Compound VC Mechanism for Native Multicast in ATM Networks*

Josep Mangues-Bafalluy and Jordi Domingo-Pascual

Computer Architecture Department – Polytechnic University of Catalunya
Campus Nord. Mòdul C6. Despatx E102.
Jordi Girona 1-3 – 08034 Barcelona (Spain)
Phone: +34 934011638 | Fax: +34 934017055
Email: jmangues@ac.upc.es and jordi.domingo@ac.upc.es

Abstract : The growth of new kinds of traffic, and particularly multimedia, leads to changes in the conception of the network. Multimedia group communications pose stringent requirements to the distributed system supporting them. Maybe the most important requirement is that they should allow the communication from a user in a group to the rest of the group in an efficient way. These communications are referred to as multicast communications. The most efficient way to offer them in terms of resource consumption is at the level where either routing or switching are carried out. ATM poses further challenges due to its connection-oriented nature. Some mechanisms have been proposed in the past either for integrating IP multicast with ATM or for providing native ATM multicast but they present some drawbacks, especially in terms of overhead, management, and scalability. This paper presents a new native ATM multicast proposal that solves some of the problems found in other mechanisms. The multicast group is associated to a group of adjacent VCs (compound VC). A unique entry in the switching table is needed to switch the traffic of the entire group by means of a mask. The variable part of the VCI is dynamically changed each time the first cell of a new PDU arrives to a switch.

Keywords:
AAL5 Cell-interleaving, ATM Multicast, Cell Forwarding

1. Introduction

New kinds of traffic generate new needs for services that must be fulfilled by the network. The growth in importance of multimedia traffic is one example of this. Apart from the well-known requirements of quality of service, maybe the most important challenge is to allow a user in a group to communicate with the rest of the group in an efficient way. Efficiency is measured in terms of resource consumption in the links (e.g. bandwidth) or at intermediate nodes and end-systems (e.g. memory, processing). This communication scheme is known as multicasting.

Multicasting can be carried out at different levels; e.g. one application could create a point to point communication to each of the members of a group. In this case, the same data are duplicated and sent over the same link unnecessarily. As a consequence, there is a waste of resources, which increases with the size of the group.

A more efficient way to do multicasting is at the routing level or switching level, because in these cases, merging points and splitting points can be managed inside the network. Consequently, the sending application forwards just one copy of the transmitted data and a distribution tree is built with just one copy of the information being forwarded through each branch of the shared tree.

The requirements for a generic multicast mechanism as stated in [1] can be summarized in six main points: multicast group address assignment, group set-up, membership maintenance, group tear-down, error recovery, and flow control. [2] appends packet forwarding to the list of main requirements for multipoint communications in ATM networks.

The inherent connection-oriented nature of ATM introduces new challenges to the multicasting problem with respect to normal IP multicasting over broadcast media. ATM is a Non Broadcast Multiple Access (NBMA) technology. Thus, multicast mechanisms cannot benefit from inherent broadcast facilities offered by a broadcast medium like Ethernet whose characteristics are exploited by connectionless multicast protocols like IP multicast.

ATM introduces further challenges to the multicasting problem due to its inherent QoS provisioning and its connection-oriented nature. When establishing group communications, the routes to the members must be computed according to a requested QoS. Therefore the signaling and routing protocol responsible for that (e.g. PNNI) will be more complex than those found in IP multicast networks, which are based on best-effort service. Furthermore, QoS should be enforced during connection

* This work has been funded by the Spanish Ministry of Education (CICYT) under grant number TEL97-1054-C03-03.

establishment (CAC) and during data forwarding (UPC). The problem is further complicated due to the heterogeneous and dynamic nature of groups. All these characteristics make congestion and flow control mechanisms more complex.

When offering multicast service over ATM, there are two main options depending on whether current infrastructures and technologies are used or not. In the first case, interoperability with present equipment is possible without much effort, but the price paid is extra overhead added and consequently less efficiency. A solution of the first type is IP multicasting over ATM, proposed in [3], where the adaptation of the connectionless nature of IP over the connection-oriented nature of ATM is carried out by and address resolution server and a Multicast Server (MCS). The MCS is used to distribute all the multicast data to the group members through point to multipoint VCs. When a member wants to transmit a packet, it is sent to the MCS by means of a unicast connection and the MCS distributes it. If AAL5 is used, the MCS must reassemble all cells belonging to the same incoming packet (AAL5 CPCS-PDU). Cells belonging to the same packet must be transmitted together through the point-multipoint VC in order to avoid cell interleaving.

A more efficient option is to provide multicasting mechanisms at the ATM level, that is Native ATM Multicast mechanisms. This alternative implies some modifications in the design of current switching equipment.

Though ATM is cell-based, one expects the information in higher layers to be generated as packets. Moreover, AAL5 seems to have been widely accepted and used for transmitting most of data and multimedia communications over ATM. But AAL5 has no fields in its SAR-PDU allowing the multiplexing of cells belonging to different CPCS-PDUs, like the MID in AAL3/4. Therefore, when designing the forwarding part of the multicasting mechanism, the cell-interleaving problem must be solved in order to allow the receivers to reassemble the original packet.

The focus of this paper is on describing a new forwarding mechanism that allows native ATM multicasting when all end-systems in a group use AAL5. Thus, our goal is to solve the cell-interleaving problem.

Before introducing our proposal, a brief review of the previous proposals of native ATM multicasting is presented so as to study what are the unsolved problems to be tackled by our mechanism.

