Performance Issues of ATM Multicasting based on per-PDU ID Assignment*

Josep Mangues-Bafalluy and Jordi Domingo-Pascual

Polytechnic University of Catalunya (UPC)
Advanced Broadband Communications Center
Computer Architecture Department
Campus Nord. Mòdul D6. Sala 008.
Jordi Girona 1-3. 08034 Barcelona (Spain)
{jmangues | jordi.domingo}@ac.upc.es

Abstract – One of the problems to deal with when offering multicasting at the ATM level is forwarding. When using ATM adaptation layer 5 (AAL5), the cell-interleaving problem must be solved to be able to properly reassemble AAL5 packets at end-systems. Compound VC (CVC) is one of the solutions that have been proposed. Its main characteristic is the utilization of dynamically assigned packet identifiers to allow multiplexing. The convenience of this approach is validated through simulation. This paper also presents some preliminary results on the behavior of the packet loss probability due to running out of identifiers. The curves obtained will help in providing some dimensioning rules for the number of identifiers required for a given multicast group according to its traffic characteristics and a negotiated packet loss probability.

I. Introduction

ATM Forum's User Network Interface (UNI) 4.0 provides mechanisms to establish and manage point-to-point and point-to-multipoint connections. However, it doesn't deal with the multipoint-to-point and multipoint-to-multipoint cases. Some problems arise in these latter cases that make them difficult to solve. One of these problems is the management of the shared-tree, including the dynamic modification of the tree as new senders join or leave the group and the negotiation and renegotiation of the QoS parameters.

Forwarding also presents a major problem when using ATM adaptation layer 5 (AAL5), namely the cell-interleaving problem. Though the basic transmission unit in ATM is the cell, the goal of multicast forwarding mechanisms is the preservation of full packets. This is so because once one cell is lost, the remaining cells of that packet are no longer useful to the end-system. Thus, once one cell is lost, the rest of the packet may be discarded. AAL5 has been adopted in most cases because of its powerful error detection capability and its low overhead when compared to other AALs. But segmentation and reassembly packet data units (SAR-PDU) of AAL5 do not have a multiplexing ID (MID) field like in AAL3/4. Therefore, when cells belonging to different AAL-PDUs get interleaved, the end-system is unable to separate cells belonging to one PDU from the rest. This problem appears at merge points of multipoint-to-point and multipoint-to-multipoint connections. Therefore, when providing the multicast service, a mechanism is required to make end-users able to correctly reassemble AAL5-PDUs.

The IP multicasting over ATM [1] model solves the cell-interleaving problem either by reassembling AAL5-PDUs at a centralized point called multicast server (MCS) or by using point-to-multipoint connections among all the members in the group. However, both options present some major problems if the multicasting service is to be deployed in wide area environments.

On the other hand, the multicasting techniques generically referred to as VC Merging (or Native ATM Multicasting) provide solutions for offering true multipoint-to-multipoint connections by solving the cell-interleaving problem at the ATM level, i.e. without any reassembly inside the network. True multipoint-to-multipoint refers to those group connections using a unique shared tree for all the members in the group.

A. *Classification of Native ATM Multicast Mechanisms*

A classification of native ATM multicast mechanisms is included in [2]. However, VC merging is used there to refer to a particular case of the generic VC merging techniques we refer to when using this term in this paper. Table I presents this classification with other mechanisms recently appeared in the literature and a some notational changes.

Techniques belonging to the first type solve the cell-interleaving problem by avoiding cells from different packets to get interleaved. Buffering techniques reassemble all the cells of each PDU in separate buffers and forward them without mixing cells belonging to different buffers (or PDUs) ([3], [4], [5]). SMART uses a token passing scheme to allow just one sender to put data in the multicast tree at any instant [6]. In the second type, the VPI identifies the connection and the VCI is used as the multiplexing ID (identifying the source [7], or the PDU [8]). In Compound VC Switching techniques, two or more VCIs are used as PDU IDs for the same compound multicast connection ([2], [9]). And the last type allows multiplexing inside the same VC either by adding overhead in the transmitted data ([10], [11]), by using the GFC field in the header of the ATM cell [12], or by negotiating, at connection establishment, the sequence with which cells are going to be transmitted to the downstream node [13].

