Improved VC-Merging for Multiway Communications in ATM Networks

R. Venkateswaran
Lucent Technologies

S. Li
New Jersey Institute of Technology

X. Chen
Lucent Tech. Bell Laboratories

C.S. Raghavendra
Aerospace Organization

N. Ansari
Chinese Univ. of HK (on leave from NJIT)

Abstract

The routing and signaling protocols for supporting multipoint-to-multipoint connections in ATM networks have been presented earlier. VP-Merge and VC-Merge techniques have been proposed as the likely candidates for resolving the sender identification problem associated with these connections. The additional buffer requirements in the VC-Merge mechanism and the excessive use of VPI/VCI space in the VP-Merge mechanism have been the main reasons for concern about their effective utility. In this paper, we propose improvements to the traditional VC-Merge technique to minimize the need for additional buffers at intermediate merge points. Aptly named Dynamic Multiple VC-Merge (DMVC), Fixed Multiple VC-Merge (FMVC) and Selective Multiple VC-Merge (SMVC), these mechanisms define a generic scheme for merging the data from multiple senders onto one or more outgoing links. By appropriately choosing the number of connection identifiers per connection, these schemes lead to a large reduction in the buffer requirements and an effective utilization of the VPI/VCI space. Based on extensive simulations, we show that by using two connection identifiers per connection, there is an 80% reduction in buffer requirements for DMVC and FMVC when compared to the buffer required for traditional VC-Merge.

1 Introduction

Multiway communication involves transferring of data simultaneously from multiple senders to one or more receivers, using a single, shared multicast tree. Such a mechanism can be managed simply by maintaining a separate multicast tree, rooted at each sender. But, this simple scheme does not efficiently utilize the network resources like bandwidth. Moreover, connection management becomes difficult when participants join or leave the connection during the multiway session.

Multiway communication can be supported more efficiently if all the senders share a single multicast tree. Such connections are also called as *multipoint-to-multipoint connections*. A multicast group can be supported using a single multipoint-to-multipoint connection, even when there are multiple senders. Several multimedia applications like video conference, interactive video games and distributed interactive simulations require this support from the underlying network layers. In a multipoint-to-multipoint connection, data from multiple senders are merged into a single connection at appropriate "merge points" and forwarded towards the receivers.

The routing and signaling protocols specified in the current standards for ATM networks do not support multipoint-to-multipoint connections. Only a rudimentary support for point-to-multipoint connections is specified in the standards. But, recently, several protocols to establish multipoint-to-multipoint connections in ATM networks have been proposed for possible standardization[6, 7, 8]. In this paper, we assume that one such mechanism is already implemented to establish multipoint-to-multipoint connections.

Based on the above assumption, a single connection identifier (VPI/VCI) is associated with each multipoint-to-multipoint ATM connection. Since VPI/VCI values have a local significance on a given link, a direct implication of this association is that all the ATM cells of this connection, even those from different senders, use the same VPI/VCI value on that particular link. This conserves the VPI/VCI space and the switch resources, while simplifying the signaling mechanisms when there are several simultaneously active multicast groups. This results in a scalable mechanism for supporting multipoint-to-multipoint ATM connections.

At the sender, the ATM cells are generated by the fragmentation of higher layer packets. Each ATM cell, therefore, does not carry information about the sender and the receiver. The ATM Adaptation Layer (AAL) at the receiver is responsible for re-assembling the original higher layer packets from the individual ATM cells. Cells of different packets originating from different senders intended for the same multicast group may get interleaved with each other.

Since all the cells use the same VPI/VCI value, the receiver may not be able to uniquely identify the sender of a particular ATM cell. Therefore, the original packet cannot be re-assembled at the receiver. This is called the *sender identification problem*, associated with multipoint-to-multipoint connections in ATM networks.

Several solutions have been proposed to solve the sender identification problem for multipoint-to-multipoint connections in ATM networks. In this paper, we provide a systematic study of these solutions. The solutions are classified into two, based on their inability or ability to support interleaving of ATM cells belonging to different packets intended for the same multipoint-to-multipoint connection. We compare the fundamental characteristics of each of these classes of solutions. The factors used for comparison include the buffer requirements, the extra overheads carried within each cell or packet, the complexity of the mechanism, the changes required to existing network components and inter-operability.