This paper is organized as follows: First, a review of the previous proposals of native ATM multicasting is presented. The following section describes the CVC mechanism in detail. The motivation, operation, and characteristics of the mechanism are explained. Next, a comparison of native multicast mechanisms is presented. Some simulation results are discussed in the following sections along with a comparison with some previous mechanisms. Interoperability and deployment scenarios are also discussed. And finally, the conclusions and future work are explained.

2. Review of Native ATM Multicast Proposals

In the context of this paper, native ATM multicast refers to the mechanisms implemented at the switches to allow the correct ATM level forwarding of the information being interchanged by the members of a group. That is, the cell-interleaving problem is solved without having to reassemble cells into AAL5 CPCS-PDUs inside the network, as it is performed when a Multicast Server (MCS) is used [3].

A classification of these mechanisms can be found in [4], where they are referred to as VC merging techniques. However, we don’t use this notation. Throughout this paper VC merging is considered as a particular case of native ATM (see table 1 below). Furthermore, we understand some of the mechanisms in a different way.

In that paper, three main types are identified. The first type of techniques solves the cell-interleaving problem by avoiding cells from different packets to be interleaved. Simple and Efficient ATM Multicast (SEAM) [5] is the most representative of them. It buffers cells of a packet until no other cells are being forwarded to the same output VC. This buffering, when carried out at each switch in the path, presents the additional effect of increasing burstiness, latency and CDV and thus, the traffic characteristics may be violated. SEAM implements cut-through forwarding. That is, the forwarding of a PDU starts when the first cell of the PDU arrives if the outgoing buffer is empty and lasts until the last cell of the PDU arrives. Therefore, a slow source could block the rest of sources for remarkable time intervals. The MultiProtocol Label Switching (MPLS) proposal [6] follows the same idea, but in this case, the first cell of a packet is not forwarded until all the cells of a packet have been buffered. We will jointly refer to SEAM and MPLS as VC Merging techniques.

Though these techniques may present an easier implementation when compared to the other types, their main drawback is the buffering requirements at input queues of the switch. This buffering, when carried out at each switch in the path, presents the additional effect of increasing burstiness, latency and CDV and thus, the traffic contract may be violated. As a consequence, their main application would be data transmissions, and not real-time transmissions.

An extension of VC Merging techniques for providing some kind of quality of service (QoS) classes is explained in [7]. In that paper, different output buffers are used for traffic requiring different QoS. Each output buffer is assigned a different VC and a cell level scheduling mechanism is in charge of interleaving cells of different classes so as to minimize traffic distortion. But problems derived from buffering will remain for traffic belonging to the same class. Therefore, if the class granularity is not very small, i.e. if there are not many different buffers, one could expect that all the multimedia traffic will pass through the same buffer and thus, it will have its traffic parameters distorted. On the other hand, if a lot of classes are defined to allow cell interleaving between classes, VC consumption will increase. Therefore, the scalability advantage claimed by VC Merging over other strategies diminishes.

We also think that the first type should be further divided into two subclasses to consider the mechanisms in which cell-interleaving is avoided by means of a token passing protocol that allows only one sender to put information in the shared tree at any given time, like SMART [8]. In this case, the shared tree is accessed as if it was a shared medium. This mechanism allows the enforcement and accomplishment of the traffic contract because enforcing the contract of the group at any time corresponds to the enforcement of the sending end-system. But it is a complex protocol because all switches must interchange RM cells in order to allow the token to go from sender to sender, which imposes a remarkable overhead. The mechanism is further complicated if more than one tree at a time should be managed to allow some senders to send to the group at the same time.

The second type corresponds to VP switching techniques, where the VPI identifies the connection and the VCI is used as the multiplexing ID. A further subdivision differentiates between VCI identifying just the packet, or VCI identifying the source. DIDA [9] follows the former scheme, while [10] proposes a modification of the latter, named VP-VC switching, which tries to combine the advantages of what the authors call VP switching and VC switching. VP switching in [10] corresponds to a type 2 strategy with globally unique VCI identifying the sender. And, what the authors call VC switching corresponds to a type 2 strategy with the VCI value being changed at each switch. Therefore, additional mechanisms are required to identify the sender. The VCI mapping is static and once established it lasts until there are no more cells coming with a given input VCI.

Though these strategies could be implemented with small or no modifications to the current switches, their main drawback is scalability in terms of the number of groups that can be established, as the VPI field just has 8 bits at the UNI, or 12 at the NNI. Furthermore, network operators could make use of the VPI, limiting the use an end-system could make of it.

The third type groups those techniques that propose a modification of AAL5, and particularly its SAR-PDU, by adding an extra field that carries multiplexing information for each cell. And again, this field could be used to identify the packet or the sender. In this latter case, a global ID assignment is needed. SPAM [11] is an example of these techniques which uses per packet IDs. These techniques make use of an overhead that reduces the bandwidth available to user information on a given link. Furthermore, with SPAM, the switch needs to do AAL processing by looking at the multiplexing ID in the SAR-PDU, a task that, in principle, corresponds to the end-system.

We think it would be better to make a more generic definition of the third type by naming it Multiplexing Overhead Field so as to include Cell Re-labeling At Merge Points (CRAM) [12] as one particular case. This mechanism carries the multiplexing information not in the SAR-PDU but in a RM cell which precedes a block of interleaved cells. Therefore, we think CRAM would belong to the third type and not to the first, as classified in [4].

