference configuration used in our discussion is displayed in Figure 1. It consists of a switch, two or more CPUs, different memory units (such as cache and primary memory), I/O units, one or more network interfaces, and several multimedia components including camera and display.

We assume the switch itself will not be the performance bottleneck, since it will only interconnect a limited number of devices, each with a bandwidth requirement of at most a few Gb/sec. This is achievable for the next generation switches and even for some of the experimental ones existing today (Le Boudec, 1993). Instead we will assume the bottlenecks will be the destination devices and their links.

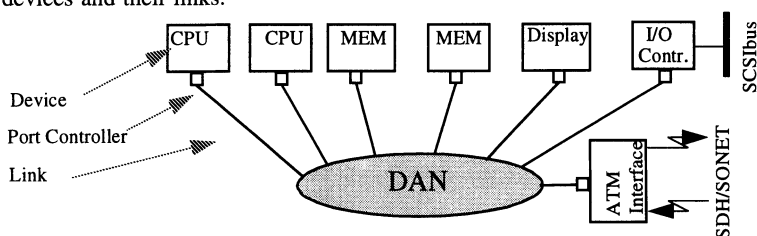


Figure 1 Reference architecture for a DAN.

An ATM based interconnect will have a small geographical distribution which implies a small bandwidth-delay product. At a Gb/s link rate, an ATM cell will stretch 80 meters. This is well within the distance from a source to the switch and implies that there is only one cell at a time on a physical link.

2.2 Functional requirements for memory interconnect

A traditional backplane bus differs from a network interconnect in how the medium is accessed and utilized. A traditional bus typically has four phases for read/write transactions, and during a transaction the requesting module, e.g. the CPU, is blocked.

The four phase architecture is difficult to extend to high capacity interconnect for multiprocessors; the capacity of the bus does not scale with the length of the bus and the number of connected devices (Dutton, 1992). In response to this, so called split transaction types of buses, like IEEE SCI (IEEE, 1992) and Xerox/Sun Microsystems XBus, have started to appear. In these buses, the CPU and the buses are not blocked while waiting for the response from an addressed device. Instead, requests and responses are given identification labels and are sent asynchronously on the bus. We expect that this transaction type of memory communication will be used for a DAN memory interconnect.

The set of transactions vary from interconnect to interconnect. There are at least some variants of a read and a write such as:

- `write(address, identification, return_address, size, data)`
- `write_confirmed(address, identification)`
- `read(address, size, identification, return_address)`
- `read_response(address, size, identification, data)`

Our aim with these transaction examples is to describe the functions and parameters required in order to compare them with the services offered by ATM. A write transaction has a return address parameter so that the memory can send back a confirmation whether the operation was completed or not. The identification parameter is used to match a request with a response. The parameters *size* and *data* are pointers to the data to be sent and the size of it. The read transactions are symmetric to the write transactions.

The outlined transactions cannot be mapped directly onto the ATM services. Instead they must be offered through extending an existing AAL service or by creating a new one. In the next sections we will discuss how ATM can meet required services with respect to data size, request/response types, addressing, identification and control.

Data size

The ATM cell payload of 48 bytes of data is too small for most fundamental data unit sizes of a memory interconnect, see table 1. With the fundamental data size we mean the smallest number of bytes handled as a unit in the different devices, such as the line size of a cache. The standard ATM adaptation layer for data communication, AAL 5 has a maximum size of 64kbytes which is large enough for most sizes. The length field of AAL 5 can therefore be used directly to indicate the size of the current data unit.

Addressing

ATM has a different addressing model compared to transaction buses. ATM uses a signalling protocol or management functions to set up a communication channel between a source and one or more destinations. After a channel is established, the cells belonging to this channel are given a 28 bit (VCI/VPI) identifier with local significance.

In contrast, memory transactions are connectionless. The complete state is contained within the transaction, including the absolute memory address. Such a transaction can be directly emulated in ATM by setting up a channel for each transaction, sending the request or response and closing it afterwards.

