An Experimental
Flow-Controlled Multicast ATM Switch

T. Blackwell, K. Chan, K. Chang, T. Charuhas, B. Karp,
H. T. Kung, D. Lin, R. Morris, M. Seltzer, M. Smith, and C. Young
Harvard University, 29 Oxford Street, Cambridge, MA 02138

O. Bahgat, M. Chaar, A. Chapman, G. Depelteau, K. Grimble, S. Huang,
P. Hung, M. Kemp, I. Mahna, J. McLaughlin, T. Ng, J. Vincent, and J. Watchorn
Bell-Northern Research, Ottawa, Ontario, Canada K1Y 4H7

Abstract

BNR and Harvard have jointly designed an experimental ATM switch called *CreditSwitch* with sixteen 622-Mbps ports. Expected to be operational in early 1995, the switch will support credit-based flow control and full-speed multicast. This paper gives a brief overview of the switch architecture and its design goals.

1. Introduction

ATM switches will need a new level of sophistication to handle a high-volume mix of bursty data and multimedia traffic under, respectively, ABR ("Available Bit Rate") and CBR ("Constant Bit Rate) or VBR ("Variable Bit Rate") services. Such traffic will require switch support for scheduling, policing, congestion control, and multicast. Not only should a switch support these features, but it should support them in a cost-effective manner.

Harvard and Bell-Northern Research (BNR) have designed an experimental ATM switch with sixteen 622-Mbps (Mega bit per second) ports to study new switching methods to meet these requirements. As of October 1994, all the board types have been designed and manufactured, and system integration is under way. The first copy of the switch is expected to be operational in early 1995.

This effort is part of the CreditNet project. BNR and Harvard are developing the ATM switch (CreditSwitch) described in this paper, while CMU and Intel are developing an ATM host interface for the PCI bus standard.

After a short discussion of some basic concepts, we describe our switch architecture with a focus on buffer management and traffic scheduling issues. The paper ends with a brief overview of some interesting aspects of the switch implementation.

2. Some Basic Concepts

ATM *s* a transport protocol based on the notion of a connection between two entities, called a *virtual circuit* or VC. ATM transports data in fixed-size cells, each with a circuit identifier. A network makes decisions, for purposes such as resource allocation, based on circuits, rather than on independent packets of data.

An ATM network is typically made up of a mesh of hosts and switches connected by links; a host has one link to a nearby switch, but each switch may be connected to multiple hosts and other switches. Thus a circuit may traverse many switches and links between the two hosts that it connects.

A switch has a port connected to each of its links. To implement bidirectional links, a port conceptually has an input half and an output half. The job of the switch's transport mechanism is to copy each incoming cell to the appropriate output port, based on the cell's circuit identifier.

3. CreditSwitch Architecture

As depicted in Figure 1, the CreditSwitch has sixteen 622-Mbps ports, supporting 15 port modules and one switch control module. Thus each port module runs at a speed comparable to that of many workstation I/O busses. If used as part of a WAN, each port module is fast enough to multiplex dozens of conversations at the speeds typical of WAN access links.