* This work has been funded by the Spanish Ministry of Education (CICYT) under grant number TEL99-1117-C03-03 (SABA-2)

TABLE I
CLASSIFICATION OF NATIVE ATM MULTICASTING MECHANISMS

Type	Subtype	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	Group ID	FMVC
	Packet ID	DMVC SMVC CVC
Allow Multiplexing inside a VC	Added overhead	SPAM CRAM
	GFC	Subchannel (WUGS)
	Signaling	VC-Merge Scheduler

B. Multiplexing ID vs. Buffering Strategies

The focus of our work is on mechanisms using multiplexing IDs. The goal is to provide real-time multicast connections at the ATM layer. This real-time constraint entails the need for the preservation of the traffic characteristics. Thus, strategies using reassembly buffers, or those not allowing the interleaving of cells from different sources, may limit its application to real-time environments.

This point is further confirmed by the results presented in [14], where comparisons concerning the delay behavior of some mechanisms are carried out. In particular, there is a comparison between store-and-forward VC merge (referred to as hardware VC merge in that paper) and non-hardware VC merge. One example of these latter techniques is VP-merge, and as far as delay in the buffers is concerned, Compound VC (CVC) may also be considered under this classification (The only difference among both may appear due to their different processing at the nodes.) The interesting part of that work, was the utilization of real traffic traces. The results showed that the delay for hardware VC merge was 65% higher across the range of network load until 85%. Standard deviation also was 95% higher in the hardware VC merge case, and this increase stayed approximately constant up to about 85% utilization.

Focusing on multiplexing ID, we find two main groups, namely Source ID and PDU ID. In Source ID mechanisms, the ID is related to the source that transmitted the packet. Therefore, there is a binding between the source and the ID at each switch. This binding must be unique at each switch so as to avoid ID collision at merge points. The ID collision problem may be solved either by globally assigning IDs for the group or by locally remapping the IDs at each switch [7]. The management required in the former option may limit its scalability. On the other hand, local remapping maintains a list of free IDs at each switch, where a local mapping of IDs is carried out.

Source ID mechanisms usually overdimension the size of the ID so as to solve the worst case in which there could be a lot of senders (usually up to 2^{15} or 2^{16}). However, not all groups will have such a huge number of senders, and most overhead will be unused, e.g. in the local area.

PDU ID strategies assign an ID to each packet, and this assignment is independent of the source this packet came from. A new incoming PDU to the switch is assigned an ID from a pool of free IDs. Thus, packets coming from all the sources in the group share the identifiers. In this way, ID consumption is smaller than with Source ID. However, the solutions proposed up to now also use fixed size identifiers except in one case, CVC. DIDA uses a 16-bit field, which is also overdimensioned, even more than in the Source ID case, because these IDs are shared by all the senders. GFC, on the other hand, uses small IDs (the 4 bits of the GFC field), which may be insufficient for bigger groups. In this case, more than one such GFC-connection should be used and the group management is then increased.

The exception comes from Compound VC (CVC), which allows flexible ID size negotiation at connection establishment so as to adapt to the ID consumption required by each group. In this way, the overhead is minimized for two reasons: the PDU ID philosophy and the ID size negotiation. The next section briefly explains how CVC works.

In this paper, we will focus on the behavior of Compound VC Switching, and in particular of Compound VC (CVC). The goal is to study how the number of IDs is chosen depending on some traffic parameters and the number of senders in the group. Some preliminary simulation results are presented for that purpose.

The next section is devoted to explain the working philosophy of Compound VC Switching strategies, and CVC in particular. Following that, some options for CVC signaling are also discussed. Section IV presents the simulation results and a discussion of these results. Finally, section V presents the conclusions and the future work.