Table 1. Classification of Native ATM Multicasting mechanisms

Type	Subclass	Examples
Avoid Cell-interleaving	VC Merging (Buffering)	SEAM MPLS
	Token control	SMART
VP switching	Source ID	Standard VP switching VP-VC switching
	Packet ID	DIDA
Compound VC switching		CVC
Allow Multiplexing inside a VC	Added overhead	SPAM CRAM
	GFC	Subchannel (WUGS)

These techniques make use of an overhead that reduces the bandwidth available to user information on a given link. Furthermore, with SPAM, the switch needs to do AAL processing by looking at the multiplexing ID in the SAR-PDU, a task that, in principle, corresponds to the end-system. In the case of CRAM, RM cells should be processed in the switch. This processing should be added to the table look-up operations. Moreover, the processes of analyzing and creating a block consisting of the RM cell followed by the cells indexed by it, needs some buffering and it could affect latency, CDV, and burstiness, though this effect is not as harmful as in SEAM.

Another mechanism that cannot be classified within one of the three types above is explained in [13]. Further modifications to the classification in [4] must be considered. The multiplexing identifier, named subchannel identifier, is carried in the GFC field of the ATM header. That gives the possibility of fifteen simultaneous packets in a switch (one ID is left to indicate an idle subchannel). It identifies a burst or packet formed of data cells that are transmitted with RM cells at both ends. The subchannel ID is dynamically changed at each switch for each packet. With this strategy, there is no extra overhead due to multiplexing information, but there is some due to RM cells, though it could be avoided if no reliability concerns are imposed and the end-of-packet indication of AAL5 is

used to differentiate the packets. This mechanism is implemented in the Washington University GigaSwitch (WUGS).
Turner also proposes the use of more than one VC if fifteen IDs are not enough to absorb all the traffic. And establishing a block of VCs (one per subchannel ID) that are shared by all senders solves interoperability with non-subchannel switches.
One of the main problems of this mechanism is the utilization of the GFC field. Therefore, it won't be available at the UNI for user access, e.g. in a passive optical bus, or other purposes. Moreover, it could limit the potential widespread use in NNI interfaces, as there is no GFC in those interfaces. And, when considering the proposed interoperability mechanism, the establishment of fifteen new VCs for each VC allowing subchanneling, implies a lot of sudden extra processing, which is especially critical at the end-system.
Thus, table 1 presents the classification as we think it should be with some new mechanisms that didn't appear in [4].
A new type was added to the table so as to consider our mechanism. It is between types 2 and 3 because it shares some characteristics with both types. Like in DIDA, VCI is used to identify an AAL5-PDU. Like in SPAM and the like, multiplexing is allowed in the CVC connection.

3. Compound VC Mechanism (CVC)

3.1. Motivation

The main goal of the mechanism proposed in this paper is to solve some of the problems of the mechanisms that have been commented above, namely buffer requirements, alteration of traffic characteristics, overhead, and flexibility. First, it is presented for a local environment. Further studies will be carried out to state the potential application of the mechanism in a wider scale, either by studying soft transition scenarios or by proposing interoperability mechanisms.
With that final goal in mind, scalability should be a major concern, mainly in terms of the number of groups that the mechanism can handle. Moreover, allowing heterogeneity in the size of groups should also be a key aspect for every multicasting strategy. For instance, in DIDA all the VCIs of a VP are assigned to a group even if the group is small, because the VP identifies the connection. Thus, having 2^{16} identifiers in a group is the smallest granularity. A similar problem arises in SPAM, CRAM or in Turner's, where multiplexing identifiers have fixed-length, though in these cases the VCI (not the VPI) identifies the connection. With CVC, there could be many small groups in the same VP and the length of the identifiers could be adjusted to the traffic generated in the group.
However, in particular cases, having one of such mechanisms may be appropriate according to traffic constraints. Therefore, our mechanism is as generic as possible, so as to allow the negotiation at connection establishment of the number of identifiers according to traffic characteristics. Thus, some of the mechanisms that have been discussed above are particular cases of CVC. Furthermore, point-to-point connections can also be switched by using the same table as that for multicast groups. Point-to-point connections are also a particular case of our mechanism.
But solving scalability must not limit the potential application of a mechanism to multimedia, which is growing in importance. Multimedia traffic imposes stringent QoS constraints that make mechanisms that interleave cells more suitable for two reasons. First, the traffic contract must be respected. For instance, in the case of VC Merging, buffering at the switches could violate the bounds established in the contract of a multimedia communication. And secondly, multimedia traffic is often associated to group interaction, e.g. a multipoint videoconference and collaborative work applications. While respecting QoS is important, so is the sharing of resources assigned to a group so as to allow simultaneous transmission from different sites. In the case of SMART, though it proposes a solution for that scenario, the management and overhead involved are too heavy. CVC doesn't share this approach.

3.2. Previous assumptions

CVC focuses on forwarding, that is, the way information is distributed from a sender to the rest of the members of a group through a shared multicast tree. All the information sent to the group will use the same tree. The adaptation layer is AAL5. Thus, the cell-interleaving problem is the major concern of this paper.
Other important problems like signaling and routing are not considered in this paper. The discussion on signaling and membership management would be similar to that of [5], with some variations to consider the way CVC connections work. For instance, some especial kind of traffic negotiation is required to calculate the number of multiplexing IDs needed to be able to handle the traffic that senders generate. And the information interchanged during signaling would be slightly different to consider the nature of CVC, e.g. to handle a group of VCs as if it was a whole, and new information elements like masks, as explained below.
Some assumptions about routing are also required. VC mesh is not considered for CVC because of the high signaling overhead during establishment and during group management. Though it presents important advantages when compared to the shared-tree approach, like load sharing between switches and shorter delays, drawbacks outweigh advantages. Furthermore, all the latest mechanisms consider shared-tree routing. Thus, CVC assumes the existence of a routing protocol capable of establishing multipoint to multipoint shared trees with some QoS constraints.

