Micronet Machines - New Architectural Approaches for Multimedia End-systems

Tom Pfeifer

Technical University of Berlin - Dept. of Computer Science
Franklinstr. 28/29; 10587 Berlin; Germany
email: pfeifer@fokus.gmd.de

Abstract: The new quality in the relation between networks and end-systems requires a harmonization between these worlds. Based on the requirements and possibilities of new applications, particularly in the field of multimedia, the architecture of personal end-system has to be reconsidered completely.

In this proposal the network is not only connected to the station, the station's internal communication system is a cell switching network itself, as close to ATM technology as possible. Topologies are discussed, and advantages of buses vs. networks are compared. Consequences for processing elements, the switching fabric and peripheral devices, as file systems, are discussed.

1 Introduction

Upcoming 'Gigabit Networks' change the relation between communication systems and workstations dramatically: the former hand their role as bottlenecks to the latter. Completely new approaches for Global Distributed Processing with shared network-memories have already been proposed [2], see Figure 1 for an example.

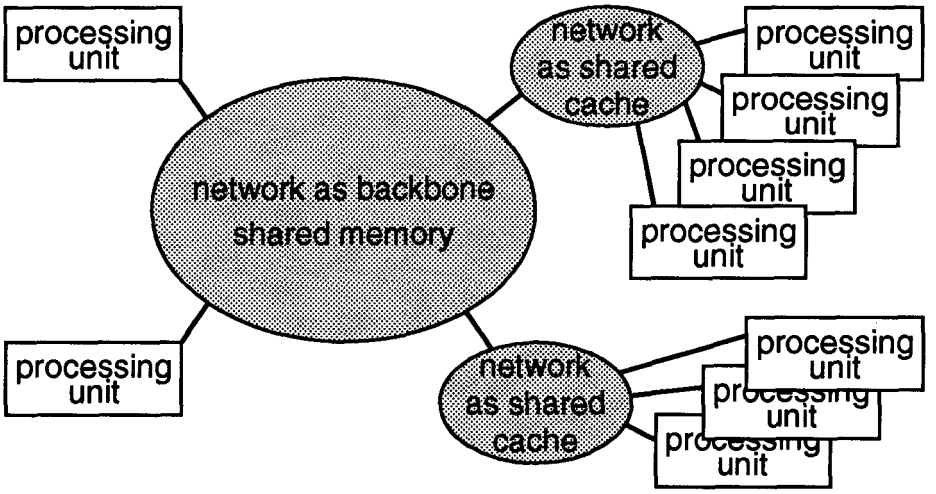

FIGURE 1 The interconnected shared memories architecture [2]

In such an environment, the role and therefore the architecture of the end systems has to be reconsidered completely. Network access (in the context of shared network-memory) will not be any more an I/O-process among others, it will become the central component of all computing activities.

The proposed end system architecture is a step towards a harmonization of the two worlds of data processing and telecommunication.

2 Bus Systems - the well known bottleneck

With increasing performance of processors and peripheral systems, a well known bottleneck of the classical von Neumann architecture becomes obvious: the bus system (Figure 2).