VC-Merge and VP-Merge are representative solutions for the two categories of solutions. VC-Merge is fast and scalable, but it requires the use of additional buffers at intermediate merge points. VP-Merge, on the other hand, needs no additional buffers, but its scalability is restricted due to the excessive use of VPI/VCI space. It is therefore desirable to design a scheme that combines the advantages of the VC-Merge and VP-Merge mechanisms. Such a scheme would require very little additional buffers and at the same time, will not be restricted by its use of VPI/VCI space. Design of such schemes is the focus of this paper. This paper proposes a generic scheme for merging data from multiple senders onto one or more outgoing links.

The paper is organized as follows. In Section 2, we categorize and analyze the various solutions to the sender identification problem. Section 3 compares the VC-Merge and VP-Merge schemes under various scenarios. In Section 4, we propose a generic VC-Merge scheme and analyze three mechanisms based on this generic scheme. Section 5 concludes the paper with the results and the direction for future work.

2 Solutions to the Sender Identification Problem

The sender identification problem arises because the receiver may not be able to uniquely identify the source of an ATM cell when the cells from different packets intended for the same multipoint-to-multipoint connection are interleaved. One way of solving the problem is by preventing the interleaving of cells. Since cells from different senders may arrive in any order at an intermediate switch, special mechanisms are required to prevent interleaving of cells. In the next section, we briefly describe some of these mechanisms.

ATM networks, which supports statistically multiplexing, derives some of its advantages due to interleaving of cells. Therefore, it may not be desirable to prevent cell interleaving. In order to support cell interleaving, the identity of the sender of each cell has to be conveyed to the receiver. In a subsequent section, we discuss some of the mechanisms used to convey this information from the sender to the receiver.

2.1 Mechanisms that prevent cell interleaving

The mechanisms discussed in this section prevent cell interleaving by sending all the cells of a packet contiguously. The sender can be identified from the reassembled packet at the receiver(s). Note that interleaving of cells belonging to different connections is *not* restricted by any of these mechanisms.

In the *Multicast Server (MCS) approach*[1], a centralized multicast server ensures contiguity of all the cells of a packet. The senders of a multipoint-to-multipoint connection first send the data to a pre-assigned multicast server responsible for forwarding the data packets to all the receivers. The scheduler at the MCS prevents interleaving of cells belonging to different packets. This approach can be easily deployed on existing networks supporting point-to-multipoint connections. But, the lack of scalability and single point of congestion and failure are its main disadvantages.

A token-based approach[3] requires a user to possess a token for sending packets to a multipoint-to-multipoint connection, thereby, restricting multiple users from simultaneously sending packets to the same connection. The token is passed on from one sender to another. Though this scheme works especially well for links with limited bandwidth, it does not scale well to large number of senders because of the overheads involved in token-passing and recovery of lost tokens.

Buffering of cells at appropriate "merge points" and intelligent scheduling can be used to prevent cell interleaving. In one possible implementation called the *store and forward VC-Merge* mechanism[4], an intermediate switch buffers all the cells of a packet till the entire packet reaches the switch. The cells are then scheduled to be contiguously forwarded towards the destination(s) using the entire bandwidth allocated for this connection. Note that cells of a packet can interleave with cells of packets intended for a different connection, which distinguishes this scheme from traditional packet switching. An alternate implementation called the *virtual cut-through VC-Merge* scheme allows an intermediate switch to schedule partial packets for forwarding. But once a packet is scheduled, cells of other packets intended for the same multipoint-to-multipoint connection have to be buffered till the scheduled packet is completely forwarded. This wait depends on the rate at which the scheduled packet is arriving.

The VC-Merge approaches described here are efficient because the cells *need not* be reassembled at each intermediate switch. Instead, the end-of-packet (EOP) indicator as specified in AAL5 is used to detect the end of a particular packet. VC-Merge has very little computational overhead, but needs additional buffers at appropriate intermediate switches, whose size depends on the number of senders, the traffic characteristics and the packet sizes. In subsequent sections, we study the buffer requirements and propose mechanisms to reduce the amount of additional buffers.