An alternative is to utilize part of the 28 bit VCI/VPI as a relative memory address. In a DAN with only a few devices, a few bits of the VCI/VPI space is sufficient for channel identification. The remaining bits in the label could be used for relative addressing within the device. It implies a limitation on the available address space, and requires a modification to the switching algorithm in the switch. As an example, using 5 bits (32 channels) for ATM switching would leave 23 bits (8 M) for addressing within a device.

Of the two previous alternatives, the signalling approach will be too slow to be attractive. Using parts of the VCI/VPI field imposes a restriction on physical address space and it is unclear how much additional processing it requires.

The simplest and hence the most attractive we believe, is to include the addresses in the payload even if they increase the latency. It requires setting up two permanent channels, one in each direction for each pair of communicating devices. The memory addresses are carried in the adaptation layer payload. This implies processing for extracting and inserting the addresses from the payload, which adds to the latency and reduces the effective payload. At the

server side, the return VCI/VPI for each request must also be retained in order to identify the return channel for the response.

Identification

In a transaction memory system, responses may come back in a different order than the corresponding requests were sent. To be able to match a response with a request, both are given identification labels, such as a sequence number. The maximum size of the sequence number is decided by the number of allowed outstanding requests. We expect this to be a relatively small number due to limited distances in a DAN and short processing latencies.

Neither ATM or AAL5 provide any sequence number or optional field that could be used directly. However, an ATM channel preserves the order of requests and responses in the channel which could be used for some *weak* ordering constraints. If a device does not reorder requests from the same channel and does not lose any of them, then the order on the response channel will be the same as on the request channel. If an identification label is used, a reordering at the device as well as the network can be allowed. Such an identification label must be put into the payload and consequently it will reduce the useful load and increase processing latency. If a *total* ordering is required by a client (e.g. CPU) between concurrent channels, then all these requests must be given a total ordering at the requesting client. A *global* ordering, i.e. request from different devices are ordered according to global time, is out of scope for this low level protocol. We believe that total ordering is necessary since the programmer may not know the memory hierarchy model.

Commands

A command field, used to identify the type of requests and responses, must also be a part of a payload field. There is a 3 bit payload type field in the ATM header that may be considered. It is now used for ATM management, user signalling (AAL5 EOF) and congestion control. To use this field to carry memory access commands would be incompatible with WAN interoperability. The memory access commands must therefore be carried as part of the payload in an AAL frame. As for the other fields in the payload, this will reduce the effective transfer rate as well as increase the latency.

Feedback mechanisms

A memory interconnect needs mechanisms for access control to devices, for ensuring correct transfer, and for regulating the data flow to/from the devices. They are most efficiently implemented by feedback mechanisms. In a traditional bus, handshake schemes between source and destination ensure synchronized and ordered transfers. In a split transaction technology, like Scalable Coherent Interface (SCI), there is a more elaborate feedback "echo packet" used to provide error and flow control (IEEE,1992).

ATM lacks a general feedback mechanism. Instead it must be implemented as part of an adaptation layer running on top of two separate ATM channels, one in the forward and one in the reverse direction. This has implications on system performance, since the packet size on the return channel cannot be smaller than 48 bytes. The actual feedback information is typically a few bytes and padding the of rest of the cell wastes bandwidth and adds latency. For high rates of packets on the forward channel this can lead to substantial bandwidth requirements on the reverse channel. As an example, if feedback is required for every transfer of a 64 byte cache line (needs two cells), the reverse channel will use half the bandwidth requirement of the forward channel.

ATM mechanisms for traffic control used in the memory interconnect

A WAN ATM network must implement resource reservation, monitoring and enforcement of resource usage, transmission rate, and congestion control. The primary resources are link bandwidth and buffer space in switches. At the setup phase of a connection, Call Admission Control is used to reserve resources or to deny a connection setup if it cannot be supported with the available resources. The policing mechanism ensures that the traffic on a connection does not exceed its resource reservation and interfere with other connections.

