A new cut-through forwarding mechanism for supporting ATM multipoint-to-point connections

A. Papadopoulos, T. Antonakopoulos and V. Makios

The IEEE International Conference on Networking - ICN'01

COLMAR, FRANCE, JULY 2001

A new Cut-through forwarding Mechanism for ATM Multipoint-to-Point Connections

Athanasios Papadopoulos[1], Theodore Antonakopoulos[2] and Vassilios Makios[2]

[1]Computers Technology Institute, Riga Feraiou 61, 26100 Patras, Greece
[2]Department of Electrical Engineering and Computers Technology, University of Patras, 26500 Rio Patras, Greece
Tel: +30 (61) 997 346, Fax: +30 (61) 997 342,
antonako@ee.upatras.gr

Abstract. This paper presents a new Cut-through forwarding mechanism for multipoint-to-point communications over ATM networks. Initially, we present the cell interleaving problem and a concise description of current ATM multicast proposals. Then, we present an improvement to existing Cut-through forwarding schemes for providing a good level of fairness among the connections competing for the available bandwidth of the output link, while maintaining good throughput performance. This scheme is referred as Conditional Cut-through forwarding (CCT) and as simulation results show, it supports multipoint-to-point connections efficiently.

1 Introduction

ATM networks use various operational modes, like unicast, broadcast and multicast in order to support different application requirements. Multicast serves as a communication abstraction, allowing message delivery to multiple destinations in a single step and thus reduces overall bandwidth consumption, helps control network traffic and decreases the amount of processing at every host. Efficient implementation of multicast is useful in many applications, such as distributed computing, parallel discrete event simulation (PDES) and multimedia applications.

ATM multipoint communications have been studied at various international organizations, like ATM Forum and ITU. The Internet Engineering Task Force (IETF) also studies the mapping of IP multicasting to ATM networks [1], but direct ATM multicast service is still in its early phase of definition. The ATM User-to-Network Interface (UNI 3.1 or 4.0) signalling supports source-based tree approach with point-to-multipoint virtual channels (VCs) [2]. In current ATM implementations, AAL5 does not have any provision within its cell format for the interleaving of cells from different packets to a single connection. Therefore, point-to-multipoint connections are unidirectional, permit the root (a single source) to send data to the leaves (multiple destinations) but do not permit the leaves to transmit to the root on the same connection. In order to reduce the number of virtual connections maintained for each group, ATM switches must support multipoint-to-multipoint connections.

The point-to-multipoint connections supported by UNI achieve multipoint-to-multipoint communication using either a centralized multicast server, or many point-to-multipoint VC connections in order to completely connect hosts in a mesh topology. Both of these approaches have efficiency limitations, since either they have to perform data retransmission (in the multicast server case), or they have to use many network resources in order to establish the required connections (in the case of overlaid VCs). In order to use multipoint-to-point connections, where many input links are mapped to an output link, we must solve the cell interleaving problem: when traffic is merged on a multipoint-to-point connection, cells that belong to packets from different sources use the same VPI/VCI and may interleave at the receiver site and the AAL5 entity cannot reassemble the data.

2 ATM Multicast Proposals

AAL5 is the mostly employed ATM protocol. Several approaches have been proposed for solving the cell-interleaving problem over AAL5. These mechanisms include VC merging, VP switching, AAL5 modifications, use of resource management cells and the use of sub-channels within a VC.

VC merging uses buffering of cells at the network switches and individual forwarding per packet. The MPLS (MultiProtocol Label Switching) and SEAM (Simple and Efficient ATM Multicast) proposals follow this technique. MPLS [3] implements a Store-and-Forward technique, while SEAM [4] aims to improve the performance of MPLS with a Cut-through forwarding algorithm. VP switching uses the VCI field to identify the sender and the VPI field to forward the cells. In schemes like DIDA (Dynamic IDentifier Assignment) [5], a value identifies the cells of a specific frame, while in the improved VP Switching proposal [6], the VCI identifies the sender, not the frame.

Modifications of AAL5 add new fields in the cell for multiplexing or change the current fields of the cell header. Simple Protocol for ATM Multicast (SPAM) [7] inserts a 15 bits MID (Message Identifier) field in the cell payload to distinguish the cells coming from different senders. In the AAL5+ method [8], the MID field is 16 bits long and its value is assigned per packet by using a uniform probabilistic function.