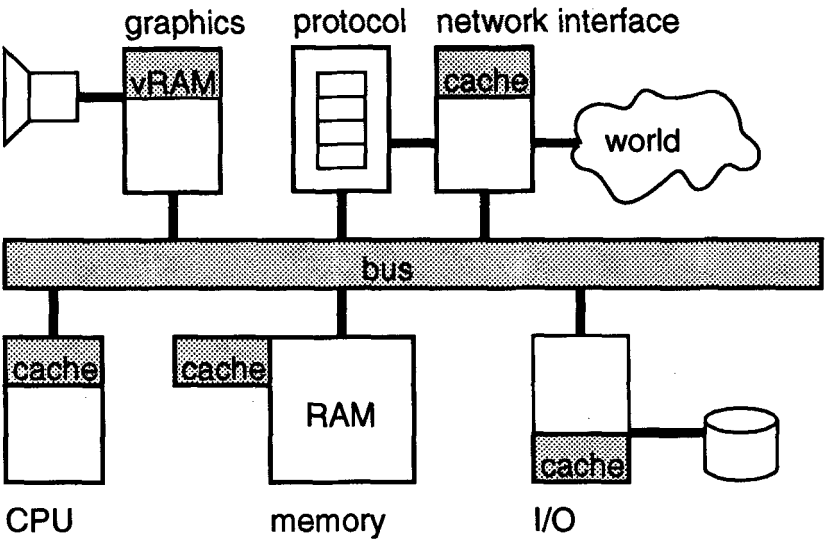

FIGURE 2 Classical bus system

The bus is a multiplexed and therefore shared resource for all other components of a workstation and reaches its limits particularly when multiple continuous media are involved. Current approaches to increase the bus performance in the stations lead to bizarre constructions of separate local buses [10], special video buses, etc. (Figure 3).

While in the first place the bus has provided a high degree of flexibility and modularity, such stations are now losing their flexibility as a universal computing system because the dedicated communication paths do not allow a flexible, open architecture.

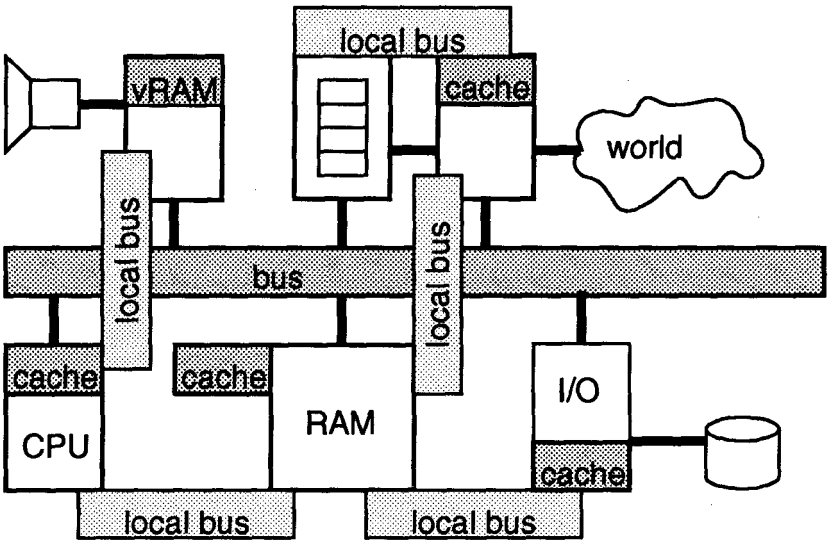

FIGURE 3 Bus-focused architecture with add-on local buses

If, for example, the video codec delivers its decoded high-volume data stream directly to the graphics output only, there is no possibility to edit the decoded stream, and the station remains dedicated for playback purposes.

Another problem of a bus shared by several devices is the low scalability of the system [1], because the fixed capacity of a bus limits the number of attached devices and processing elements.

Multimedia workstations of the future need a completely different architectural approach. They have to be scalable and configurable for the needs of different classes of applications.

Covaci and Zeletin [2] have also identified several bottlenecks in current workstation and network design. The bus system is among them, and another one is the general lack of compatibility between data in external networks and in the station's own internal communication system. Neither the access mechanisms nor the handled data units are related to each other in any way, and the network adaptors have to provide a full gateway functionality.

The developments in multimedia technology merge the two worlds of data processing and of telecommunication. This process needs to be supported by a harmonization between the internal and external communication architecture of multimedia computer systems.

3 Network techniques inside the end-system?

Communication systems inside computers which are different from buses have already been discussed in literature about computer architecture. There are a lot of known approaches with network structures, other with hyper cubes and crossbars [11]. As far as I know, such approaches have not been used for workstations and personal computers for a larger market, probably because such an architecture would be too expensive to implement.