II. COMPOUND VC SWITCHING

Compound VC Switching strategies use more than one VCI per multicast connection. In practice, this should not pose any problem because there usually are much less connections than VCIs in a link.

The goal of these strategies is to reduce buffering requirements compared to strategies using reassembly buffers (first type described above). As a consequence, the delay will be lower and a better QoS may be offered to the traffic in the group.

In Dynamic Multiple VC Merge (DMVC) [9], each switch maintains a set of unassigned IDs at each outgoing link pertaining to a given connection. When the first cell arrives, an ID is assigned, and it is maintained for all the cells of the packet. When there are no free IDs, cells are stored until one ID becomes free.

The difference with Compound VC (CVC) [2] is that CVC does not use reassembly buffers. This latter mechanism is oriented to real-time communications, and thus, any buffering in the forwarding path may cause unwanted extra delay. The price paid is an increase in packet losses that make occur in case of not having free IDs to assign when a PDU arrives.

CVC allows the flexible negotiation of the PDU ID size at connection establishment. This PDU ID is carried in the VCI field of the ATM cell header. The VCI field is divided into the compound connection ID and the PDU ID (see Fig. 1). In CVC, all the VCIs assigned to a compound connection are handled together as a single one.

The focus of this paper is on the negotiation process that takes place either when establishing a CVC connection or when renegotiating its parameters. The parameter under study is the number of PDU IDs required for a given connection so as to bound the packet loss probability.

III. SIGNALING

The PDU ID size negotiation will take place at connection establishment. How this size is determined is the concern of the following section. Before entering those aspects, it could be interesting to see the different possibilities for establishing a CVC connection.

The establishment of the connection, with the consequent creation of the group, could be handled in a similar way as that of SEAM [3], which uses core-based trees. Member-initiated joins are supported for scalability reasons. The join procedure could be similar to Leaf Initiated Joins (LIJ) defined in UNI 4.0, not just for receivers, but also for senders.

Core-initiated joins are also considered. This latter approach would be interesting when quick establishment of a group communication initiated by a central coordinator is required.

Apart from all these previous considerations, some distinctive features appear in the signaling of CVC because of having to consider compound VCs instead of normal VCs. Therefore, the signaling at the network to network interface (NNI) must be modified to treat a set of VCs as a group.

For the UNI, there are three options. The first one consists of designing an extension of UNI signaling for CVC. It would consist of modifying current messages by introducing elements that consider the compound VC characteristics as a whole instead of those of each individual VC inside the CVC connection. For instance, there would be a joint traffic descriptor.

Another option would be to establish, by using standard UNI, as much VCs as IDs the CVC connection requires. In this case, a higher level entity would be responsible for managing the information coming from these individually established VCs, and to consider them as a whole. Besides, the egress switch will distribute the information flow among these VCs.

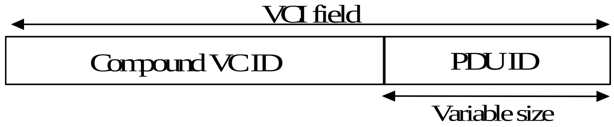

Fig. 1. VCI field in Compound VC

The third one would be to keep standard UNI signaling. That is, the end-user would receive all the information from the group through a single VC. If this solution were adopted, the egress switch would be in charge of avoiding cell interleaving by applying a buffering technique at UNI interfaces, as suggested in [8]. Traffic characteristics will be modified. However, it may be acceptable because buffering is just carried out at the egress switch.

In a generic case, the number of IDs at each link connecting two switches could be different in each direction depending on the aggregate traffic from the sources. However, signaling could be simplified by considering the same number of IDs in both directions, but this would lead to ID space being wasted.

A detailed study of the number of PDU IDs needed for different group sizes and traffic characteristics of the members is carried out in the next section. The most important aspect is that the assignment of the number of PDU IDs may be negotiated at CVC establishment and is not related with the number of buffers of the switch.

