Digital Audio and Video in Industrial Systems

Hugh C. Lauer,* Chia Shen,* Randy Osborne,* John Howard,* Qin Zheng*
Mitsubishi Electric Research Labs, Cambridge, Massachusetts

Morikazu Takegaki,† Hiromitsu Shimakawa,† Ichiro Mizunuma†
Mitsubishi Electric Corporation, Amagasaki, Japan

Introduction

In industrial environments such as power plants, automated factories, sewage treatment facilities, railways, etc., digital audio and video play at least three important roles:–

- *On-line documentation and training.* Pre-stored video in documentation databases is typically viewed interactively, both during routine operation and during emergency situations.
- *Monitoring and surveillance.* Video cameras posted around plants allow operators to keep track of security and proper operation and to provide a visual record for subsequent auditing and analysis.
- *Sensors for plant control.* Video and image processing are being used increasingly in the automated operation of the plant itself — for example, in equipment to measure speeds, count objects, search for production flaws, detect wear of machinery, etc.

In these kinds of settings, it is often desirable to integrate many different functions into the same network — for example, functions or applications with hard real-time requirements, continuous media such as audio and video, functions requiring rapid response, and traditional applications using traditional data protocols such as TCP/IP. At first glance, this may not seem too difficult if one simply dedicates a portion of total network bandwidth to the traffic with hard real-time requirements, then a portion of the remainder to audio and video, etc.

However, bandwidth is only one of the resources and problems that must be considered in a complete network system. Because of the widely different communication requirements of these functions, the demands of their traffic characteristics, flow control, constraints, and performance criteria are typically more challenging than they would be in typical local area or office networks with workstations, PCs, client and server machines, etc.

In this position paper, we discuss a number of issues regarding industrial networks, digital audio and video in those networks, and implications on current research directions. These are considered in the context of ATM (Asynchronous Transfer Mode) networks having speeds ranging from 100 megabits/second to one gigabit/second.

---

*. Address: Mitsubishi Electric Research Labs, 201 Broadway, Cambridge, MA 02139
†. Address: Industrial and Electronic Systems Laboratory, Mitsubishi Electric Corporation, 1-1, Tsukaguchi-Honmachi 8-Chome, Amagasaki, Hyogo, 661, JAPAN

Resource Allocation

Resources in an ATM network include:–

- Bandwidth.
- Buffer space in switches and end systems.
- Priority levels.
- Redundant paths for fault tolerance.

In traditional LANs, resources are typically allocated on demand and shared "fairly" among competing users and/or applications. In industrial environments, resources for critical connections — e.g., hard real-time and some audio and video — are allocated in advance. There are many theoretical solutions to aspects of this problem, especially for allocating bandwidth and buffer space in order to ensure bounded delay, given *a priori* knowledge of the characteristics of the traffic on each critical connection [2, 6]. ATM networks are particularly attractive for implementing such solutions because connections can be controlled to enforce those traffic characteristics.

One kind of resource receives very little attention in traditional LANs, namely redundant paths for fault tolerance. At least two approaches are possible, both of which require assigning primary and alternate paths at the time a connection is created:–

1. When a fault occurs, reconfigure the tables of the switches in the network to reroute each affected connection from its primary to its alternate path. The problem is that re-configuring an ATM switch is time-consuming in today's technology and compounds the problems of fault recovery rather than solving them.
2. At each source, multicast the data stream over both the primary and alternate paths at all times. At each destination, maintain cell counters to detect the forward progress of each branch and to select one. This has the advantage that the alternate paths are continuously exercised and that no reconfiguration is required in any switch at the time of a fault.

Note that in emergency situations, a lot of video streams are likely to be open at one time, imposing extreme demands on the network.

A ripe area for research is to investigate how these theoretical approaches to resource allocation apply in practice, especially in industrial environments at ATM speeds. The biggest gap between theory and practice lies in the assumptions — assumptions that are usually drawn from experience with less demanding networks.

Switching, Scheduling, and Priorities

It is inevitable in a network integrating many different kinds of applications with widely varying requirements that cell-by-cell scheduling is required. Two open questions are

1. how many priority levels are required in practical networks, and
2. whether static or dynamic priorities are required.

In the past, static priority scheduling algorithms such as Rate Monotonic have been preferred because of their closed form solutions to certain classes of problems. Dynamic priority algorithms such as Earliest Deadline First are more general but have been shunned because it was assumed that they are too hard to implement, especially at ATM speeds. In fact, both require the same basic function in the scheduling fabric of the switch, namely the ability to make an n-way comparison among a set of queues

during every cell cycle to determine the queue position or dispatching order, where n is the number of possible priority levels. The question of complexity then reduces to how many priority levels are required in a practical environment and how to implement a fast enough comparator for that many levels. Fortunately, silicon implementations that support a separate priority for every cell in a queue are beginning to emerge [1, 3].