3.3. CVC operation

CVC is somewhat related with some of the mechanisms that appear in the literature, like DIDA or VP switching with source ID, but its aim is to be as generic as possible in the sense of integrating these mechanisms and providing new features. At the same time, it tries to solve some of their problems, especially scalability in terms of number of groups.

Like in DIDA, there is a multiplexing ID per PDU that is carried by each cell belonging to that packet. CVC is based on local remapping of multiplexing IDs at the switches. Thus, no global ID assignment mechanism is needed, which would limit its potential deployment in wider environments than the local area. A further advantage when compared to mechanisms that identify senders is that more traffic can be switched with a reduced set of identifiers, as IDs are dynamically assigned to the PDUs generated by the senders. A drawback of this approach is that the sender ID must be carried in the payload of the AAL5-PDU, e.g. in the form of the source address of the IP packet.

VP switching techniques use the VPI to identify the group connection. This limits scalability and potential use of the VPI by the operator. The solution CVC proposes is to treat a group of VCs as a whole when switching cells. The number of VCs in this group could be a multiple of 2 between one and 2^{16}. Thus, by providing this flexibility, scalability is increased and one major problem of VP switching techniques is overcome.

Figure 1 presents the block diagram of an ATM switch implementing CVC forwarding. The main difference with a normal switch is the new block labeled ‘Dynamic VCI Assignment (DVA)’. This block is in charge of assigning a free ID to a new incoming PDU that will be carried by the variable part of the VCI field of a CVC connection.

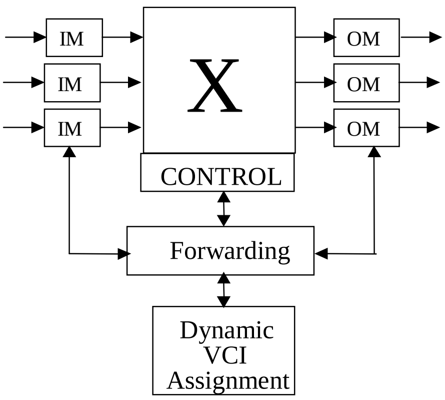

Figure 1. CVC switch architecture

When the input module (IM) detects the first cell of an incoming PDU, the forwarding module examines the values required to carry out the forwarding. The forwarding mechanism consists of a table lookup operation to map an input port, input VPI, and input VCI to an output port, output VPI, and output VCI. The difference when applying CVC is the use of the mask to carry out the mapping of the VCI field. The fixed part of the value of the VCI field for a given CVC connection is mapped as in a normal switch by means a similar table, but the variable part is assigned according to a table of free IDs that is maintained by the DVA block. Once both parts of the VCI field have been assigned, the control block will switch the cell to its destination through the switch fabric and the OM will give it the right output values.

When the IM detects the last cell of a PDU, the DVA frees the ID assigned to this PDU as soon as the cell is sent to the output buffer.

Therefore, a new column in the switching table is required to consider a mask, but table size is not globally increased, as there is just one entry for the whole group. The mask determines the portion of the VCI that will identify the group of VCs and the portion that will contain the multiplexing IDs. The idea of using a mask to group these VCs and to switch them together is similar to that of IP subnetting mechanisms. A comparison between both types of masks –IP and CVC- shows the following parallelism: 1) IP masks reduce the size of the routing table, and CVC masks reduce the size of the switching table when compared to those of VP switching techniques; 2) IP masks allow a lot of smaller subnets where there was a big but inflexible network, and CVC masks allow a lot of smaller groups with diverse sizes where there was just one group connection inside a VPI.

The structure of a table for output port 3 of switch S3 in figure 2 would be of the form presented in table 1.

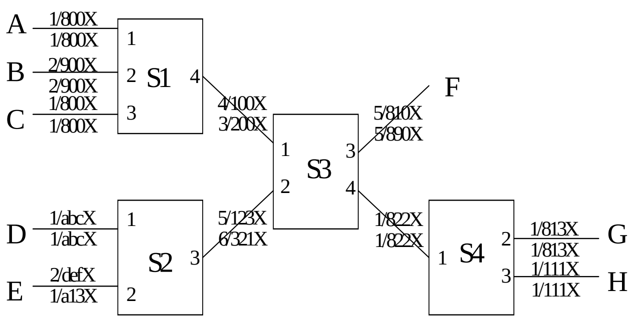

Figure 2. Example of topology for a CVC connection

Table 1. Example of switching table for output port 3 of switch S3 (X=not considered)

Mask	IN			OUT		
	Port	VP	VC	Port	VP	VC
FFF0	1	4	100X	3	5	810X
FFF0	2	5	123X	3	5	810X
FFF0	4	1	822X	3	5	810X

A group composed of eight members (identified by letters A thru H) is presented in this figure. The four switches are labeled S1 to S4. The numbers inside the box of the switch identify the port. The number in the upper part of each link corresponds to the identifiers for a flow traveling left to right and that in the lower part corresponds to traffic flowing in the other sense. The notation used is VPI/VCI. During connection establishment, the length of the multiplexing identifier is negotiated to be 4 bits according to traffic characteristics. Thus, ID corresponds to the last hexadecimal digit of the VCI.