Other proposals use Resource Management (RM) cells. In SMART [9], a host must hold a token (control message) before it starts transmitting data to a tree. In CRAM [10], each group of cells belonging to the same sender is preceded by a Resource Management (RM) cell, which contains a number of Sender Identifiers. In the Sub-channel Multiplexing technique [11], 4 bits from the Generic Flow Control field in the ATM cell header are used to carry the multiplexing identifier (sub-channel ID) to distinguish between multiple sub-channels in a VC.

All approaches require some modifications either to the ATM switches or to the edge devices. Some of them suffer from excessive overhead (SPAM, AAL5+) and high complexity (SMART, CRAM). VPs should not be used by edge-devices (DIDA, Improved VP), because they are used for the accumulation of traffic in the backbone. Also, the sub-channel multiplexing technique is not very scalable, since only fifteen simultaneous senders can use the VC. Proposals based on Store-and-Forward and Cut-through algorithms (MPLS, SEAM) add small delay to data traffic [12], are scalable and simple to implement in order to support multipoint communications. As an additional benefit, these approaches can enhance intelligent discarding schemes (Early Packet Discard or Partial Packet Discard – EPD/PPD) by reducing the number of partial transmitted packets. This capability reduces further the wasted bandwidth since the EPD/PPD packet dropping policies reduce congestion and transmission of traffic that would inevitably be retransmitted [13].

2.1 Cut-through forwarding

The Cut-through mechanism avoids the cell-interleaving problem in multipoint-to-point VCs by forwarding the first cell of a packet before receiving the End-of-Packet (EOP) cell, whenever the outgoing VC is idle and continues to transmit cells until the last cell of the packet. The ATM switch, which keeps separate buffers for each sender, has to buffer other incoming packets for the same outgoing VC, until the current packet has been forwarded completely. However, if a long packet from a slow source is forwarded or if the last cell has been lost, the delay increases, since cells belonging to other packets must wait in the input queues until the EOP cell of the first packet arrives, as shown in Figure 1.

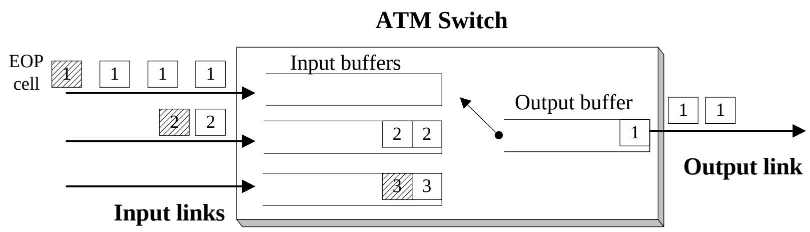

Fig. 1. Large packets from slow sources increase the delay in Cut-through forwarding

SEAM proposes a timer to overcome the loss of an EOP cell and the Store-and-Forward mechanism for slow links. But the value of timer is really critical and could significantly impact buffer lengths into switches. If the duration of the timer is too short, it would lead to an undesirable discard of good packets, and if it is too long, the delay of buffered packets would increase seriously. Additionally, slow links do not exploit the advantage of the cut-through mechanism, which is the immediate transmission of a new cell, and slow sources from faster links can still block the procedure. The modifications of the Cut-through discipline that are proposed by Stolyar in [14] do not solve these issues.

3 The Conditional Cut-through (CCT) forwarding scheme

The design goal of the Conditional Cut-through (CCT) forwarding scheme is to minimize the delay of buffered packets that is due to large packets from slow incoming links and sources or due to the loss of an EOP cell. In CCT, we define the variable "time in buffer" for each incoming packet. The variable "time in buffer" measures the number of arrived cells and is increased by 1, whenever a new cell arrives into the input buffer. When an EOP cell arrives into the buffer, the packet is marked as "Ready". Then, the "time in buffer" has its maximum value and is equal to the total number of cells for that specific packet. It is obvious that each packet has its own "time in buffer".

When packets come from a source that generates cells at a rate less than the maximum supported rate (defined by the link speed), contiguous cells within a packet will be spaced by idle slots. We assume that the switch measures and stores the cell input rate for each partially forwarded packet and for every new packet that arrives into any empty buffer. Additionally, the switch measures and stores the mean and max values of the "time in buffer" for each input buffer separately.