On the other hand, research in parallel computing focuses on sharing processing tasks between arrays of equal processors, or between similar workstations. While this approach has a lot of applications, integrated multimedia-systems need another one. They need specialized elements for certain tasks like video compression, or at least for more universal subtasks like DCT transform, image block comparison, Forward Error Correction (FEC), etc. Such specialized elements will be part of mass market workstations in future, as the built-in clock/calendar is of today's.

The current development of high speed network technologies gives reason to rethink the approaches of workstations based on loosely coupled processing elements, communicating over network structures. Cell switching ATM network technology is gaining popularity. A lot of research is being done to connect workstations directly to ATM networks. Everybody agrees that connecting ATM to workstations requires adaptors with specialized hardware [3, 4, 5] for protocol processing (CRCs, VCI/VPI control), for transforming the data and accessing the station's memory by DMA. One good proposal tries to harmonize the cell traffic with characteristics of the bus (burst mode on the bus for cell transmission) [3].

In this paper I want to go beyond this point and discuss a new architectural concept not only for connecting the station to the network but also including cell switching network technology in a multimedia workstation as its internal communication system. Connection oriented services and the possible allocation of resources would make the quality or performance of application processes more predictable.

A major point is to discuss whether it would be useful to have the full ATM functionality inside the station. In this case, every device inside the station would need all ATM/AAL protocol capabilities. This would make the devices very complex. But, in order to make the communication with the outer world as easy as possible, all mechanisms implemented inside the station should be as compatible as possible to ATM. So, in any case, the fixed cell length of 53 bytes should be kept equal everywhere.

The question arises as to what changes and simplifications can be made to the ATM specification for this internal usage.

First, the inside traffic could be considered as reliable, as buses are today. Therefore, a lot of protocol handling of the different AALs could be left to the network adaptor.

Second, the header functionality of ATM cells could be further reduced for internal purposes. This step can make the header recognition easier.

The idea of header controlled virtual channels and paths (VCI/VPI), as it is used in ATM, fits perfectly into the idea of internal networks. Within the locality and limitations of a workstation, it would also be possible that channels/paths between certain devices are predefined (e.g. a predefined path number between hard disk and graphics output). The network adaptor would translate these internal identifiers for the outer world when required.

Finally, it could be assumed that the cells are not misordered when they travel internally. This makes internal routing and reassembling much easier. Handling of out-of-sequence cells from outside should also be left to the network adaptor.