A memory interconnect will have a simple topology with few devices, and we do not anticipate a large degree of multiplexing of bursts onto the same destination. To reserve transmission capacity on the links and in the switch for channels between all possible combination of source and destination will result in poor link utilization and longer delays. A better alternative is to give all sources full access to the transmission capacity with the inherent risk of congestion and cell loss. Although a switch has the ability to absorb a limited period of congestion, this alternative must be supplemented by congestion control to throttle sources with longer bursts. With the short distances in a DAN, we believe that congestion control with explicit feedback will be responsive enough. Hence, in a memory interconnect, resource reservation and policing are of limited value.

The ATMforum is in the process of standardizing an available bit rate service (ABR) that incorporates a feedback congestion control. Potentially, such a mechanism could be utilized in a memory interconnect. Regardless, such a scheme assumes the ability to shape the traffic at the source. Shaping implies that the port controller must be able to delay cells until the "right" transmission epoch occurs, which adds both complexity and latency.

Table 1 Performance requirements

<i>Entity</i>	<i>Size in bytes</i>	<i>Latency in microsec.</i>	<i>Comments</i>	<i>Throughput in MB/s</i>	<i>Classification</i>
Cache line - 2:nd level cache to main memory.	64	0.375	On Sun's XDBus	250	Memory to memory
Cache line or a page moved through a 2-level memory hierarchy.	256	2.250	Estimated from number of processor cycles	250	Memory to memory
A block to/from disk.	8k	20000	SCSI-2	10	I/O to memory
High Resolution display (24*1280*1024)bit	4M	30000	Low latency is desirable	15	I/O to memory
WAN ATM interface	53	-	48 bytes payload	19	I/O bus

The ATM mechanisms are at "link level". From an end-to-end type of argument it can be argued that reservation, rate control, policing and possible congestion notification mechanisms at the ATM level can not replace end-to-end flow control (Saltzer 1984). These mechanisms are in terms of cells, while the end-to-end flow is in terms of data units. Congestion in end device therefore has to be handled by end-to-end flow control.

2.3 Memory interconnect performance requirements

In this section we will identify representative devices found on traditional buses, their type of communication, their fundamental data unit sizes, and discuss if their performance requirements can be met by ATM. Table 1 summarizes their data unit sizes, latency and throughput

requirements. Note that the performance figures and sizes vary widely for each type of device and should not be taken as exact figures. The purpose is to discuss whether an ATM based system is feasible and we are therefore more interested in the order of magnitude rather than the precision of the figures.

We sort the devices into classes according to the requirements on latency, size of data unit and type of communication.

- *Memory-to-memory.* For the memory-to-memory class we count all traffic between cache, main memory and from a local memory to another local memory in a multiprocessor, such as NUMA and COMA processors. A cache line is the normal data unit of the memory bus. The size of a line varies, from 8 bytes for first level, SRAM based caches, to 512 bytes for second level caches and even bigger for multiprocessor machines. The communication is characterized by a transaction type of communication with a request to read or write a cache line followed by a response. Low latency as well as high throughput are crucial for this class of traffic since one or several CPUs may be blocked during a transaction. In Table 1 we have chosen Sun Microsystems and Xerox PARCs XDBus to represent second level cache to main memory traffic (Gwennap, 1993). The calculated latency time for the XDBus include DRAM access to a non-interleaved memory. For the local-to-local memory latency estimation we have used calculations in (Joe,1994) on number of processor cycles needed to access a cache line over a two level bus hierarchy.
- *Memory-to-I/O.* Representative for this class are DMA transfers to or from primary memory. We do not expect CPUs to be blocked while waiting for a data unit as is the case for the CPU in the memory-to-memory class. Hence, there are less stringent requirements on latency. In Table 1, this class is represented by a SCSI II device and the transfer of high resolution bitmaps from memory to a screen.
- *I/O-bus interconnect.* The I/O devices in this class are relatively insensitive to latency and most of them do not demand high throughput either. The latency is likely to be dominated by the application latency, operating system CPU scheduling or by external transmission times, e.g. for a WAN connection. In Table 1 the class is represented by a 155 Mb/s WAN ATM. We also include non transaction type of transfers to this class, such as a stream of digitized samples of video units, where the sender do not expect a confirmation for each unit.