When the switch receives the first cell of packet X and if there are no cells from other sources queued or being forwarded, it directly forwards all cells on the output buffer without queuing. The switch copies each cell of the partially forwarded packet into a backup buffer. During the transmission of packet X, all the cells of a new packet Y arrive also into an empty buffer. Then, the switch will stop the packet forwarding and will start transmitting the new packet Y, if all the following conditions are satisfied:

a. the "Ready" packet Y has less "time in buffer" than the current "time in buffer" of the partially forwarded packet X,

$$Y_R < X_C \qquad (\mathbf{1})$$

b. the "Ready" packet Y has arrived with higher cell input rate than X,

$$CIR_Y > CIR_X \qquad (\mathbf{2})$$

c. the "time in buffer" of the current forwarded packet is less than the mean value of the corresponding input buffer,

$$X_C < X_{mean} \qquad (\mathbf{3})$$

The switch transmits a "null cell" to indicate to the receiver that the previous cells have to be rejected. The stopped packet X is stored into the backup buffer until the EOP cell of packet arrives. Then, the switch transmits the packet X, irrespective of the status of other packets.

If the transmission of a packet has been completed successfully, the next packet is selected by the "time in buffer" of the awaiting "Ready" packets. The switch compares all the first-in-buffer "Ready" packets among them. The packets with smaller "time in buffer" (smaller packets) have higher priority. The switch must transmit all the "Ready" packets from the results of the current comparison, before it moves on to a new comparison with new incoming "Ready" packets.

In case the previous conditions are not satisfied, the loss of an EOP cell could block the procedure. Then, if there is an input buffer Y with two or more "Ready" packets and their total "time in buffer" satisfies the condition:

$$Y_R + Y_{R+1} + Y_{R+2} + \ldots > \max\{X_{max}, Y_{max}\} \quad \textbf{(4)}$$

then the first packet of this buffer will stop the forwarding action. According to Condition (4), where Y_{R+1} is the "time in buffer" of the second incoming "Ready" packet in buffer Y, CCT gives to packet X enough time to be forwarded successfully, taking into account the case when the size of packets between two buffers differs significantly. When an EOP cell loss event occurs, the packet would be eventually discarded by switch. Consequently, the other input links are not blocked waiting for the output link to be released.

A system using the CCT forwarding scheme is shown in Figure 2. We assume that packet 1 is currently being served and buffers 2 and 3 have no complete packets. Packet 1 is forwarded even though it has not arrived completely and it is copied into the backup buffer. During the transmission, the EOP cell of packet 3 arrives. According to the CCT algorithm, if conditions (1) to (3) are satisfied, the switch generates the "null cell" to indicate the rejected cells and it starts transmitting packet 3.