The mask FFF0 identifies the portion of the VCI used to handle the entire group, in this case, 12 bits. The rest (4 bits) is the part assigned to multiplexing IDs, which are not important for this table. In the example, three entries are enough to handle the traffic generated by a group of 8 senders (A thru H) sharing 16 IDs.

It can also be deduced from the table that DIDA is a particular case of CVC that uses a mask of 0000, that is, the entire VCI identifies the packet. The mechanism proposed in [13] for interoperability corresponds to a CVC connection with a mask of FFF0 (it uses sixteen simultaneous VCs). Therefore, our mechanism is generic and flexible in the sense that it can benefit from the advantages that some previous mechanisms present in particular situations. At the same time, it presents new features for cases where those mechanisms are not adequate.

Point-to-point connections are also a particular case of CVC that uses a mask of FFFF, allowing a single table to handle all the traffic, be it multicast or unicast. But with mixed multicast and unicast connections, the gain in table size is not as important as with only multicast connections.

Maintaining a list of free IDs for each group solves collision of IDs. They are selected and marked as occupied when the first cell of a PDU arrives. When the cell with the end-of-packet indication arrives, the ID returns to the list of free IDs.

3.4. Potential Drawbacks of CVC

It could be argued against CVC that there is a large memory requirement in the switches to implement the remapping of multiplexing IDs. But these requirements are compensated by the utilization of the masks, which summarize all switching information for one group in a few entries, as seen in the example. Mechanisms that do local mapping of IDs do also need such mapping tables. The main difference is that the management of variable tables in CVC could be a little more complex.

The utilization of VCI to carry multiplexing IDs could reduce the number of available VCs, thus, limiting its scalability. But this is not a problem in the local area because IDs will never be exhausted.

When considering a wide area environment, the rest of the mechanisms also present major problems (e.g. scalability) that must be overcome before their deployment. VC consumption could limit the scalability of CVC, for this reason we focus on the local area.

But the scalability problem in CVC is not as critical as in VP switching techniques, because CVC adds the flexibility of dynamically assigning the size of the ID field. Consequently, it supports more simultaneous groups; i.e. it is more scalable. Furthermore, when compared to VC Merging techniques, which are claimed to be scalable, other considerations must be pointed out. These techniques are frame-oriented, that is, they avoid cell interleaving between PDUs by forwarding all the cells of the same PDU together, which modifies the traffic characteristics. The goal of CVC is to be applicable to multimedia communications. Consequently, the traffic characteristics must be respected. In conclusion, the scalability gain of VC Merging is obtained at the cost of limiting its applicability to multimedia. The solution in [7], and commented above, to provide QoS to VC Merging proposes the use of an output buffer and VC for each class of service. That could make VC usage in VC Merging similar to that of CVC, thus, the same scalability concerns should be tackled for both mechanisms. Therefore, the conclusion that can be drawn from this discussion is that QoS is obtained at the price of using more IDs, no matter the strategy used.

4. Comparison of the behavior of native multicast mechanisms

Figure 3 presents an example of the behavior of the different mechanisms that were studied. In the example, there are four incoming PDUs (A thru D). The upper part – labeled as incoming cells- represents the instants at which the cells belonging to each PDU arrive to the switch. For instance, PDU A is composed of 4 cells arriving with a timing of one every four time units.

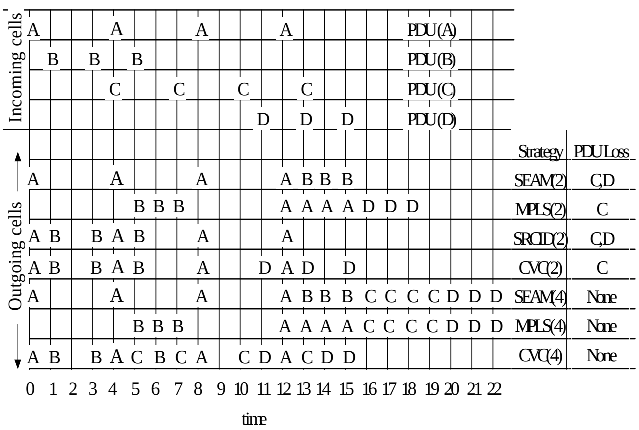

Figure 3. Comparison of Native ATM Multicasting mechanisms

The number in parenthesis is the number of IDs or of reassembly buffers depending on the strategy

SRCID corresponds to ID assignment per source

The part of the figure labeled as outgoing cells presents the exit instants of the cells above for each mechanism after having passed through processing in the switch. Each horizontal line represents the exit instants for the mechanism specified in column ‘Strategy’. The number in parenthesis represents the number of IDs (for CVC, SRCID) or the number of reassembly buffers (for VC Merging techniques). The column labeled as ‘PDU Loss’