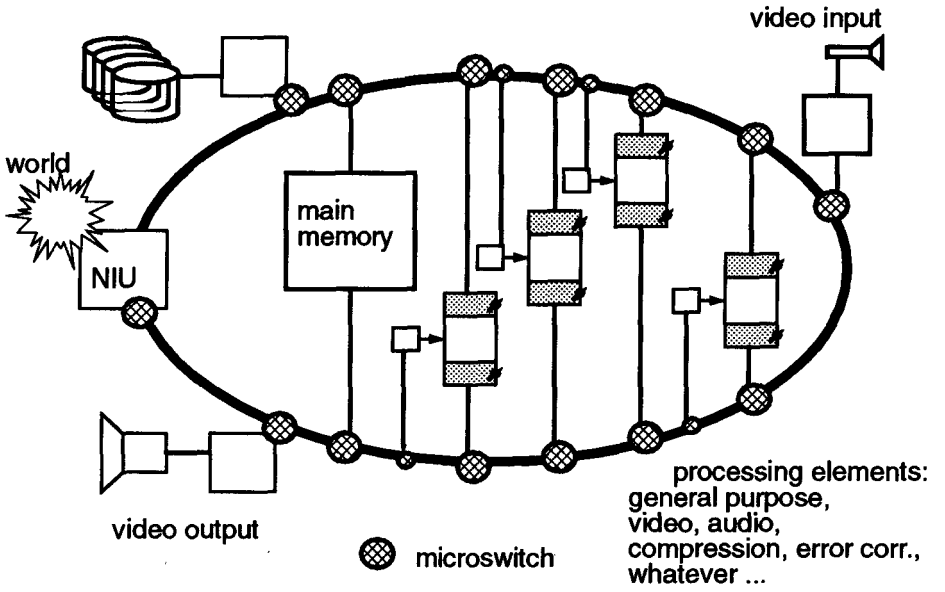

FIGURE 4 The Micronet Machine

The essential question for internal cell switching is the layout of the switching fabric. Considering the provisions given above, this proposal favours an architecture as shown in Figure 4. The favoured topology is a ring, meshed by several devices with separate input and output. Additional links for paths with high traffic are possible. Topological variations are discussed later. The processing elements, provided with local caches/memories, are loosely coupled and perform stream oriented data processing. A more detailed description is given in section 4.

All devices and processors should perform their tasks as independently from each other as possible in order to support modularity and flexibility. Required independence has certain consequences for attached devices, such as the mass storage. An example of consequences for hard disks is discussed in section 5.

3.1 Bus versus network: the best of two worlds

Not all properties of bus systems limit the performance. The individual properties of buses versus switching networks have to be carefully compared for different typical processes in multimedia environments.

One of the most important aspects in this field is the problem of serial or parallel connections.

Because network technology has focused on separated nodes so far (from several meters to global distances), the data are serialized on bit level and transmitted over single or double lines in all cases. Workstation buses, on the other hand, were designed to cover distances of centimetres, and employ parallel lines for words of 8...64 bit. With the same clock rate, the parallel bus can transmit more data than a serial line, of course, or for a given transmission rate, the technology for parallel transmission would be much easier.

ATM technology known so far uses serial links in the physical layers. But, there is no reason to give up the advantage of parallelity when the internal communication system of a workstation becomes a cell switching network. While it would be a good idea to preserve the ATM cell length of 53 byte, these cells can be easily transmitted in a word-parallel mode. With only 40 parallel lines, all five header bytes would already be transmitted at once (in one clock cycle) and could be recognized in one step.

Another important property of buses is the broadcast character of the data transferred. While it is generally not desired in the internal network to broadcast all traffic, this might be a feature for special purposes (e.g. cache updates). ATM specifies predefined VPI/VCIs for broadcasting purposes [12] which are easily recognizable by simple hardware.

3.2 Microswitches

When the basic idea is to employ cell switching technology for internal communication, the task of switching the cells can be performed by a central instance or by distributed elements. In the latter case, every device needs to be connected to the network by an element that I want to introduce as a *microswitch.*

As this element is new for the architecture of workstations, and because it would be necessary in a certain quantity, the switch needs to be as cheap as possible, comparable to bus latches or amplifiers in today's stations.

The functionality of a microswitch for cell switching would include:

- recognition of cell headers
- comparison of path information (VCI/VPI) to configured pattern
- input: redirection of the payload to the device, if required
- possible generation of empty cells when payload switched into the device
- output: insertion of loaded cells into the network