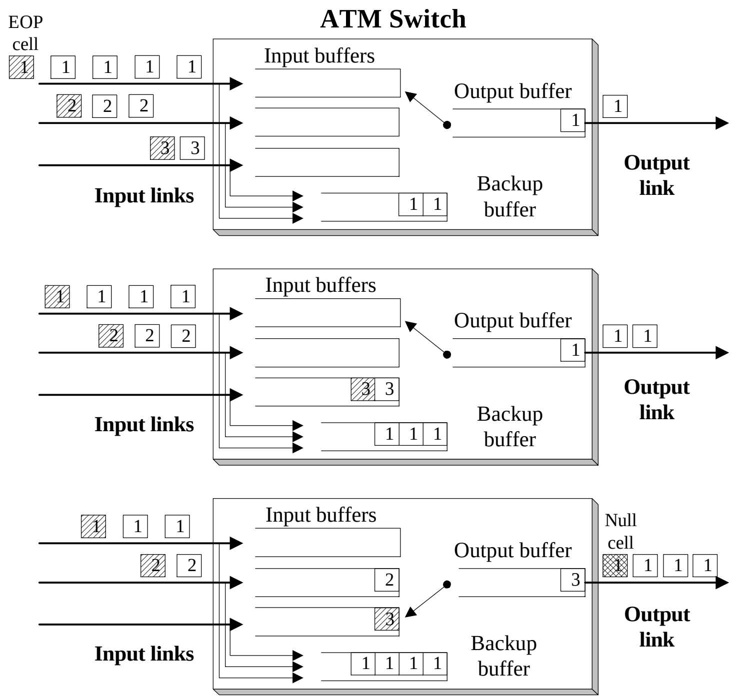

Fig. 2. The Conditional Cut-through (CCT) forwarding scheme

A generic flow chart for CCT forwarding scheme is shown in Figure 3. CCT gives higher priority to smaller packets from faster links or sources. The algorithm does not wait the completion of packet X. Once the EOP cell of packet Y arrives, it compares the “time in buffer” values and the cell input rates for the two packets. If the conditions are satisfied, the switch stops the forwarding procedure and packet Y is transmitted to the output link immediately. Thus, the delay for packet Y is almost negligible.

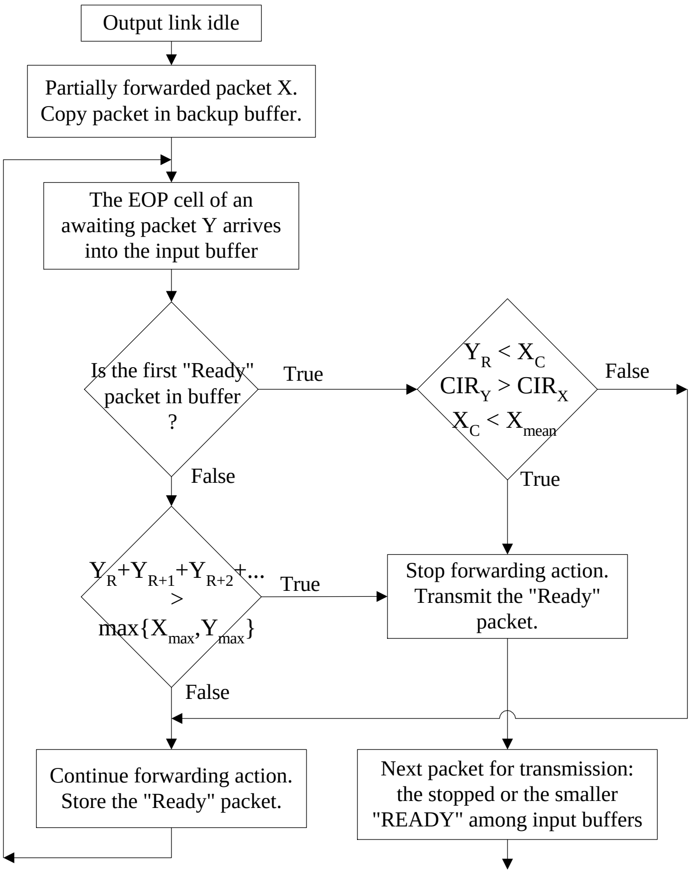

Fig. 3. A generic flow chart of the CCT algorithm

Whenever a Cut-through transmission begins, the switch calculates the cell input rate only for the partially forwarded packet and for the packets that arrive into empty buffers. After a successful transmission, the switch always checks the backup buffer if the stopped packet has been completely received. Thus, the CCT scheme does not add any additional delay to the stopped packet. The switch copies only the cells of the partially forwarded packet into the backup buffer. The goal is to resolve the fairness issue and to achieve better QoS support.

4 Comparative Results

Simulations results are used to analyze the proposed method's performance. We compare the effectiveness of the CCT scheme to the Store-and-Forward and other Cut-through mechanisms under homogeneous and heterogeneous input sources and links. The sources follow an ON-OFF model. OFF periods represent inter-packet generation time. We simulate a single switch with 5 input links and 1 output link. The fast recovery from slow arriving large packets is the most critical factor that characterizes CCT performance. Thus, we extensively investigate the impact of slow sources, the packet length and the load of slow links on the effectiveness of VC merging schemes. The simulation tool that was used for defining the most appropriate solution is Compuware's COMNET III ® since it includes libraries for supporting Store-and-Forward and Cut-through mechanisms.

Initially, all links are assumed to be of the same speed. We consider a data rate of 155.52Mbps for all input links. Also, the output link has the same speed. The packet length is a random variable uniformly distributed in the range (2, 30) cells and the total offered load would vary from 0.1 to 0.8. Since the average packet length is the same for all sources, OFF period determines the offered load. Figure 4 shows the average cell delay under various loading conditions when each source produces a cell at every time slot during the ON period. Cell delay includes storing time in the input buffer, the cell scheduling time and the transmission delay.