gives the PDU losses when applying each of the mechanisms.
Some assumptions have been made: 1) If the first cell of a PDU is discarded, so are the remaining cells, like in Early Packet Discarding strategies. 2) Only for comparison purposes, we have taken 2 and 4 reassembly buffers in VC Merging techniques, because we took 2 IDs for the other strategies. It does not claim to be a general case in practice. 3) The initial state of the system (time=0) corresponds to empty buffers. 4) In practice the arrival instant and the instant at which the cell exits the switch is not the same due to processing time at the switch. However, we didn't represent this time difference to make the figure simpler to understand.
We will focus our discussion on the strategies using 2 IDs or 2 reassembly buffers. The results for Source ID strategies (SRCID) show the inefficiency of this kind of mechanisms, because all PDUs sent by sources that couldn't get an ID are lost. The solution adopted by these mechanisms is to dimension the field containing the ID so as to consider the maximum possible number of sources. Therefore, the size of the field is fixed to a length that could consider the biggest expected group, e.g. 16 bits (2^16 senders). But for small groups, as in a local area videoconferencing, a lot of resources are wasted, because most bits are not used. Thus, these mechanisms are not flexible.
Another limitation that shows up when examining the figure is that of SEAM applying a cut-through strategy. When the first cell of the slowest PDU arrives, no PDU is being forwarded, therefore it starts to transmit and blocks the rest of sources for 12 time units, while there are time intervals when nothing is transmitted though there are complete PDUs waiting for in the buffers to be transmitted. Furthermore, another important drawback that limits the application of SEAM -and all VC Merging techniques in general- for multimedia applications is that the traffic characteristics are modified at each switch. The traffic becomes more bursty due to buffering. Consequently, cells are not transmitted at the rate the source generated them but at the peak rate negotiated for the VC.
MPLS does not present the first drawback described for SEAM, because no cell of a PDU is transmitted until the last one arrives. Thus, PDU B can be transmitted while A is being buffered. In that way a reassembly buffer is freed and can be used by other PDUs (D). But the same traffic distortion occurs. Thus, it may cause problems with multimedia traffic. Furthermore, VC Merging techniques require more buffering than the rest of the mechanisms.
The results for CVC(2) show the same throughput as MPLS but the traffic characteristics of individual sources are maintained. Therefore, there is no limitation in the application of CVC to multimedia traffic, which makes it interesting as this traffic presents high resource consumption.
The comparison of CVC(4), SEAM(4), and MPLS(4) shows that CVC provides a way to efficiently use resources while the traffic characteristics are being maintained (apart from the CDV introduced when two cells arrive at the same time, e.g. when time=4). For VC Merging techniques, the throughput is the same as for CVC(4) but a lot of delay and CDV variation is introduced. For instance, the first cell in PDU C enters the switch at time=4 and exits at time=16, with the rest of the cells of C being sent at the peak rate of the VC.
The price paid by CVC is a higher VC (or IDs) consumption than VC Merging techniques, but this is not a major problem for a local environment in which we are planing to use it.
But VC consumption is not that high with respect to VC Merging if these latter techniques are modified as in [7] to offer some kind of quality of service (QoS) as commented above. If we want to attain the QoS granularity as in CVC, we should use many VCs, and VC Merging techniques will end up by having the same scalability concerns as CVC due to VC consumption. Furthermore, QoS is not respected as in CVC because reassembly buffers remain, and that will introduce some delay inherent to VC Merging that is not present in CVC. In normal scenarios data will pass through many switches, and the delay and CDV will accumulate.

5. Simulation Environment

The simulated scenario consists of some sources sending traffic to the same switch. The sources are homogeneous, i.e. they have the same statistic (geometric) for the arrival process with the same mean interarrival time. The length of the PDUs also follows a geometric distribution for all sources. An ON-OFF model models each source. Once in the ON state, the source transmits cells at its PCR. The number of sources is varied to compare the behavior of the mechanisms with different input traffic loads. For the results presented in next section, the output PCR for the group doubles that of each source.
The switch implements three multicasting strategies – namely MPLS, SRCID, and CVC. An important parameter characterizing the switch is the number of identifiers (or reassembly buffers for MPLS).
The VC Merging discipline implemented is that of MPLS. That is, we don't consider cut-through forwarding as in SEAM. MPLS does not use ID assignment as there is no cell interleaving. When comparing strategies, the number of identifiers is matched to the number of reassembly buffers in the switch. Though this comparison does not strictly represent a general case, we did it when comparing strategies for two reasons. First, finding a free reassembly buffer could be assimilated to finding a free ID. And secondly, taking the same number of reassembly buffers in MPLS as IDs in the other mechanisms is a good choice for comparison purposes. Though in a real case the number of reassembly buffers will generally exceed that of IDs for a given CVC connection, these buffers are shared by all the traffic going through a given output port. Therefore, there is interference between multicast groups. On the other hand, CVC uses less IDs but they are not shared. As we just simulate traffic belonging to the same multicast group,

we think we can compare these mechanisms if the above assumptions are made.
In the case of SRCID, it can only give service to a number of sources not greater than the number of IDs. Therefore, we assumed that, at the beginning of the simulation, each ID was assigned to a source and this binding lasted until the end of the simulation.
We also assumed an infinite output buffer in the switch, so as the only losses are due to ID (or reassembly buffer) exhaustion.

6. Results

The number of sources in our simulation ranged from 1 to 10, with a PCR≈15Mbps and a SCR≈4Mbps each. The PCR of the output link was 30Mbps and the mean PDU length 5 cells.

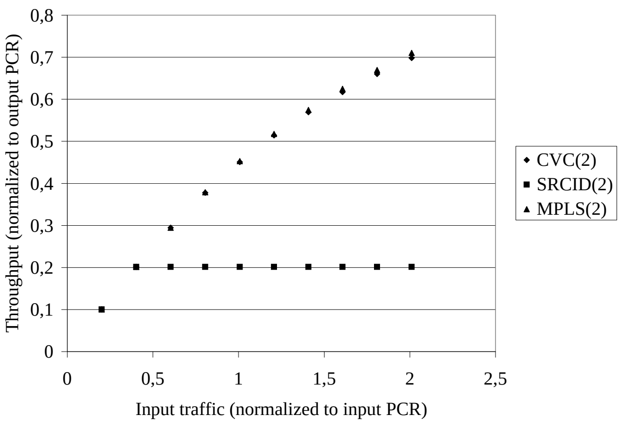

Figure 4. Comparison of SRCID(2),CVC(2), and MPLS(2)