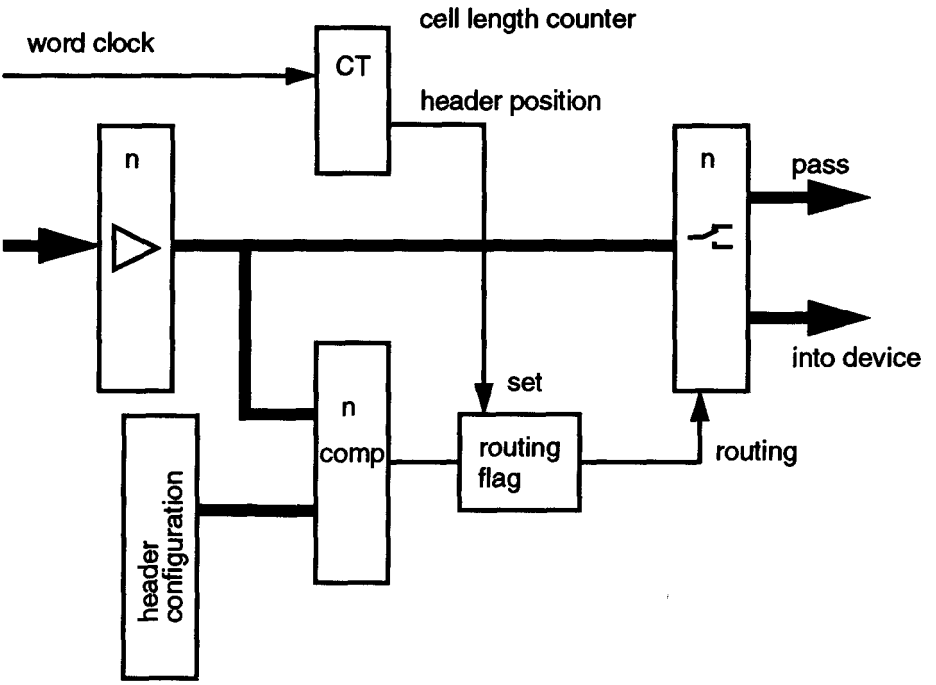

FIGURE 5 Simple microswitch design for device input (parallel)

Figure 5 shows a possible switch design for traffic into a device. The parallelity of n bits forming a word should allow headers to be transmitted in one cycle. A header could be recognized by counting words (as in the figure), or by using a separate control line inside the station. For predefined VCI/VPIs, the word for comparison could be configured during initialization, allowing a few 'don't care' bits for broadcast recognition.

Enhanced switches may be as intelligent as possible, as long as they do not increase the delay too much.

3.3 Topology of the switching fabric

Several topologies could be discussed in order to overcome time multiplexing bus structures. There are rings, meshed networks, crossbars, etc. [11].

A unidirectional ring topology is the first approach to disentangling continuous input and output data streams between different devices. However, the utilization of the ring segments depends very much on the application scenario, as illustrated in Figure 6. For determined scenarios the order of the devices could influence the performance drastically. If, for example, in Figure 6 b the CPU would be placed between the input and the display, the load on the upper ring segment could be reduced for this scenario.

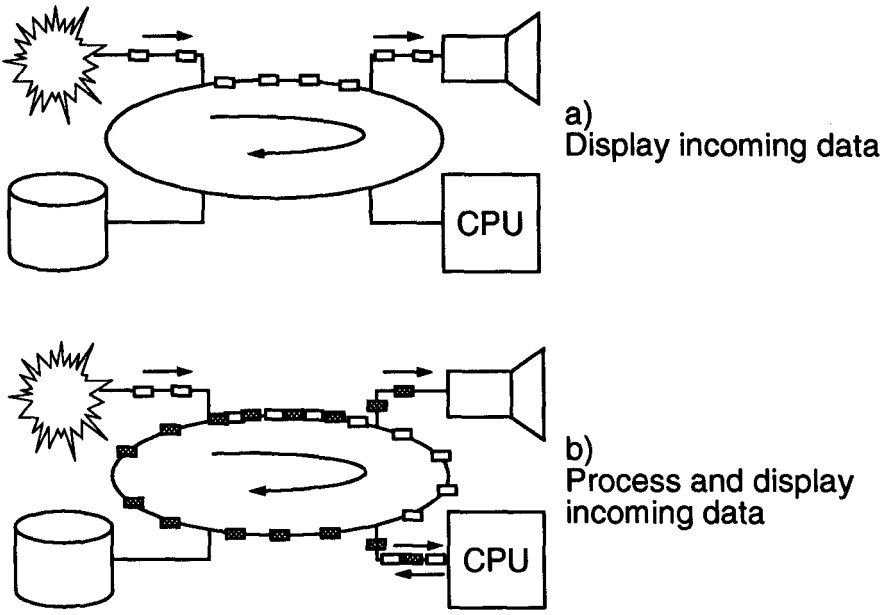

FIGURE 6 Traffic conditions for different application scenarios

The introduction of additional links at critical positions leads to meshed structures, and so do devices with separate input and output connections at different points in the ring. When every device is, or can be, connected with each other, either the complete interconnection topology or at least the structure of a crossbar is reached.