2.2 Mechanisms that support ATM cell interleaving

In order to support ATM cell interleaving for multipoint-to-multipoint connections, the identity of the sender *must* be included in each cell. In the VC-Mesh approach[1], each sender to a multipoint-to-multipoint connection establishes a separate point-to-multipoint connection identified by distinct VPI/VCI field. This allows cell interleaving, but complicates dynamic changes to the set of senders and receivers due to the maintenance of large number of connection states.

Alternately, the sender information can be encoded in the 10-bit multiplex ID (MID) field of AAL3/4 ATM Adaptation Layer. Each sender has to be assigned a unique MID value using some additional mechanisms, thereby, restricting the number of senders to 1024. AAL3/4 is not widely used because the payload is limited to only 44 bytes (as defined in AAL3/4), limiting the effective utilization to 83%.

In the *VP-Merge* technique, only the VPI field is used to identify a multipoint-to-multipoint connection and a VCI value, unique to each sender within the VPI value, is used to identify the sender. The cells are switched on the VPI value and the VCI value is carried undisturbed. This scheme can be implemented on existing VP switches, but the scalability is limited because the number of independent multipoint-to-multipoint connections on a given link can be at most 4096 ($= 2^{12}$).

In the widely prevalent AAL5 adaptation layer standard, there is no field for sender information in the header or the payload. A new AAL (currently non-standard) that includes the sender information as part of the 48 byte payload can be proposed for multipoint-to-multipoint connections. In one possible implementation, the value of first two bytes of the ATM payload, uniquely assigned to each sender, can be used to identify the sender. In this implementation, the actual payload is only 46 bytes long, thereby, limiting the effective utilization to 86.79%. Proposing this new protocol may lead to incompatibility with existing infrastructure. Further, a single bit error in the non error-corrected sender value will affect packets from two different senders.

A scheme proposed in [5] facilitates the use of standard AAL5 protocols by introducing a Resource Management (RM) cell to carry sender identities. Each RM cell contains identities of the senders of the following few ATM cells intended for a particular multipoint-to-multipoint connection. The receiver interprets the RM cell to correctly identify the senders of these ATM cells. The RM cell on a given link is significant only to the switches at either end of the link. The RM cell is therefore created at each switch in accordance with the scheduling policy of that switch. This prevents the propagation of wrong information due to lost cells. Though the effective utilization for two-byte long sender identities is 86.79%, the performance may be affected due to the creation of RM cell at each switch. Since the loss of an RM cell can affect several packets, the RM cells are sent with CLP-bit 0 to minimize the chance of losing these cells.

3 Comparison between VC-Merge and VP-Merge

We now compare the two categories of solutions described in the preceding sections. The VC-Merge techniques and the VP-Merge techniques are used as representative techniques for the two categories of solutions for solving the sender identification problem. The VC-Merge uses a coarser granularity of multiplexing based on packet interleaving, while VP-Merge uses a finer granularity of multiplexing based on cell interleaving.

VC-Merge has been accepted to support multipoint-to-point connections in future versions of PNNI specifications. We believe that VP-Merge is an equally good alternative. Connections based on VP-Merge technique closely resemble ATM connections because it supports statistical multiplexing of ATM cells. Moreover, the buffer requirement for VP-Merge is much smaller than that for VC-Merge. To study the buffer requirements, we simulated a typical scenario for VP-Merge and VC-Merge at a merge point. A merge point is a switch that has at least two incoming links for the multiway connection and one outgoing link that is different from the two incoming links.