Figure 4 presents the comparison of the throughput obtained for the three mechanisms with 2 IDs (or reassembly buffers). It shows what could be expected after examining the behavior of the mechanisms described in figure 3. The results for the three methods show no significant differences with aggregated input loads lower than 40% of the PCR of one source. But these low loads will not be very common even for local environments as one of the main applications of multicast is multimedia communications. Such communications are characterized by high bandwidth consumption, essentially due to video transmission.
For SRCID, the throughput is limited to the traffic generated by the sources that allocated an ID at the beginning of the simulation. That is, once the number of sources reaches the number of available IDs for a group, the throughput stops increasing and the traffic from the rest of the sources is discarded, as there are no free IDs. That is the reason why this mechanism presents a constant throughput for input loads greater than 0.4, which is the load imposed by 2 sources. Usually, the mechanisms using this philosophy overcome this drawback by overdimensioning the number of available IDs, but this comes at the price of extra overhead that will never be fully used as it is overdimensioned to consider the worst case, that is, the biggest possible group. That is the case for 'VP switching with Source ID' which uses a source ID size of 16 bits.
The throughput obtained for MPLS and CVC presented no significant differences, which can also be deduced from figure 3. But, in MPLS the accumulated delay and CDV after passing through some switches could make MPLS impractical for multimedia communications. There are two main causes for such delay and CDV: First, the reassembly of all cells of a PDU. And secondly, the multiplexing with PDUs going out through the same output port. This PDUs may belong to the same multicast group or not. That is, the traffic from all the connections passing through a given port affect each other as they share the same reassembly buffers.

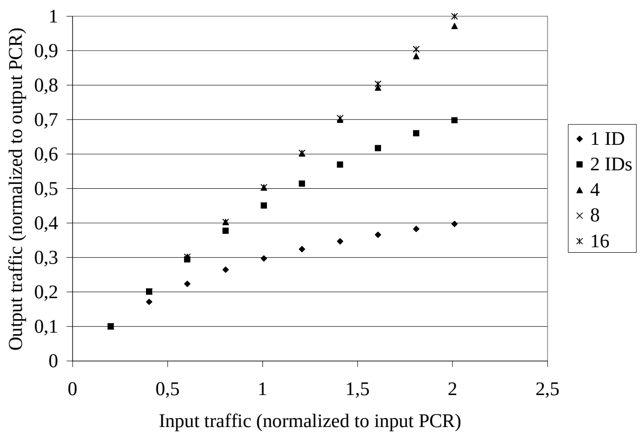

Figure 5. Throughput for CVC with different number of IDs

A comparison of the throughput obtained with different number of IDs with CVC is presented in figure 5. It can be observed that with few IDs the throughput obtained is high even with a number of IDs much lower than the number of sources. Even with 1 ID, the throughput obtained with a global input load of 2.0 is remarkable (0.4). When the number of IDs is increased, there is a substantial gain in the behavior of CVC, because throughputs up to 0.7 are obtained for 2 IDs. And with 4 IDs, the behavior is very close to the ideal case. The throughput obtained in this case for 10 sources is 0.97. No difference between the 8 IDs case and the ideal case is shown. That is, for this traffic, with just 4 IDs the traffic characteristics are respected and the behavior is approximately that of the ideal case.
Therefore, CVC presents good throughput characteristics, the same as MPLS, and at the same time the traffic characteristics are respected as cell-interleaving of cells belonging to the same group is allowed. Furthermore, the buffer requirements are lower than those of MPLS because no reassembly buffers are required. Moreover, cell-interleaving is carried out in a flexible way, unlike in SRCID mechanisms, as the number of IDs assigned to a

group is chosen according to group characteristics at connection establishment.

7. Interoperability

A typical scenario to study interoperability would be that composed of islands of CVC switches and islands of current equipment. The most reasonable solution in this case seems to be that proposed by Turner [13] where the egress switch of the CVC island establishes as many VCs as multiplexing IDs are assigned to the group (N). Conventional signaling will be used to establish these circuits, but the egress switch will schedule the traffic going through each VC, so as to share all the VCs among all the senders. The end-system should signal N VCs. Further work must be carried out to determine whether the cost of establishment and processing overhead of one VC per class (as in [7]) is smaller than establishing and managing a CVC connection.

Another option to reduce the signaling overhead at end-systems consists of implementing VC merging at the egress switch. This would allow the end-system to receive all the traffic in the group by establishing just one or very little number of VCs. On the other hand, the switch would be more complex and the delay, CDV, and burstiness could be slightly increased. But as this operation is just carried out at one switch, its effect could be acceptable.

Ingress switches (in a CVC island or for those in contact with a common end-system) should be in charge of transforming conventional VCs into CVC connections by maintaining a special table that would map a VC to a group connection. Mapping would assign a free ID to each PDU just as it does in normal CVC operation. A previous setup phase of the CVC connection is required from the ingress switch to the rest of the switches.

Another potential application of CVC to current equipment would be at Multicast Servers (MCS). To avoid the burden of the reassembly processing at the MCS, it could be slightly modified to provide some kind of CVC to allow cell multiplexing at the point-multipoint VC. The unicast connections between end-systems and the MCS need not be modified. There are two options for the point-multipoint VC. The first is to use a CVC connection. That would require all switches in the network to implement CVC. A simpler approach that would not require modifications in the network would be to apply a scheme such as the interoperability mechanism described in the last section to the point to multipoint VC. That is, N point-multipoint IDs should be established. The signaling burden could become very important, but in the local environment in which MCS is used, it could be acceptable.