However, crossbars grow to huge devices with hundreds of connection pins when the word parallelity of 32..64 bit lines is required. Connection pins are a very limiting factor in manufacturing integrated circuits. Therefore such a station would not be scalable beyond a few integrated devices, and the cost of adding units grows with the square of their number.

For these reasons, the example of the micronet machine given in Figure 4 is a ring structure, meshed by the separated input and output of the connected devices and processors.

If the topology of a meshed network is configurable in some way (e.g. by reordering the cards in the backplane slots, or by reconfiguring card addresses in EEPROMs), it would be possible to support later even such application scenarios with maximal performance that are not considered today. An architecture open to be reconfigured in order to fit the application's requirements is a more economic approach than directly connecting everything to everything in advance.

4 Stream processing

Multimedia applications are a good example for new data structures. Instead of computing, sorting etc. of financial data spread in the memory, often long streams of continuous media data have to be processed in the same way, e.g. image or audio compression, filtering, value editing. A computing unit which multiplexes the same communication channel for input and output cannot work efficiently when input and output streams are of the same large order near the maximal channel performance.

The processing unit can be the universal general purpose processor, or a specialized device for number crunching, image processing, audio signal processing, compression, FEC, or whatever. It needs to have separate input and output channels for the data, and if possible a separate memory and an input channel for instructions. The data channels should not interfere with each other in the computer's communication system, as they do currently in multiplexed bus systems.

We are very close to the principles of data flow architecture here, because the actions performed by the processors would be triggered by the type of data arriving. A compressed stream of video data arriving at the video processor would trigger decompression, for example.

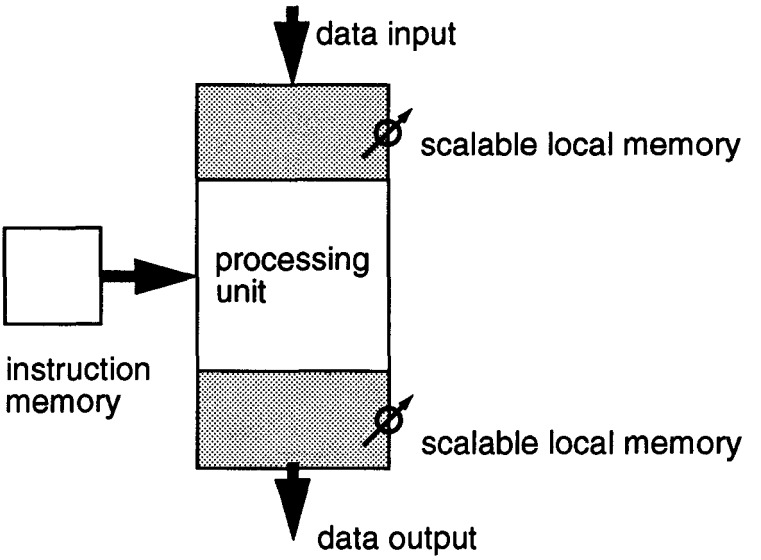

FIGURE 7 Stream oriented Processing Element

Figure 7 shows such a processing element with separate input and output data streams. To be consistent, the instruction stream is also separate, and the code is stored in a special memory associated with the processor. The processing element itself would be classified separately as SIMD (single instruction stream - multiple data stream) in the Flynn classification scheme [11], while the whole micronet machine is one of the dozens of MIMD types.

There are already different types of semiconductor memories in current workstations, e.g. main memory, video memory on graphics cards, disk caches, or several buffers in

communication cards. This indicates a migration process from a centralized main memory to local, dedicated, specialized stores. This process is supported by the (generally) falling price and increasing capacity of semiconductor memories.