Latency and data unit size

At a first glance, it looks like ATM will meet the throughput requirements but will have problems with latency. A 2.4 Gb/s (300 MB/s) ATM link seems to meet all throughput requirements for a memory interconnect. Higher throughput could be achieved by multiple ATM links to a device.

As concluded before, a straightforward solution for ATM as a memory interconnect is to create control fields in an AAL 5 payload for addresses, commands, and identification. These new fields create an overhead cost which is significant for small data units. For example, assume four byte addresses, two bytes for identification and two bytes for the command field. To this AAL 5 adds a 8 bytes trailer. Altogether this overhead adds up to 20 bytes

ATM transfers only complete cells. Any data unit that does not fill up a cell is padded which wastes bandwidth and adds latency. For large AAL frame sizes this is hardly noticeable, but for cache lines the overhead can be substantial. For example, the transfer of a 32 byte cache line over AAL5 needs two cells which only yields the fraction 32/106 of the offered rate. This clearly motivates the use of a specialized memory adaptation layer that can transfer a 32 byte cacheline in a cell. For a cacheline of 64 bytes the required ATM rate will be 413

MB/s. Or to put it another way, a cell will arrive every 120 nanoseconds to a memory device. Within this time a device must do all ATM and Adaptation layer processing and for big application data units also the reassembling or fragmentation.

Doing the same calculation for a 256 byte cache line yields a required ATM rate of 281 MB/s. This considerably lower rate is due to a longer data unit size relative to the header and completely filled cells except the last one. We believe it is likely that links, switches and devices in the future will match this speed.

The memory-to-memory latency over an ATM connection consists of several parts; first there is a processing part at the sender, then a transmission time, a propagation time for the cell and a switching time. From the switch to the receiver, the parts are in opposite order, see

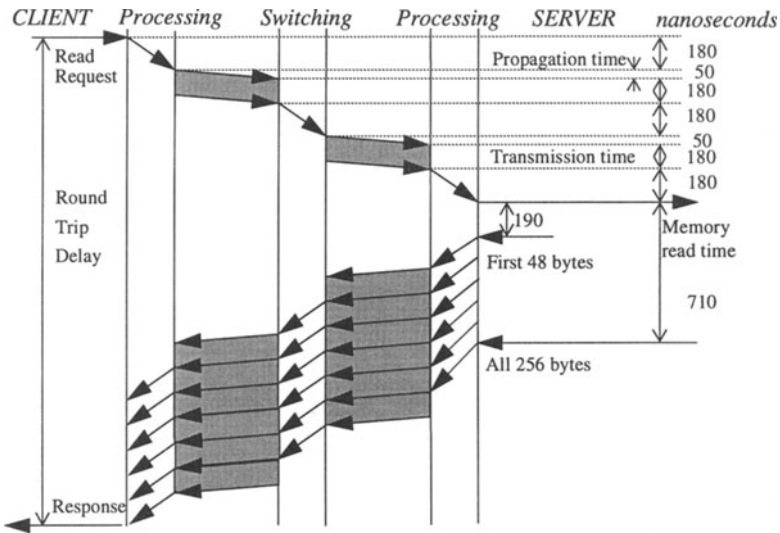


Figure 2 Latency distribution for a memory read request.

Figure 2. The propagation time is decided by the speed of light in the fiber, which is about $2.1 \cdot 10^8$ m/s. For a 10 meter fiber to a switch the propagation time will be about 50 nanoseconds. The transmission time is given by the rate of the network. Assume a transmission rate of 300 MB/s. For a 53 bytes cell it will then take about 180 nanoseconds for the transmission. A read memory transaction consists of a request cell followed by one or more response cells in the other direction. For a two cell response the second cell will arrive directly after the first one, which adds a transmission time of 180 ns and additional memory read time. As illustrated in the figure, a cell must be completely received in a switch before it can be forwarded to the destination, unless a cut through technology is used. The switching delay is at most 180 nanoseconds since we assume that the switch is not the bottleneck.