8. Deployment scenarios

As it preserves the traffic characteristics, CVC is suited to offer service to multimedia communications. Some of the characteristics of CVC that support this point are the flexibility offered by CVC in negotiating the size of the multiplexing ID, and the assignment of this ID to a PDU and not to a source. The former aspect allows a flexible dimensioning of groups. Therefore there could be many simultaneous groups of different sizes inside the same VP, because the VCI space is partitioned by means of a mask. The latter, allows high resource utilization with very few IDs as opposed to Source ID mechanisms.

The scalability problem could be solved in cases where traffic is mainly local. For instance, we could consider the common case of an enterprise with some sites –each with its local network– connected to other sites through a backbone. In this scenario one should assume that most part of the traffic is locally transferred, and a small part is transferred to the rest of the sites through a backbone. In this case, a smaller number of IDs than those required for the local area could be selected during CVC call setup. Thus, the CVC connection for the ports that communicate the sites through the backbone would multiplex less traffic.

9. Summary, Conclusions and Future Work

In summary, CVC is a generic mechanism that has some of the mechanisms exposed above as particular cases, but that allows more flexibility in terms of group size and number of groups allowed. Finally, as it is a mechanism allowing cell multiplexing, it presents a good trade-off between the possibility of enforcing the traffic contract and resource sharing among senders.

The characteristics of the CVC mechanism have been discussed by comparing its behavior with Source ID and VC Merging mechanisms by means of an example and through simulation. CVC allows more traffic to pass through the CVC connection assigned to the group when compared to Source ID mechanisms for the same number of IDs. The solution implemented by Source ID mechanisms of overdimensioning the ID field supposes a waste of resources. CVC allows a more flexible dimensioning of the field ID according to the size of the group by means of a masking mechanism.

VC Merging techniques and CVC present no significant differences in terms of throughput, but the traffic characteristics are modified in the former and not in the latter. That makes CVC suitable for multimedia applications with stringent QoS requirements and generally discards the application of VC Merging techniques.

Requirements of CVC for interoperating with existing infrastructure will be further developed in the future. Interoperability will serve us to introduce possible transition scenarios.

The relationship between the traffic characteristics and the number of IDs required will also be studied. A detailed description of the internal operation of a CVC switch will also be considered.

At present, there is a lack of mechanisms that could be used in the wide area because of scalability problems. The adaptation of CVC to this scenario will be our final goal.

And at last but not least, traffic and group management should be a major concern. There is the need to propose a procedure to determine the number of IDs assigned to a group given some traffic constraints. The criteria used to enforce the traffic in the group should also be stated, as well as flow control considerations. And finally, the modifications to allow dynamic integration of new users to the group will be studied.

10. References

[1] : Braudes R and Zabele S. ‘Requirements for Multicast Protocols.’ IETF RFC 1458. May 1993.

[2] : Fahmy S, Jain R, Kalyanaraman S, et al. ‘A Survey of Protocols and Open Issues in ATM Multipoint Communications.’ http://www.cis.ohio-state.edu/~jain/papers/mcast.htm

[3] : Armitage, GJ ‘IP multicasting over ATM Networks.’ IEEE Journal on Selected Areas in Communications 15(3): 445-457, April 1997.

[4] : Baldi M, Bergamasco D, Gai S, and Malagrinò D. ‘A Comparison of ATM Stream Merging Techniques.’ Proceedings of IFIP High Performance Networking (HPN’98): 212-227, Viena, September 1998.

[5] : Grossglauser M and Ramakrishnan KK. ‘SEAM: Scalable and Efficient ATM Multicast.’ Proc. of IEEE Infocom’97: 867-875, Kobe (Japan), April 1997.

[6] : Rosen EC, Viswanathan A, and Callon R. ‘Multiprotocol Label Switching Architecture.’ IETF Draft, draft-ietf-mpls-arch-02.txt, July 1998.

[7] : Widjaja I and Elwalid AI. ‘Performance Issues in VC-Merge Capable Switches for IP over ATM Networks.’ Proceedings of INFOCOM’98. San Francisco (USA), pp. 372-380, March 1998.

[8] : Gauthier E, Le Boudec J-Y, and Oeschlin P. ‘SMART: A Many-to-Many Multicast Protocol for ATM.’ IEEE Journal on Selected Areas in Communications 15(3): 458-472, April 1997.

[9] : Calvignac J, Droz P, Baso C, and Dykeman D. ‘Dynamic IDentifier Assignment (DIDA) for Merged ATM Connections.’ ATM Forum/97-0504, July 1997.

[10] : Venkateswaran R, Raghavendra CS, Chen X, and Kumar VP. ‘Support for Multiway Communications in ATM Networks.’ ATM Forum/97-0316, April 1997.

[11] : Komandur S and Mossé D. ‘SPAM: A Data Forwarding Model for Multipoint-to-Multipoint Connection Support in ATM Networks.' Proc. of the 6th International Conference on Computer Communications and Networks (IC3N). Las Vegas, September 1997.

[12] : Komandur S, Crowcroft J, and Mossé D. ‘CRAM: Cell Re-labeling At Merge Points for ATM Multicast.’ Proc. of IEEE International Conference on ATM (ICATM’98), Colmar (France), June 1998.

[13] : Turner J. ‘Extending ATM Networks for Efficient Reliable Multicast.’ Proc. of Workshop on Communication and Architectural Support for Network-Based Parallel Computing, Springer Verlag, February 1997.