IV. SIMULATION RESULTS

The simulated scenario consists of some sources sending traffic to the same output port of a switch. The sources are homogeneous, i.e. all of them have the same statistical parameters. The PDU arrival process follows a geometric distribution with the same mean interarrival time. Each source sends just one PDU at any given time. The length of the PDUs also follows a geometric distribution. An ON-OFF model is used for each source. In the ON state, the source transmits cells at its PCR. The traffic coming from all sources is multiplexed at the switch. The output queue is modeled as a counter of the number of simultaneous PDUs per slot. The number of sources is varied depending on the mean traffic introduced to the switch so as to assure that the losses only occur due to running out of identifiers and not due to overload. The simulated time is 10^{10} μs.

The traffic parameters under study are the peak cell rate (PCR), the average per source (A), and the burst length (B). Simulations were carried out varying these parameters.

Fig. 2 represents the distribution of the number of simultaneous PDUs at the output port of a switch where merging occurs. These results and those in the following two figures were obtained for the reference 3-tuple (PCR, A, B) = (15 Mbps, 0.1 Mbps, 5 cells). Each curve corresponds to a different number of sources (N) in the group, and thus, different average aggregate loads at the switch. The number of senders is chosen to make the system always stable, i.e. the total load from all senders never exceeds the output link capacity. N ranges from 500 to 1500. PDUs are composed of an average of 5 cells and its length follows a geometric distribution. The PDU interarrival time also follows a geometric distribution with an average of 7500 cells. Logarithmic scale has been chosen for both axis to provide a further detail for the range of values of interest, i.e. 16 and 32, which correspond to PDU IDs of 4 and 5 bits, respectively. We focus on these values because few bits in the ID provide low losses due to running out of ID, as will be presented in Fig. 3. Statistically non-significant values have been removed from the figure.

All the curves in Fig. 2 present similar behaviors. The only difference is a shift when the number of sources is varied. Higher values of N show a peak value at higher values of the X-axis. This is so, because the probability of having a big number of simultaneous PDUs is higher when the aggregate load of all the sources is higher, which is the case for the highest values of N that have been studied.

Fig. 3 represents the packet loss probability (PLP), i.e. the probability that an arriving PDU does not find a free ID at the switch, vs. the number of IDs for the 3-tuple (PCR,A,B) = (15Mpbs,0.5Mbps,5 cells). Only values that presented a 95% confidence interval are represented. These curves are obtained from those in Fig. 2 by adding the probabilities corresponding to the cases where an arriving PDU finds all IDs busy.

Fig. 4 represents the probability that an arriving PDU to the switch is correctly multiplexed and forwarded through the output port because it was able to allocate a free multiplexing ID. Note that not all the range of values of X is represented. This range was chosen to provide a deeper insight for the values of interest.

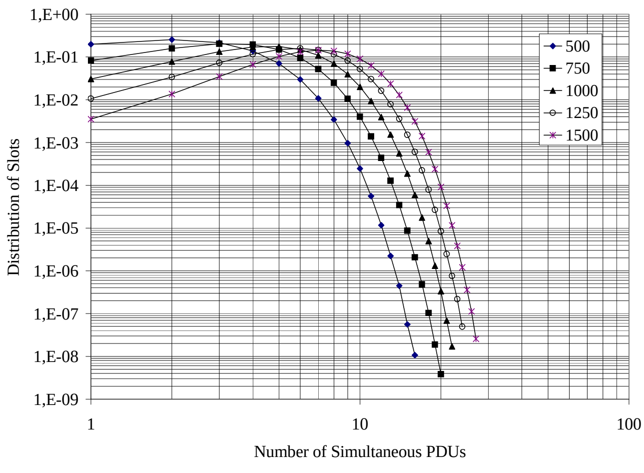

Fig. 2. Distribution of the number of simultaneous PDUs at the output port of a switch where merging occurs. Each curve corresponds to a different number of sources. (PCR,A,B)=(15Mbps,0.1Mbps,5 cells).