Assume a random access time of 70 nanoseconds (ns) to a CAS DRAM and 10 ns for each additional 4 bytes accessed (Przybylski,1993), then a 48 bytes payload has a total access time of 190 ns. Similarly, for a cache line of 256 bytes the time will be 710 ns.

Altogether, a conservative estimate of the round-trip latency for one cell sums up to 2190 ns including the memory access time. However, it is likely that ATM processing can be done partly in parallel with the transmission. At most this cuts 720 ns from the total time in our example. Comparing the round-trip time to the latency estimations of the XDBus in table 1 indicates that an ATM DAN can not match the latency requirements of a second level cache, even if the network transmission rate is increased.

The latency requirements for local-to-local memory seems to be within reach. Larger data units are used and they can be sent and processed in a pipeline fashion which reduces latency. With a cache line size of 256 bytes there will be six response cells sent back-to-back, each with a transmission time of 180 ns (see Figure 2) and the whole transaction will take about 3090 ns given the same assumptions as above. This time does not include any address directory accesses or cache consistency maintenance. We conclude that ATM DAN is within reach for this type of memory interconnect.

3 WAN INTERCONNECT

In this section we discuss the potential for synergy between an ATM based memory interconnect and external ATM based communication. By using the same technology “internally” and “externally” the two can be closely integrated and an explicit network interface can be avoided. The latency and processing overhead associated with protocol conversion, buffering and alignment in an interface can be reduced. In addition, the functionality associated with the external ATM connection can be extended end-to-end, from external memory to local memory. However, these synergy effects can only be achieved through more complex devices. How a WAN connection is integrated into the memory interconnect is the main design issue determining the cost and benefit of the concept.

There is a spectrum of feasible architectures for a WAN interconnect, ranging from a “traditional” architecture with a separate interface to an “open” architecture where each device is directly accessible from the WAN. In the open architecture each device can terminate WAN connections. In this architecture, the WAN functionality must therefore be replicated at all devices. In the traditional architecture a common interface terminates all WAN connections on behalf of all other devices. Conceptually, the interface must perform all ATM and AAL functions. In an actual implementation though, these functions may be distributed over several processors in the memory interconnect. The placement of functions is independent of the interconnect technology and therefore orthogonal to our discussion. Between these extremes there are architectures where some devices are visible to the outside, by replicating all or a subset of the needed WAN functions, while the remainder continues to be served by a centralized interface. In the next paragraphs we will discuss three architectures in more detail.

3.1 The open architecture

In the open architecture, the WAN link attaches directly to the memory interconnect switch which makes each device directly connected to the WAN (see Figure 3(a)). It is cost effective since a separate network interface is not needed. The architecture provides lower latency for external communication by avoiding conversions from “internal” to “external” protocols, such as buffering and alignment. However, this can only be realized at the cost of a more complex device software and hardware, since each device must implement the WAN adaptation layers

and perform signalling and congestion control, as well as shaping in order to terminate external ATM connections. None of these services are essential for the ATM memory interconnect.

The fact that all devices are reachable directly from the outside is not only an advantage. It has consequences for security, object location and relocation. When an externally reachable object is relocated within the DAN, the new location must be propagated to external name or location servers. With the open architecture, the devices can no longer rely on communication with trusted devices, and each device must therefore guarantee the confidentiality and the integrity of their communication.