3.1 Simulation Study

We studied the buffer requirements at a merge point for multipoint-to-point connections. We first considered a merge point with several senders, as shown in Figure 1. We call this the *Star configuration*. There are several senders S1, S2 ... Sn, sending data towards a merge point M. The cells from these senders are merged and sent out from the

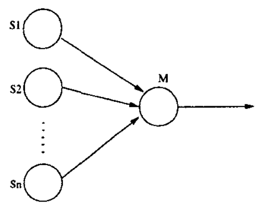


Figure 1. Star configuration

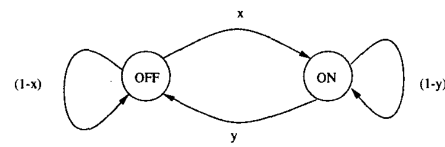


Figure 2. Simple ON-OFF traffic model

merge point. We analyzed the amount of buffer required at the merge point by varying the number of senders. Each sender is assumed to be a bursty source. All the senders are assumed to be identical sources and the capacity of the outgoing link from the merge point is normalized to 1. Therefore, for a stable system, it is required that the sum of the loads of the senders is less than 1. We assume that there are no cells lost in transit.

A bursty source, characterized by the peak cell rate (P), the average cell rate (A) and the average number of cells per burst (B), can be modeled as an ON-OFF source in the discrete-time domain (two-state Markov Modulated Deterministic Process (MMDP)[2]), as shown in Figure 2. In the OFF state, the source does not send any cells. In the ON state, the source sends data cells at the peak cell rate (P). In a discrete-time domain, the source can independently shift from one state to the other only at the end of a time-slot. Here, we assume that the duration of the time-slot is the time taken to generate a cell. At each time slot, the source in the OFF(ON) state changes to the ON(OFF) state with a probability $x(y)$. It must be remembered that there is no correlation between the two probabilities. The probabilities of the source being in the OFF state and ON state are given by $P_{off} = \frac{y}{(x+y)}$ and $P_{on} = \frac{x}{(x+y)}$ respectively. In terms of the bursty source parameters, the state transition probabilities are $x = \frac{A}{B(P-A)}$ and $y = \frac{1}{B}$.

Using this model for the sources, we simulated the buffer requirements at the merge point. We simulated three cases, which are described here.

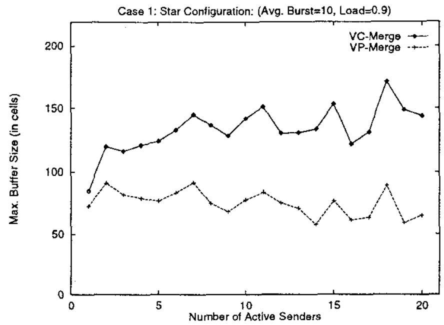
- Case 1: The entire packet is transmitted at the end of a burst. Further, the incoming links are slow links and the outgoing link from the merge point is a faster link.
- Case 2: The entire packet is transmitted at the end of a burst. But, the incoming and outgoing links have the same speed.
- Case 3: The packets are of fixed length and the complete packet may not be transmitted at the end of a burst. The incoming and outgoing links have the same speed.

The plot of comparison for the first case is shown in Figure 3(a). In this figure, the average burst length is 10. From the figure, it is clear that, on an average, VC-Merge technique uses 83% more buffer than the VP-merge technique. We repeated the same experiment using different values of average burst length. As the average burst length increases, the buffer requirement for the VP-Merge technique increases correspondingly. This leads to a decrease in the overall percentage increase in buffer for the VC-Merge technique. The plot for average burst length = 20 is given in Figure 3(b). From the figure, we can see that the average increase in buffer requirement is about 59% for VC-Merge technique over the VP-Merge technique. The difference becomes even more pronounced as the traffic becomes smoother as shown in Figure 4. This is due to the fact that very little buffering is required for the VP-Merge technique when the traffic is smooth.

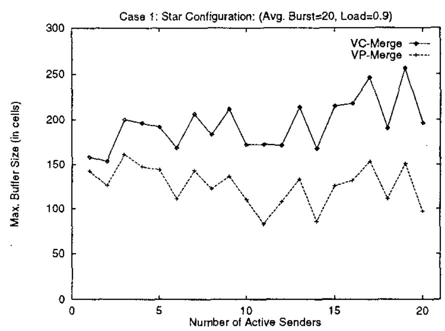
In the second case, we studied the buffer requirements when the incoming and outgoing links have the same speed. In this scenario, the peak cell rate of the sender is equal to 1, the normalized outgoing link rate. The average cell rate (the load) is adjusted as the number of senders increase to ensure the stability of the system. Again, it is assumed that the entire packet is generated during a burst. The buffer requirements when the average burst length is 10 are plotted in Figure 5.