The main conclusion that may be drawn from these results is that PDU ID strategies use more efficiently the overhead reserved for the multiplexing ID. Furthermore, the provision of flexible ID size negotiation would allow to obtain low PDU loss probabilities with a few bits (4 or 5), even with a high number of sources. This result is confirmed by Fig. 4, where PDU throughputs higher than 99% are obtained with 16 identifiers (note the linear scale in the y-axis). This scenario, which represents a possible scenario in future group communications, shows the advantages of PDU ID over Source ID multiplexing, which requires a number of IDs assigned to the group equal to the number sources (in this example, up to 1500). As a consequence, the overhead introduced by the multiplexing ID for source ID strategies is much higher than with CVC, which allows the negotiation of the ID size to adapt to the traffic and group requirements. These results are a demonstration of the scalability of CVC, because groups with a high number of sources may be served with a few bits.

Another characteristic that can be observed in Fig. 3 is the linearity (in logarithmic representation) of the PDU loss probability in the range of values of interest. This characteristic would allow a CVC switch to obtain the number of required IDs as a function of group and traffic characteristics and accepted PDU loss probability during connection establishment. The dependence of the parameters of these lines as a function of traffic characteristics is left for further study.

Other simulations have been carried out with different traffic and group parameters. Fig. 5 also presents the PDU loss probability results for a number of sources ranging from 100 to 300 with an average traffic per source of 0.5 Mbps. The same comments stated above apply for this new scenario.

Another of the advantages of having flexible ID size negotiation is the possibility to serve the diversity of scenarios and requirements of different groups. For instance, in multimedia group communications more losses could be accepted for video than for audio, and different number of sources would require different number of IDs.

Finally, the reader should recall that these curves were obtained for a given simulated traffic. Future work will extend the model to consider different types of traffic, and also traces of real traffic.

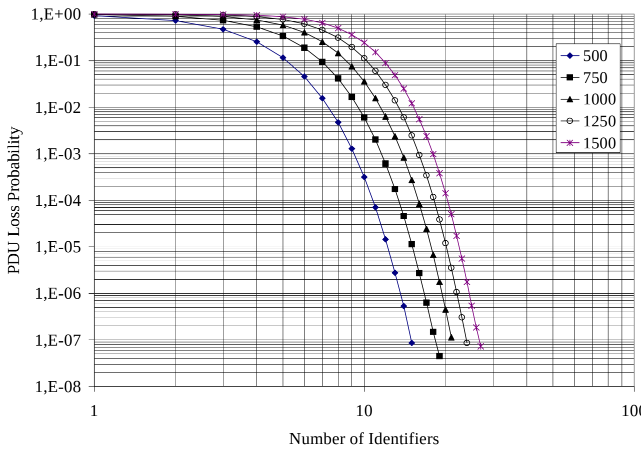

Fig. 3. PDU Loss Probability due to running out of IDs at the switch.

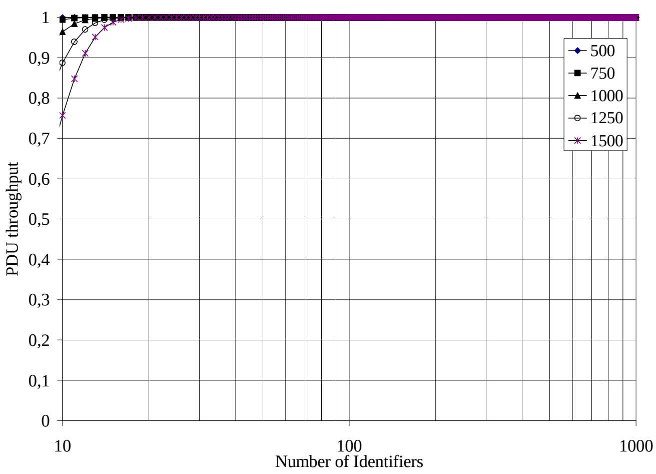

Fig. 4. Probability that an arriving PDU finds a free identifier at the switch. X range from 10 to 1000.