So it is quite reasonable to expect each processing element to bring with it all necessary FIFO-buffers, data memories, caches, etc. which are needed for this particular task. This approach reduces the data flow between a central memory instance and the processors.

These architectural elements and data flow principles support the concept of object oriented programming. The data should know how and by which processor they want to be processed. A lot of this information could be coded in the internal VCI structure. The processors hide their internal data in their own memory and have their own software library. In this way modularity is highly supported. The scalability of the local memory mentioned in the figure might be static during the system configuration or, in more sophisticated systems, a dynamic allocation depending on the needs of the processes.

5 Intelligent peripheral devices

Storage media are another subject of intensive research for multimedia requirements. They are a good example of how new concepts for peripheral devices fit into the approach of distributed processing and micronet machines.

The first hard disks were completely dependent on their host processors and their separate disk controllers. Today the controller is usually integrated into the disk device [9]. This made the interface to the host more simple and more efficient and made it possible to integrate even cache memory into the device. But the intelligence of the integrated microcontroller is only used to translate the internal track/sector structure of the device into another imaginary track/sector structure understood by the host operating system. The latter has to administer the file system and often employs an additional disk cache in the main memory. This layering of cache memories does not improve performance, but can add more uncertainty about the success of write processes and the validity of data to the system.

In LAN servers another approach is already used: the client asks for the opening and the storing of files instead of track and sector numbers. This would be possible for the microcontrollers of the disk systems too. The controller in a larger storage systems could also administer multiple drives and caches with different tasks (semiconductor caches, a short-time disk (STD) near the cache, access optimized long-time disk (LTD) arrays).

At idle times the controller can move data between the short-time disk and the long-time arrays and optimize the file structure for fast access, as described in the following.

Whenever the host wants to store data, they are written to the cache. When the file is closed, the controller certifies to the host that it is copied to the STD and therefore protected from system failures. Later, in idle periods, the controller copies the file to the LTD, optimizes and reorganizes the file (and the other files on the LTD) following certain strategies.

Figure 8 shows a larger file system with different drives for the described short-time and long-time tasks, using a disk array for the latter. In smaller systems, the different drives can be replaced by different partitions of the same physical drive for economic reasons.

To optimize all the different kinds of files some new file type attributes need to be introduced. Such types could be:

- asynchronous data files,
- synchronous or continuous media files, possibly with additional parameters (e.g. video or audio which has to be played back with certain time constraints),
- temporary files,
- swap files,
- executables, and so on.

Every type of files needs an individual optimizing strategy. While it might be good for financial files to be written unfragmented on to the disk, the time restraints for video files may require a certain sector interleave for continuous access.