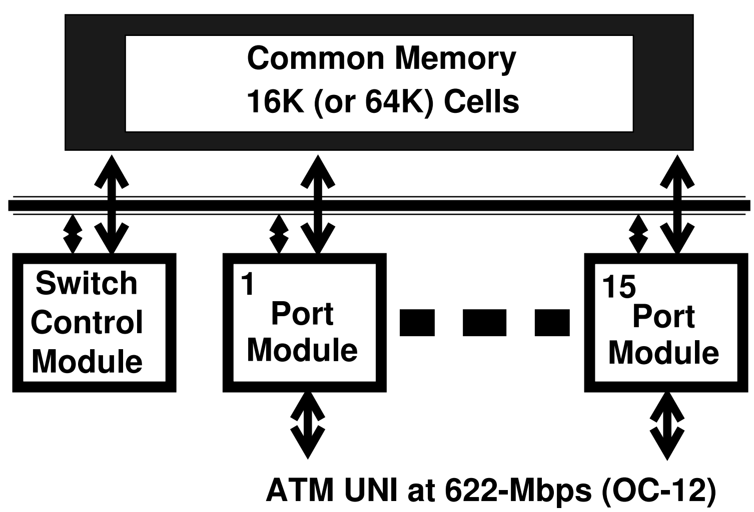

Figure 1: Architecture overview of the CreditSwitch

The switch buffers incoming cells from any port module in a shared *common memory*, which is fast enough to send and receive a cell for each port in one cell transmission time. When a cell arrives at the switch, the input port broadcasts the cell's circuit identifier and address in the common memory on a backplane. Each output port monitors this backplane; when a port notices that a cell has arrived for a circuit that leaves that port, it adds the cell's memory address to a queue. With some limitations that should have negligible impact on performance, up to 16 cells from different inputs can be added to an output port's queue in a single cell time. An output port also reads a cell from the memory every cell time and sends it over the output link.

Multicast is an important basic service that switches must provide in order to support wide-area distribution of audio, video, and possibly data. While it might be possible for a single server to send the same video stream to a few customers by simply sending each cell once for each customer, this does not scale to tens or hundreds of customers. The CreditSwitch will support wide-area distribution to an unlimited number of endpoints through the use of switch-level multicast. By performing the replication at the switches throughout the network, cells are only sent over a given link once.

A single circuit can sustain the full 622-Mbps rate through the switch if the two ports involved are otherwise idle; up to 16 such independent conversations are possible. Even with more complex traffic patterns the switch can achieve high utilization, since it is free from head-of-line blocking. The main threat to efficiency under heavy load is that the switch will run out of space to buffer temporary (or permanent) bursts of traffic.

4. Buffer Memory Management

Allocating enough switch buffer space to allow the switch to buffer traffic bursts and only rarely run short of memory does not appear to be practical, as the requirements for high-speed memory can become very large. An alternate solution is to provide a fairly large amount of buffering, and to discard data in a sensible way under sustained overload, to force traffic sources to regulate their output. The CreditSwitch can support a variety of discard policies. It monitors the buffer use of individual circuits, and can enforce progressively more drastic discard policies on offending circuits as the memory fills up.

The above approach assumes that the traffic sources will eventually notice the dropped cells, interpret this as an indication of congestion, and reduce the load they offer to the network. However, even sophisticated protocols such as TCP can take a long time to react to congestion. In addition, some kinds of traffic, such as video, may not react at all.

To solve these problems, the CreditSwitch supports a credit-based flow control system [2, 3]. The switch monitors the buffer use of each circuit, and provides feedback to the immediately preceding switch or host along the circuit's path. Since each switch has precise knowledge of the resources a circuit is consuming, and the feedback loop is only one link long instead of the length of the entire end-to-end connection, this flow control system allows much more efficient use of buffer memory and link bandwidth. A companion paper describes the ideas behind this mechanism and presents some simulation results [1].

5. Scheduling

Under heavy load, a switch must make sensible decisions about the order in which it sends traffic. For traditional data traffic, it may be enough that the switch allocates bandwidth fairly among the competing circuits. Networks that carry video or voice, or public networks in which customers pay for a specific level of service, require more sophisticated guarantees. An additional constraint is that the CreditSwitch's flow control mechanism requires that data be sent on a circuit only if the next node has space to buffer it.

The switch achieves these scheduling goals with two mechanisms. Each output port maintains a separate queue of cell addresses for each circuit, i.e., the switch implements *per-VC queueing*. If each output had only a single queue, it would be forced to send the cell at the head of the queue. The decision of which circuit's queue to service next is made by a microprocessor, based on a hardware summary of which circuits have data queued and also have buffer space downstream based on credit information. The microprocessor's software can implement a variety of schedules, including round-robin among circuits, strict priority, and weighted priority. This flexibility will support research into how service-level guarantees can best be implemented.