In this scenario, the VC-Merge requires about 27% more buffer on the average. The difference is smaller than the previous case because entire packets reach the merge point faster because of the faster incoming links. Therefore, cells on an incoming link become available for switching at a faster rate. The difference decreases further as the average burst length increases due to the same reasons as explained in the previous case.

Next, in the third case, we studied the buffer requirements when the entire packet is not generated during a burst. The packets are assumed to be of fixed length and consisting of 30 data cells. Again, the incoming and the outgoing links have the same speed. The results of the plot for average burst length = 10 are shown in Figure 6. In this case, the buffer requirement for VC-Merge increases because the entire packet may not arrive in a single burst. The buffer requirement for VP-Merge remains the same as in the previous case.



(a) Burst Length = 10



(b) Burst Length = 20

Figure 3. Star Configuration: Case 1: VC-Merge vs VP-Merge

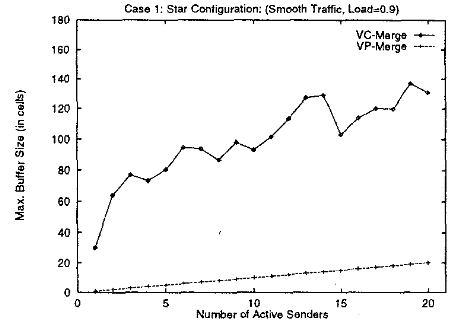


Figure 4. Star Configuration: Case 1: VC-Merge vs VP-Merge: Smooth

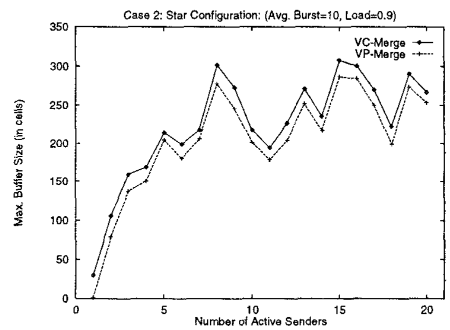


Figure 5. Star Configuration: Case 2: VC-Merge vs VP-Merge

3.2 Summary of the Results

From the simulation studies, we can conclude that VP-Merge has a clear advantage over VC-Merge based on the buffer requirements. The advantages are more pronounced for smoother traffic than for bursty traffic. In this respect, we concur with the opinions expressed in [9]. So, VP-Merge is a better alternative to VC-Merge with respect to buffer requirements. But, the main problem with VP-Merge is its poor scalability. Moreover, it is difficult to implement congestion control mechanisms like Early Packet Discard (EPD) in a VP-Merged connection.

4 Improved VC-Merge mechanisms

The ATM Forum has accepted VC-Merge to support multipoint-to-point connections in future versions of PNNI specifications. In this section, we describe some improvements to the traditional VC-Merge schemes that reduce the buffer requirements at the intermediate switches at the cost of increased utilization of the VPI/VCI space. Since the number of different connections supported on a given link is much less than the available VPI/VCI space, this increased utilization does not affect the scalability of the proposed mechanisms.

4.1 Multiple VC-merge mechanisms

We propose some improvements to the VC-Merge approach to minimize the the buffer requirements at the intermediate switches. These improvements, referred as *multiple VC-Merge mechanism*, adopt the use of multiple VPI/VCI values for a particular multipoint-to-multipoint connection. In some sense, each connection has multiple connection identifiers. Note that the VPI/VCI values still retain local significance on a given link. Some signaling protocol is required to map multiple connection identifiers to the same multipoint-to-multipoint connection. This could be done either a priori during the connection set up phase or

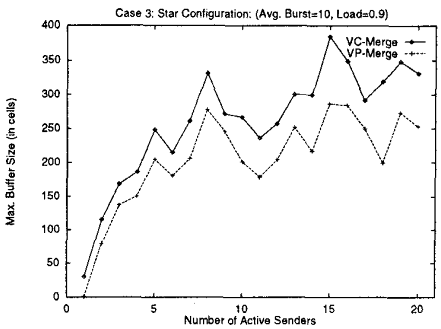


Figure 6. Star Configuration: Case 3: VC-Merge vs VP-Merge

dynamically depending on the performance of the system. The details of the signaling mechanism is outside the scope of this paper.