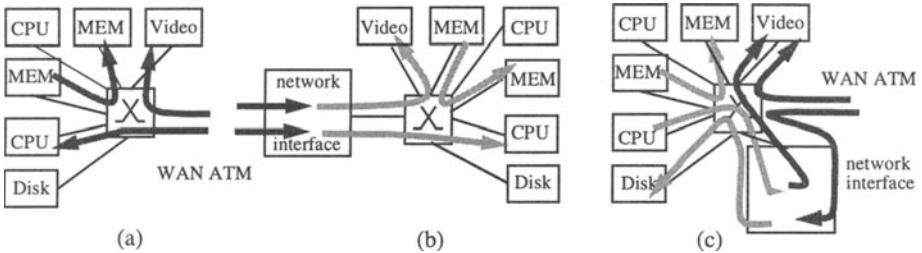


Figure 3 (a) Open architecture, (b) Traditional architecture and (c) Hybrid architecture.

3.2 The traditional architecture

In the traditional architecture, the interface acts as a gateway between the WAN protocol stacks and the stack for the memory interconnect. The network interface transfers the payload of an adaptation layer frame over the internal network to the destination (see Figure 3(b)). All WAN specific functions such as shaping, flow and congestion control, and signalling processing can be located at a common network device instead of being replicated. On the other hand, the latency for external connections increases since the network interface has to do conversions from internal to external protocols, as well as buffering and alignment of the payload.

In this architecture, the memory interconnect is in principle completely hidden from the external ATM. As discussed before, we believe that ATM in the memory interconnect should be "different" from ATM used in external communication. With different protocol stacks used on the "inside" and "outside" of the WAN interface, the synergy of using ATM on both sides is not apparent.

A possible synergy can be achieved through tunnelling the cell payload through the interface without any realignment. In the previous section we concluded that additional control fields are needed to carry commands, addressing and identification labels. To avoid realigning the cell payload due to different external and internal frame format, the additional control fields must be carried in pre- and postamble cells. However, any technology that can transfer bursts longer than 48 bytes plus the length of the control fields can achieve similar tunnelling effects.

3.3 The hybrid architecture

In the traditional architecture it is assumed that the memory interconnect only supports the specialized memory adaptation layer. Devices with the ability to terminate WAN ATM connection, like camera and displays with standard ATM interfaces, cannot be used since they

most likely do not support the specialized memory adaptation layer. A solution, that combines the advantages of the open and the traditional architecture, is to attach the WAN connection directly to the internal switch as in the open architecture in combination with a specialized interface device that handles the conversion/augmentation between the "internal" ATM and the "external" ATM (see Figure 3(c)). The interface device will have the same functionality as the network interface in the traditional architecture, and it may implement all the needed WAN functions. By attaching the WAN connection directly to the memory switch, ATM streams can be routed directly to devices that can terminate WAN connections. The architecture is also cost effective since off-the-shelf equipment with ATM interfaces can be connected directly to the switch, and since the remaining devices only need to implement the memory adaptation layer. Traffic from the directly attached devices to "internal" devices will have to be routed through the interface in order to handle the protocol conversion between "internal" and "external" ATM. A consequence is that this traffic processed by the interface must pass through the switch twice. However, the switch is not assumed to be a bottleneck, so the added switching should not be a burden.

4 CONCLUSIONS

We have discussed the advantages and disadvantages of ATM as a memory interconnect architecture and how to integrate external WAN ATM connections. Our overall conclusion is that ATM can be used in such a role, but it is neither optimal from a functional nor a performance point of view. ATM can meet all throughput requirements but it is only acceptable for memory to memory traffic that can tolerate latencies longer than a microsecond. Furthermore, ATM must be enhanced with memory transaction type services, either in the form of a new adaptation layer or as an addition to an existing one. Such a layer should provide memory addressing, identification of transactions and a feedback mechanism. The ATM signalling, shaping and resource reservation functions are not required, and should be removed in order to enhance performance.

The fixed length of ATM cells is a disadvantage in a memory interconnect. A mismatch in the size of the data unit and the 48 byte payload of ATM requires padding of the last cell. As a consequence, the interconnect must be over-engineered to meet the bandwidth requirements. Furthermore, this padding translates directly to an increase in latency.