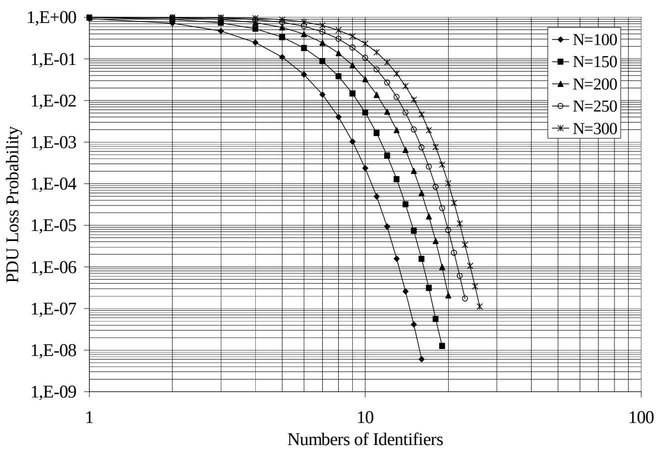

Fig. 5. PDU Loss Probability due to running out of IDs at the switch. (PCR,A,B)=(15 Mbps,0.5Mbps,5 cells)

V. Conclusions

This paper served to study one of the Compound VC Switching strategies named Compound VC (CVC). CVC allows interleaving of cells belonging to different packets. Therefore, it preserves the original characteristics of the traffic being sent by the source. As a consequence, we see that multicast forwarding may not necessarily imply higher delay due to extra buffering. This advantage is obtained at the price of higher VPI/VCI space utilization. However, in a general case, there would be no scalability problems because there usually are much less connections than VCIs on a given link.

The focus has been on one of the processes taking place during connection establishment, i.e. the negotiation of the required number of IDs as a function of the maximum group size, the traffic characteristics, and the accepted packet loss probability. The connection establishment itself was also briefly discussed.

The results showed the advantages of PDU ID with respect to Source ID strategies in terms of overhead.

The convenience of flexible ID size negotiation, such the one offered by the Compound VC (CVC) mechanism, can also be deduced from the diversity in requirements and group characteristics. The scalability of CVC was also confirmed for the scenario under study, because groups with high number of sources may be served with PDU IDs of a few bits.

The PDU loss probability in the range of values of interest, i.e. from 16 IDs (4 bits) to 64 IDs (6 bits), shows a linear trend in logarithmic representation. Finding the expression of this line as a function of the traffic and group characteristics would allow easy ID negotiation during connection establishment. The derivation of this expression is left as future work.

Other traffic models and real traffic traces will be used to obtain more generic results in future work. Implementation issues will also be dealt with.

References

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

[2] Mangues-Bafalluy J and Domingo-Pascual J. ‘Compound VC mechanism for native multicast in ATM networks.’ Proceedings of the 2nd International Conference on ATM (ICATM’99), pp. 115-124, Colmar (France), June 1999.

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

[4] Rosen EC, Viswanathan A, and Callon R. ‘Multiprotocol label switching architecture.’ IETF Draft draft-ietf-mpls-arch-06.txt, August 1999.

[5] Stolyar AL and Ramkrishnan KK. ‘The stability of a flow merge point with non-interleaving cut-through scheduling disciplines.’ Proceedings of INFOCOM’99, pp. 1231-1238, 1999.

[6] 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.

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

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

[9] Venkateswaran R, Li S, Chen X, Raghavendra CS, and Ansari N. 'Improved VC-merging for multiway communications in ATM networks.' Proceedings of the 7th Intl. Conf. on Computer Communications and Networks (IC3N'98), pp. 4-11, 1998.

[10] 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.

[11] 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.

[12] 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.

[13] Chow HK and Leon-Garcia A. ‘VC-merge capable scheduler design.’ Proceedings of IEEE ATM Workshop’99, pp. 153-160, 1999.

[14] Boustead P, Chicharo J, and Anido G. ‘Scalability and performance of label switching networks.’ Proceedings of GLOBECOM’98, pp. 3029-3034, 1998.