However, there has been very little research on the larger question of how many priority levels are needed in a practical network in a demanding environment, especially with hard real-time connections and a lot of audio and video. There is also scant evidence that vendors are addressing this question seriously. Therefore, it is likely that we will see a long period of evolution of ATM networks, increasingly more sophisticated and probably with growing numbers of priority levels.

End System Interface Requirements

By contrast, network interfaces in end systems — workstations, servers, and "non-intelligent" devices such as video cameras — are getting a lot of attention by vendors and others. One key problem is the performance of higher level protocols — that is, how to handle volumes of data quickly enough. It is now widely recognized that it is absolutely necessary to avoid copying of data between levels of the protocol stack and to avoid interrupting the host processor for each packet.

Less widely recognized is that the network interface device has to perform many of the functions of a switch. On the transmission side, its inputs are the applications and its single output is the physical link to the ATM network. In real time networks and in environments with a lot of audio and video, this interface must support multiple priority levels and be capable of cell-by-cell scheduling, using the same kinds of algorithms as are required in the network switches.

As with switches, the number of priority levels needed in network interfaces in a practical network is still an open question.

Traffic Characteristics and Flow Control

Traffic in most computer networks is extremely bursty and has been characterized as self-similar [5], but in industrial environments it is worse. For example, in a typical system, distributed real-time processes maintain a consistent model of the state of the plant through distributed, reflective, shared memory. This is a special block of memory replicated in each node and kept consistent by low level processes that periodically transmit updates from each node to all others. These updates occur at intervals of one-half millisecond or less and involve a few kilobytes or so. The result is a timely copy of the distributed state of the plant operation to every process.

These update messages are the highest priority communications in the network. They represent bursts that consume the entire bandwidth for frequent, non-trivial amounts of time. Although they are predictable, they are not constant bit rate connections and cannot be converted to constant bit rate connections by traffic shaping.

Lower in priority than the distributed shared memory are the video streams. Uncompressed video streams are fairly easy to schedule because they require constant bit rate connections. The only apparent complication is jitter due to interference by higher priority connections, something that can be avoided with enough buffering in

end systems. Compressed video streams are more interesting. For example, compression algorithms such as MPEG generate streams with bandwidth variations of a factor of ten from frame to frame. This has two effects:– it makes resource allocation more difficult, particularly when planning for emergency situations, and it makes flow control of lower priority connections more challenging. Note that lossy video is rarely acceptable in the industrial environment, and therefore methods that throw away video frames during periods of network congestion are not applicable.

It is natural to want to use surplus capacity of the industrial network for more traditional data traffic that does not have real-time constraints. In ATM parlance, this is called *ABR* (Available Bit Rate) traffic. In general, users of ABR applications want to be able to request all of the uncommitted capacity of the network at any time, on demand, without prior reservation, and subject only to fair sharing with the other ABR connections. The network, of course, must ensure that this traffic does not get in the way of real-time or audio and video traffic.

Flow control for this purpose has been the topic of extensive research and discussion at the ATM Forum. Although the ATM Forum recently voted to adopt a rate-based approach to flow control, the debate is far from over. In particular, it is already apparent that this will not work very well in the industrial environment because of the worse-than-bursty nature of traffic. The distributed shared memory updating processes and the variable bandwidth of compressed video create transients in the "available" bandwidth that are much shorter than the response times of rate-based flow control algorithms. This is considered in more detail in [4].

References

1. H. Jonathan Chao, Necdet Uzun, "A VLSI Sequencer Chip for ATM Traffic Shaper and Queue Manager," *IEEE Journal of Solid-State Circuit*, Vol. 27, No. 11, Nov. 1992.

2. D. Ferrari and D. Verma, "Real-Time Communication in a Packet-Switching Network," *Proc. Second International Workshop on Protocols for High-Speed Networks*, Palo Alto, November 1990.

3. H. Lauer, A. Ghosh, and C. Shen, "A General Purpose Queue Architecture for an ATM Switch," *Proc. of First Annual Conference on Telecommunications R&D in Massachusetts*, Massachusetts Telecommunications Council, University of Lowell, October 25, 1994, Vol. 6, pp 17-22.

4. H. Lauer, "On the Duality of Rate-based and Credit-based Flow Control," *Technical Report* TR95-03, Mitsubishi Electric Research Labs, Cambridge, MA, January 1995. (Submitted for publication.)

5. W. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic," *IEEE/ACM Transactions on Networking,* volume 2, number 1, February, 1994.

6. Q. Zheng, K. Shin, and C. Shen, "Real-time Communication in ATM Networks," *19th Annual Local Computer Network Conference,* Minneapolis, MN, October 2-5, 1994.