The multiple VC-Merge mechanism does not affect systems that use *only* store and forward VC-Merge scheme. But, it has a great impact on systems that incorporate virtual cut-through VC-Merge mechanisms. On these systems, though this scheme increases the VPI/VCI space for each multipoint-to-multipoint connection, it improves the throughput and reduces the buffer requirements at intermediate switches. This improvement is mainly due to the fact that the multiple VC-Merge mechanism restricts interleaving of cells belonging to different packets *only* on the same connection identifier, but permits interleaving of cells on different connection identifiers referring to the same multipoint-to-multipoint connection.

The multiple VC-Merge mechanism is a generalized merge mechanism. On one extreme, if there is exactly one connection identifier for a particular multipoint-to-multipoint connection, the buffer requirements and the characteristics of the connection resemble the traditional VC-Merge mechanism. On the other extreme, when the number of connection identifiers equals the number of senders for a multipoint-to-multipoint connection, the buffer requirements and the characteristics of the connection resemble the traditional VP-Merge mechanism. Typically, multiple VC-Merge mechanism operates between these two extremes.

We now discuss two possible implementation schemes for the multiple VC-Merge mechanism in a system that supports only virtual cut-through mechanism.

4.1.1 Fixed Multiple VC-Merge Mechanism (FMVC)

In this implementation scheme, a switch that acts as a merge point for a particular multipoint-to-multipoint connection statically assigns one of the corresponding connection identifiers to each sender. All the cells originating from a sender intended for that connection are forwarded on the outgoing link(s) using the assigned connection identifier. The identifier is assigned when the sender joins the connection and remains fixed till the sender leaves the connection. If the number of connection identifiers is less than the number of senders, more than one sender will be assigned the same connection identifier. This results in partitioning the set of senders into *identifier groups*, where each identifier group of senders is assigned a particular connection identifier. Though it is possible to interleave cells belonging to packets that originate from senders that are in different identifier groups, it is *not* possible to interleave cells belonging to packets originating from two senders that are in the same identifier group. Since the partitioning and assignments are fixed, it may happen that some cells belonging to particular identifier group have to be buffered even though there are no active cells belonging to some other identifier group.

4.1.2 Dynamic Multiple VC-Merge Mechanism (DMVC)

In the dynamic implementation, a switch that acts as a merge point for a particular multipoint-to-multipoint connection maintains the set of unassigned connection identifiers on the outgoing link(s) pertaining to that connection. When the first cell of a packet intended for that connection arrives at this merge point, one of the unassigned connection identifiers from that set is assigned to this packet. This identifier is then removed from the set. All the cells of this packet use this assigned identifier on the outgoing link(s). Once the entire packet of cells is transmitted, the assigned identifier is released back to the set.

If there are no unassigned connection identifiers when a packet arrives at a merge point, the cells of that packet are buffered till one of the identifiers becomes free. This results in the efficient utilization of the connection identifiers.

It is possible to implement DMVC in a network only if all the switches are capable of mapping multiple connection identifiers to the same logical queue. Typically, at a given switch, all the cells arriving on a particular input port having the same connection identifier are assigned to the same logical queue (either input or output). In a multiple VC-Merge scenario, the switch must be capable of mapping multiple connection identifiers to the same logical queue. If all the switches in a network do not have the capability of mapping multiple connection identifiers to the same logical queue, then it is possible that packets originating from a particular sender may arrive out of sequence at a receiver. Though it is possible to resequence the packets using higher layer protocols, this is against the philosophy of connection-oriented networks like ATM. In networks comprising of some switches that cannot map multiple connection identifiers to the same logical queue, a mechanism like FMVC has to be implemented to solve the sequencing problem.