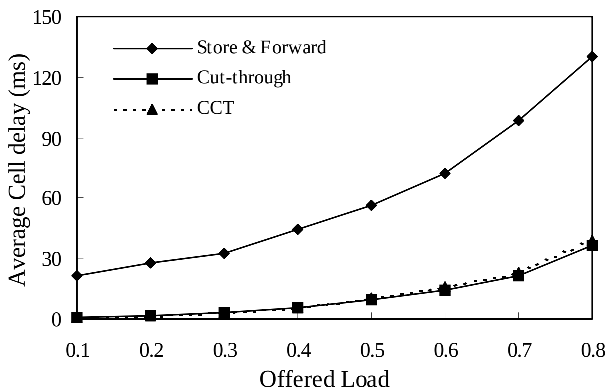

Fig. 4. Average cell delay versus network load under homogeneous sources

When the sources are homogeneous, CCT and Cut-through algorithms have almost the same behaviour. Cut-through performs slightly better since the CCT algorithm has the overhead of copying the cells of the partially forwarded packet into the backup buffer. However, when two of the sources generate the cells of a packet at the rate of 0.7 of link speed (slow sources), the results differ significantly, as shown in Figure 5. The packet length remains the same for all sources.

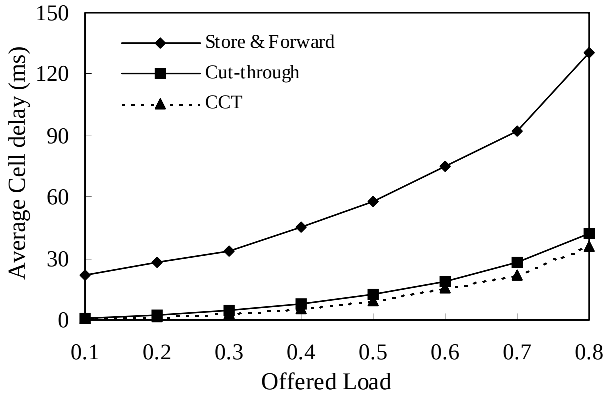

Fig. 5. Average cell delay versus network load when two of the sources are slow

In Figure 5, we see that the CCT algorithm has better response in all loading conditions. The delay due to Cut-through increases rapidly as the load increases and packets from slow sources block the procedure. CCT performs much better at high loads, since small packets are transmitted immediately.

Figure 6 illustrates the average cell delay for different mean packet sizes under constant offered load (at 0.5). We assume that the average packet length for all sources would vary approximately 2 to 5 times the packet size used in Figure 5. We keep constant the minimum value of cells and we increase the maximum value. The two sources remain slower than the other sources.

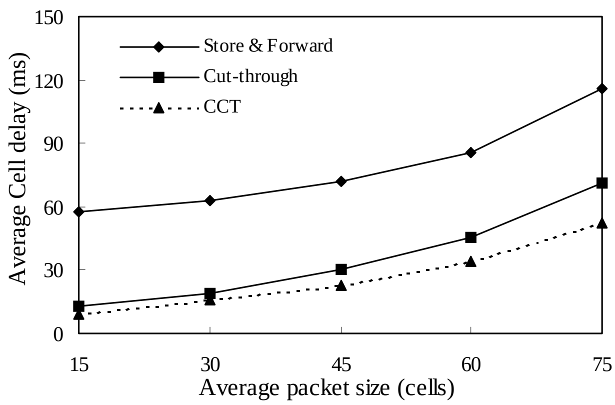

Fig. 6. Average cell delay for different packet sizes

The results verify the effectiveness of CCT algorithm versus Cut-through when the packet sizes increase. Cut-through scheme depends directly on the length of incoming packets. As the packet size increases, the performance of Store-and-Forward and Cut-through schemes decreases.

Finally, we consider that 2 of the inputs links are slow, having half the speed of the other input links and of the output link. We keep constant (equal to 0.1) the offered load of high-speed input links and we change the offered load of slow input links from 0.1 to 0.5.