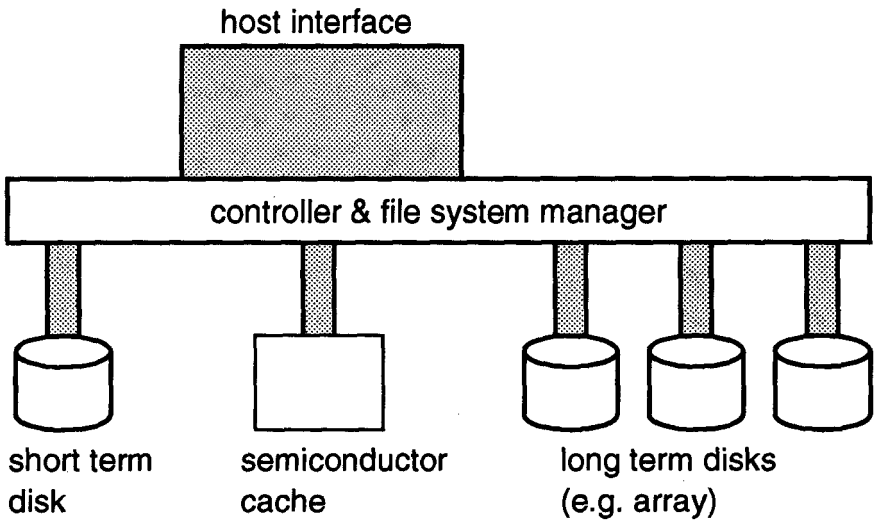

FIGURE 8 Intelligent Storage Devices

Rangan et al. [6, 7] have developed strategies and described experimental implementations of filesystems specifically designed for storing continuous media. They proved that such filesystems need intimate knowledge about timing conditions of the hard disk used. The strategies employ techniques of scattering the media 'strands' over the disk and merging dependent strands of different volume (e.g. synchronized audio and video) together to 'ropes'.

The authors emphasized the problem of the copying overhead during insertions or deletions in order to maintain the storage pattern [8].

It becomes obvious that it cannot be the task of an universal operating system to know such detailed parameters as step times, rotation speed, sector gap length, etc. about all

storage media in the world. This, however, should be the task of the storage medium's own controller. This controller can calculate whatever kind of block allocation or sector interleave is optimal for the specific file type under the conditions of its own timing parameters. Objects to be stored just tell the file system in which way they want to be optimized.

6 Future work

In this paper the idea of micronet machines could be roughly outlined only. A lot of details have to be worked out. As a next step, the possible topologies have to be further compared and simulations have to be done. After the access mechanism for the internal network has been chosen the microswitches have to be designed.

Independently, the proposed storage architecture could be implemented as a prototype. Finally, an experimental system would help to optimize the proposed configuration.

7 References

[1] Covaci, S.; Popescu-Zeletin, R.: Shaping the End-System Architecture for Global Distributed Applications. - 1st IEEE International Symposium on Global Data Networking, Cairo, Egypt, Dec. 13-15, 1993

[2] Covaci, S.; Popescu-Zeletin, R.: The Network-Memory in a Global Distributed Processing System. - for Proceedings of the 4th Workshop on Future Trends of Distributed Computing Systems, Lisboa, Sept. 22-24, 1993

[3] Elias, D.; Gavras, A.: Harmonic Network Access Architecture for ATM. - for Proceedings of the 4th Workshop on Future Trends of Distributed Computing Systems, Lisboa, Sept. 22-24, 1993

[4] Neufeld, G. et al.: Parallel Host Interface for an ATM Network. - in: IEEE Network, July 1993, pp. 24-34

[5] Berenbaum, A. et al.: A Flexible ATM-Host Interface for XUNET II. - in: IEEE Network, July 1993, pp. 18-23

[6] Rangan, P.V.; Kaeppner, Th.; Vin, H.M.: Techniques for Efficient Storage of Digital Video and Audio. - Technical report CS91-209 of USCD. - for Proceedings of 1992 Workshop on Multimedia Information Systems, Temp, Arizona, Feb. 1992

[7] Rangan, P.V.; Vin, H.M.: Designing File Systems for Digital Video and Audio. - Technical report CS 91-191 of USCD. - for Proceedings of the 13th ACM Symposium on Operating Systems Principles, Asilomar, California, Oct. 13-15, 1991

[8] Rangan, P.V.; Vin, H.M.; Ramanathan, S.: Designing an On-Demand Multimedia Service. - in: IEEE Communications Magazine, July 1992

[9] Hospodor, A.D.; Hoagland, A.S.: The Changing Nature of Disk Controllers. - in: Proceedings of the IEEE, Vol. 81 (1993) 4 (April), pp. 586-594

[10] Schnurer, G.: Moderne PC-Bussysteme. - in: c't, 10/1993, pp. 110-119

[11] Baron, R.J.; Higbie, L.: Computer Architecture. - Addison-Wesley: Reading, MA, 1992

[12] Prycker, M. de: Asynchronous Transfer Mode. - Ellis Horwood: New York, 1991