4.1.3 Selective Multiple VC-Merge Mechanism (SMVC)

The FMVC and DMVC implementations described in the previous sections do not impact the store and forward VC-Merge mechanism. These schemes can be enhanced by maintaining an additional connection identifier for store and forward VC-Merge mechanism. This scheme is called the Selective Multiple VC-Merge mechanism (SMVC). In the simplest implementation of SMVC, two connection identifiers are maintained for each multipoint-to-multipoint connection. One connection identifier is used for virtual cut-through VC-Merge scheme and the other is used for store and forward VC-Merge scheme. When all the cells of a packet intended for a multipoint-to-multipoint connection is available at a merge point, the store and forward connection identifier is used to forward these cells on the outgoing link(s). It is not necessary to initiate forwarding using the virtual cut-through mechanism because the bandwidth allocated for this multipoint-to-multipoint connection is fully utilized for the store and forward mechanism. If only partial packets are available, one of the partial packets is scheduled on the outgoing link(s) using the virtual cut-through connection identifier. All the cells of this packet will be eventually forwarded using this connection identifier. This improves the link utilization and reduces the buffer requirements at the merge point.

4.2 Simulation results

We performed extensive simulations to study the buffer requirements for each of the proposed improvements. We focus on the buffer requirements at a particular Merge Point of a single multipoint-to-multipoint connection. Specifically, we studied the "star" configuration as shown in Figure 1 of Section 3.1. In order to study the effect of heterogeneous senders on the buffer requirements, we used two sets of senders, the fast and the slow senders. The fast senders generate cells at twice the rate as the slow senders. In order to prevent buffer overflow, we maintained the total load on the outgoing link at the merge point at 90% of its capacity. For lack of space, we present the results from only one scenario used in our simulations. In this scenario, we assumed that the average burst length of each source is 10 cells and the burstiness factor is 2.

In Figure 7, we compare the DMVC and FMVC mechanisms with respect to the virtual cut-through VC-Merge technique. Since the sources are not very bursty(burstiness = 2) and the load on the outgoing link is only 90% utilized, VP-Merge mechanism requires very little buffers at

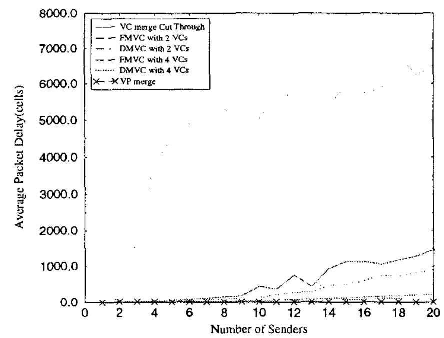


Figure 7. Comparison of DMVC and FMVC with virtual cut-through VC-Merge

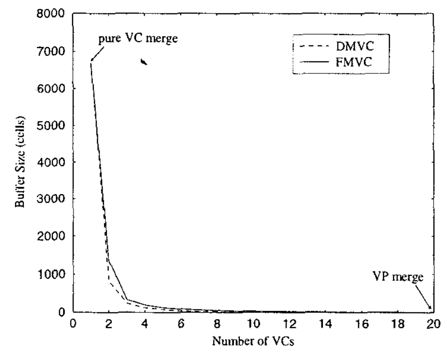


Figure 8. Comparison of DMVC and FMVC with increasing number of identifiers

the merge point. In this scenario, the buffer requirements for virtual cut-through mechanism is very large, as is evident in Figure 7. This is due to the fact that when cells from a slow source are scheduled on the outgoing link, cells from other sources have to be buffered till an entire packet of cells is transmitted out. The slow source does not efficiently utilize the outgoing link. The multiple VC-Merge techniques, DMVC and FMVC, reduce the buffer requirements by about 80% just by using two connection identifiers per connection. This reduction results from the ability to schedule two packets, one using each connection identifier, simultaneously. The utilization of the outgoing link improves by a great extent. As expected, the DMVC technique does a little better than the FMVC. The buffer improvements are marginal when the number of identifiers is increased to 4.