ATM provides little synergy effect when the DAN is connected to a WAN/LAN, since the memory interconnect requires extensive streamlining of ATM. Instead a specialized network interface for external ATM connections is preferable.

The cost part of the price/performance equation has not been analyzed, so it is conceivable that ATM could provide a cheaper solution due to the proliferation of ATM chip sets. A counter argument is that there may not be cost effective chip sets for the high bandwidth required for the interconnect, since it is substantially higher than the bandwidth target for the 'mass market.

The multicast function of ATM is one of the functions we have not discussed. We speculated that it may be used in a memory coherence scheme. With multicast, cells transmitted on the source channel are copied to a configurable number of destination channels. The advantage is the scale factor; the transfer will only consume bandwidth on the links for those destinations that are required to receive the transaction.

5 REFERENCES

- Adam, J.F., Houh H., D. Tennenhouse, D. (1993) Experience with the VuNet: A Network Architecture for a Distributed Multimedia System, *Proceedings of 18th Conference on Local Computer Networks*, Minneapolis, Minnesota, September 19-22, 1993, pp 70-76
- Dutton, T., Eiref, D., Kurth, H., Reisert, J. and Stewart, R.L. (1992) The Design of the DEC 3000 AXP Systems, Two High-performance Workstations, *Digital Technical Journal*, No 4, 1992
- Finn, G.G. (1991) An Integration of Network Communication with Workstation Architecture, *ACM Computer Communication Review*. Vol21 no 5, pp18-29
- Gwennap, L. (1993) Sun, Xerox to license XDBus Technology, *Microprocessor Report*, March 8, 1993, pp 1-10
- Hayter, M. and McAuley, D. (1991) The Desk Area Network, *ACM Operating Systems Review*, no 25, October 1991, pp 14-21.
- Hayter, M. (1993) *A Workstation Architecture to Support Multimedia*, Ph. D Thesis St. John's College, University of Cambridge, September 1993
- IEEE (1992) *Scalable Coherent Interface*. IEEE Standard 1596.
- Joe, T. and Hennessy, J.L. (1994) Evaluating the Memory overhead required for COMA architectures, *IEEE 21st International Symposium on Computer Architecture*, pp 82-90.
- Katevenis, M. (1994) Telegraphos: High Speed Communication Architecture for Parallel and Distributed Computer Systems, *FORTH-ICS/TR-123*, May 1994.
- Le Boudec, I.J., Port, E. and Linh Truong, H. (1993) Flight of the Falcon, *IEEE Communication Magazine*, February 1993, pp 50-56
- Leslie, I., McAuley, D., and Mullender, S. (1993) Pegasus-Operating System Support for Distributed Multimedia Systems, *ACM Operating System Review*, No 1 Jan. 1993, pp 69-79
- Leslie, I. and McAuley, D. (1991) Fairisle: An ATM Network for the Local Area, *Proceeding ACM SIGCOMM 1991*, pp 327-336.
- Przybylski, S. (1993) DRAMs For New Memory Systems (Part3), *Microprocessor Report*, March 29, 1993, pp 22-25
- Saltzer, J., Reed, D. and Clark, D. (1984) End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2, pp 277-288.

6 BIOGRAPHIES

Dr. **Per Gunningberg** is an associate Professor at Uppsala University and a part-time researcher at the Swedish Institute of Computer Science, SICS. He joined SICS research staff in 1985 and prior to SICS he spent a year and half as a visiting assistant professor at the University of California, Los Angeles. His interests include protocol implementations, real time systems, distributed operating systems and dependable computing. Authors address is: Department of Computer Systems, Uppsala University, Box 325, S-751 05 Uppsala, Sweden.

Dr. **Oivind Kure** is a senior researcher at Telenor Research. He joined Telenor Research in 1988 after having receiving his Ph.D from University of California, Berkeley. His research interests include data communication, performance analysis, and distributed operating systems. Authors address is: Telenor AS, P.O. Box 83, N-2007 Kjeller, Norway.