6. Implementation Overview

As shown in Figure 1, the switch is physically organized as 16 modules that plug into a backplane and memory buffer system. One of the modules is the switch control module for call processing using the Q93B signalling standard.

The rest of the modules are port modules. Each port module has two microprocessors, one for scheduling mentioned above and one to handle real-time monitoring and control. These two microprocessors are not necessary for the implementation, but they provide the programming flexibility necessary to study many research issues. For example, these processors provide the flexibility to experiment with different ways of observing and reacting to network load conditions. Each port module also has a fiber-optic link interface using SONET framing. The cell handling hardware is built from field-programmable gate arrays for control, and static RAMs for tables and queues.

When a cell arrives at the switch, the input port broadcasts the cell's circuit identifier and address in the common memory on the arrival bus on the backplane. Each output port monitors this backplane; when a port notices that a cell has arrived for a circuit that leaves that port, it adds the cell's memory address to a queue.

When a cell leaves an output port, its circuit identifier is broadcast on the departure bus on the backplane. By watching the arrival and departure busses, each input port maintains a count of the number of cells buffered for each circuit that enters that port. This count is used both to provide credit-based flow-control feedback and to decide which circuits are using so much memory that their data should be discarded.

The common memory architecture allows the CreditSwitch to support multicast in an efficient way. A common memory allocation engine maintains a list of free locations in the shared common memory. Entries from this list are allocated to cells as they arrive. When a multicast cell's information is broadcast on the arrival bus, more than one port module will enqueue this cell in its list of cells to send. However, the cell only requires one common memory location.

The allocation engine hands out addresses of free slots in the common memory to the ports on demand, so they can store incoming data. When it does this, it initializes a list of egress ports that must send this cell out. When a port sends out a cell, the presence of the cell's identifier on the departure bus tells the allocation engine to remove it from the list. When the list becomes empty, the common memory location is recycled for future use. All this is done by some efficient hardware: the allocation engine requires only four memory accesses per port per cell cycle.

For most purposes, the ingress and egress sides of a port are effectively independent. However, they have an important interaction required for the credit protocol. Essentially, the credit protocol requires a sender to have a credit for a given VC, before sending cells on it. Credit is granted by sending credit cells opposite the flow of data (from receiver to sender.) Thus, when the ingress side of a port realizes that a number of cells for that VC have left the switch, it notifies the egress side of the same port to send a credit cell.

7. Concluding Remarks

This experimental switch's fundamental purpose is to answer questions about trade-offs in high-speed ATM switching architecture. One of the most important trade-offs is that between efficiency and sophisticated control mechanisms. For instance, the switch uses one important mechanism, per-VC queuing in output ports, as the basis for flow-control, scheduling, and multicast. The switch would be simpler without this mechanism, but it could make substantially poorer use of buffer memory and link bandwidth, and could suffer from severe fairness problems. Experimental results of CreditSwitch will be reported in future articles.

Acknowledgment

This research was supported in part by BNR, and in part by the Advanced Research Projects Agency (DOD) monitored by ARPA/CMO under Contract MDA972-90-C-0035 and by AFMC under Contract F19628-92-C-0116.

References

[1] T. Blackwell, K. Chang, H. T. Kung and D. Lin, "Credit-Based Flow Control for ATM Networks," in these Proceedings.

[2] H. T. Kung, and A. Chapman, "The FCVC (Flow Controlled Virtual Channels) Proposal for ATM Networks." A summary appears in *Proc. 1993 International*

Conference on Network Protocols, San Francisco, California, October 19-22, 1993, pp. 116-127. (The postscript file of is available via anonymous FTP from virtual.harvard.edu:/pub/htk/atm.)

[3] H. T. Kung, T. Blackwell, and A. Chapman, "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing," *Proceedings of ACM SIGCOMM '94 Symposium on Communications Architectures, Protocols and Applications*, August 31-September 2, 1994, pp. 101-114. (The postscript file of is available via anonymous FTP from virtual.harvard.edu:/pub/htk/atm.)