We now compare the buffer requirements using DMVC and FMVC techniques as the number of connection identifiers increases. The plot of comparison is shown in Figure 8. In this plot, the number of senders was fixed at 20. At one

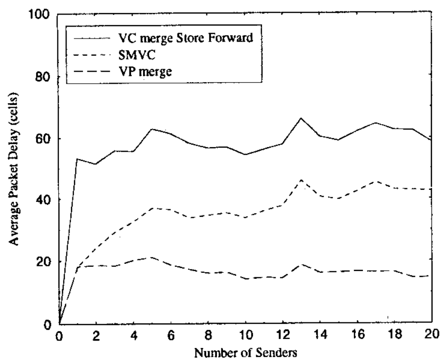


Figure 9. Comparison of SMVC with store and forward VC-Merge

extreme of the plot, when there is exactly one connection identifier per connection, the buffer requirements are identical to that of virtual cut-through VC-Merge technique. At the other extreme, when the number of connection identifiers equals the number of senders, the buffer requirements are similar to the VP-Merge technique. In a typical multiple VC-Merge scenario, we operate between the two extremes. From the figure, it is clear that the buffer reduction is phenomenal when the number connection identifiers is increased to 2. There is further improvement as the number increases to 4. But, there is hardly any improvement beyond 10 connection identifiers per connection.

Figure 9 compares the average buffer size required by SMVC with those required by VP-Merge and store and forward VC-Merge techniques. From the plot, it is evident that the buffer requirements for SMVC are about 50% less than that for the VC-Merge technique. This is due to the use of two connection identifiers for the same connection. In the extreme case when there is only one active sender, the buffer requirements for VP-Merge and SMVC are identical because both can forward the cells of the packets immediately. On the other hand, the VC-Merge scheme *has* to buffer the cells until the entire packet has reached the Merge Point. In a general sense, use of two connection identifiers improves the buffer requirements at the Merge Point by about 50%.

5 Conclusion

We propose three new schemes for improving the performance of traditional VC-Merge techniques for the support of multipoint-to-multipoint connections in ATM networks. Aptly named DMVC, FMVC and SMVC, these mechanisms define a generic scheme for merging data from multiple senders onto one or more outgoing links. The mechanisms combine the advantages of VP-Merge and VC-Merge in terms of the effective conservation of the VPI/VCI space and the reduction in the buffer requirements at intermediate merge points. Using extensive simulations, we show that there is a 80% reduction in buffer requirements just by using two connection identifiers per connection. These schemes, thus, operate between the two extremes of VC-Merge and VP-Merge. Future work will involve the design of signaling mechanisms for the support of these generic schemes.

References

[1] G. Armitage. *Support for Multicast over UNI 3.0/3.1 based ATM Networks.* RFC2022, 1996.

[2] R. Bolla, F. Davoli, and M. Marchese. Evaluation of a Cell Loss Rate Computation Method in ATM Multiplexers with Multiple Bursty Sources and Different Traffic Classes. In *Proc. of IEEE GLOBECOM*, pages 437–441, Nov. 1996.

[3] E. Gauthier, J.-Y. Le Boudec, and D. Dykeman. *SMART: A Many-to-Many Multicast Protocol for ATM.* ATM Forum draft 96-1295, 1996.

[4] M. Grossglauser and K. K. Ramakrishnan. *SEAM: A Scheme for Scalable and Efficient ATM Multipoint-to-Multipoint Communication.* ATM Forum draft 96-1142, 1996.

[5] S. Komandur, J. Crowcroft, and D. Mosse. CRAM: Cell Relabeling at Merge-Points for ATM Multicast. Personal Communication, Univ. of Pittsburg.

[6] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V. Kumar. Hierarchical Multicast Routing in ATM Networks. In *IEEE Intl. Conf. on Communications*, volume 3, pages 1690–1694, June 1996.

[7] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V. Kumar. A Scalable, Dynamic Multicast Routing Algorithm in ATM Networks. In *IEEE Intl. Conf. on Communications*, 1997.

[8] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V. Kumar. *Support for Group Multicast in PNNI.* ATM Forum draft 97-0076, 1997.

[9] I. Widjaja and A. I. Elwalid. *Performance Issues in VC-Merge Capable Switches for IP over ATM.* ATM Forum draft 97-0675, 1997.