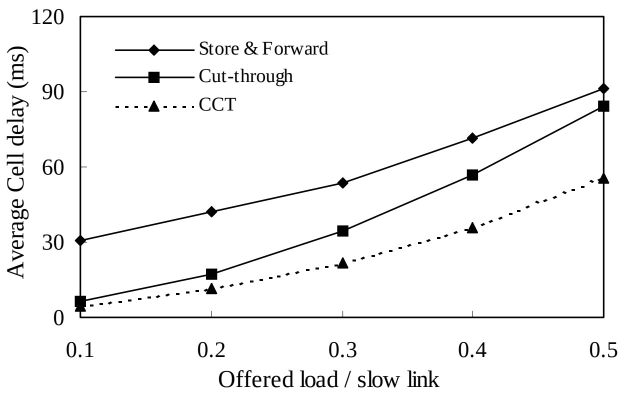

Fig. 7. Average cell delay versus utilization of slow links

In Figure 7, we see that under high load conditions at the slow links, the performance of CCT is much better that Cut-through. Further, we find out that Cut-through tends to follow the Store-and-Forward performance under high load conditions. Figure 7 shows the limitation of Cut-through to support slow-links and the effectiveness of CCT algorithm.

5 Conclusions

In this paper, we described a simple and efficient solution to the cell-interleaving problem in multipoint-to-point connections. VC merging techniques are more scalable and simple to implement in order to support multipoint communications, but Cut-through forwarding results to waste of bandwidth if the subsequent cells of the partially forwarded packet are delayed. The design goal of the CCT algorithm is to minimize the delay of buffered packets that is due to slow arriving large packets or due to the loss of an EOP cell. The proposed scheme provides better performance in terms of delay, has low implementation complexity and does not require any modifications to the cells' structure. The CCT scheme requires an additional buffer for each output link and increases its complexity slightly.

References

1. G.Armitage, “Support for Multicast over UNI 3.0/3.1 based ATM Networks, IETF RFC 2022, November 1996.
2. P.Samudra, “UNI signaling 4.0”, ATM Forum 95-1434, July 1996.
3. E.Rosen, A.Viswanathan and R.Callon, “Multiprotocol label switching architecture”, IETF Draft, <draft-ietf-mpls-arch-06.txt>, August 1999.
4. M.Grossglauser and K.K.Ramakrishnan, “SEAM: Scalable and Efficient ATM Multicast”, Proceedings of IEEE Infocom’97, pp.867-875, April 1997.
5. J.Calvignac, P.Droz, C.Basso and D.Dykeman, “Dynamic IDentifier Assignment (DIDA) for Merged ATM Connections”, ATM Forum 97-0504, July 1997.
6. R.Venkateswaran, C.Raghavendra, X.Chen and V.Kumar. “Support for Multiway Communications in ATM Networks”, ATM Forum 97-0316, April 1997.
7. S.Komandur and D.Mosse, “SPAM: A Data Forwarding Model for Multipoint-to-Multipoint Connection Support in ATM Networks”, Proc. of the 6^{th} IC3N, September 1997.
8. F.Hoymany and D.Mosse, “More Powerful ATM Connectionless Support Using AAL5+”, Proceeding of IASTED Networks, January 1996.
9. E.Gauthier, J.Y.Le Boudec and P.Oeschlin, “SMART: A Many-to-Many Multicast Protocol for ATM”, IEEE Journal on Selected Areas in Communications, pp. 458-472, April 1997.
10. S.Komandur, J.Crowcroft and D.Mosse, “CRAM: Cell Re-labeling At Merge Points for ATM Multicast”, Proc. of IEEE ICATM’98, June 1998.
11. J.Turner, “Extending ATM Networks for Efficient Reliable Multicast”, Proceedings on Communication and Architectural Support for Network-Based Parallel Computing, February 1997.
12. I.Widjaja and A.I.Elwalid, “Performance Issues in VC-Merge Capable Switches for Multiprotocol Label Switching”, IEEE Journal on Selected Area in Communications, Vol.17, No.6, pp.1178-1189, June 1999.
13. A.Romanow and S.Floyd, “Dynamics of TCP traffic over ATM Networks”, IEEE Journal on Selected Area in Communications, Vol.13, No.4, pp.633-641, May 1995.
14. A.L.Stolyar and K.K.Ramakrishnan, “The stability of a flow merge point with non-interleaving cut-through scheduling disciplines”, Proceedings of INFOCOM’99, pp.1231-1238, March 1999.