RZ 3055 (# 93101) 09/07/98
Computer Science/Mathematics 15 pages

Research Report

VC versus VP Merging in ATM Networks

A. L. Schmid

Swiss Federal Institute of Technology (ETH)
8092 Zürich
Switzerland

I. Iliadis and P. Droz

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

IBM Research Division
Almaden · Austin · Beijing · Haifa · T.J. Watson · Tokyo · Zurich

VC versus VP Merging in ATM Networks

A. L. Schmid

Swiss Federal Institute of Technology (ETH), 8092 Zürich, Switzerland

I. Iliadis and P. Droz

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract

For the implementation of multipoint-to-point connections in ATM, various approaches exist, each with its own advantages and disadvantages. VP-based methods require unique sender identification but they do not require reassembly in merging points. In contrast, VC-based methods do not require unique sender identification but they do require reassembly in merging points. It is likely that VC merging will be the method of choice as it is scalable and yet relatively simple to implement. One of its drawbacks is the increased output buffer space required at the switches because of packet pseudo-reassembly at the merging points. This paper investigates the impact of the switch architecture and characteristics on the output buffer space by means of simulation. The results obtained demonstrate that for typical switch architectures, VC merging does not require significant additional buffering compared to VP merging.

1 Introduction

In current ATM networks, there exist only *point-to-point* (pt-to-pt) and *point-to-multipoint* (pt-to-mpt) connections. For the interconnection of routers across an ATM network as well as for many other information-gathering applications, *multipoint-to-point* (mpt-to-pt) connections appear to be more appropriate. Interconnection of N routers requires order N^2 labels for the order of N^2 pt-to-pt connections as described in [1]. With mpt-to-pt connections only N labels for the N associated connections are necessary. This significantly reduces the required label space and thus makes the method more scalable. The same new ATM connection type could also be used in the context of merged connections for MPLS [2].

To implement mpt-to-pt connections, different solutions are possible. We focus on the two most important methods: *VP merging* and *VC merging*.

- *VP merging*: Each sender is assigned a globally unique identifier having the format of a VCI. The identifier is carried in the VCI field of the ATM cell. The ATM switch translates incoming VPIs for the same destination to the same outgoing VPI. The receiver distinguishes amongst the different sources by the different VCIs. The key advantage of this scheme is that no VCC resources are required in the switching nodes as only VP switching is performed. This implies no change of hardware but only a change of the connection establishment protocol. Some of the disadvantages of VP merging are the lack of scalability caused by the VPI address space limitation of 4096 entries and the need for a "global VCI uniqueness" protocol. There are proposals to circumvent the nonscalability by enlarging the VPI address space at the expense of VCI address space. This is not desirable, however, because it requires changes in the switching hardware.

- *VC merging*: This method avoids the requirement for globally unique sender identifiers, and it consumes only one VCI per traversed link. These characteristics make this approach scalable. The ATM switch translates incoming VCIs belonging to the same connection to a single outgoing VCI. This means that cells of packets belonging to different senders could be interleaved. As the receiver is not able to distinguish cells from different senders, packet pseudo-reassembly has to be performed at the merging points, and all cells from a given packet must be sent contiguously so that pseudo-reassembly at subsequent merging points and at the receiver will be possible. AAL 3/4 would solve the problem by introducing the Message Identifier (MID) field for sender identification in every cell. The use of AAL 3/4, however, has other drawbacks such as the limited space of the MID field, the inefficient encapsulation method, and the less powerful CRC capability. In this paper we consider the employment of AAL 5 because it is widely available and supported in ATM switches, especially in data networks. Packet pseudo-reassembly at the merging points introduces additional buffer requirements on the switching architecture because all of the cells of a packet sent by a sender belonging to a mpt-to-pt connection have to be stored and must wait for the last cell of the packet identified by the "End Of Packet" (EOP) marker used by AAL 5 to arrive. Figure 1 depicts the cell interleaving problem. Packet pseudo-reassembly also introduces additional delay for packets transported over a merged connection and adds burstiness to the traffic. This is because all the cells of a packet have to wait at every merging point. They appear afterwards as a burst of a whole packet at the output link. This burstiness becomes even worse as it is often cascaded and thus accumulated over numerous merging points. Reference [3] gives some hints about how to solve the problems involved in mpt-to-pt VC merging.

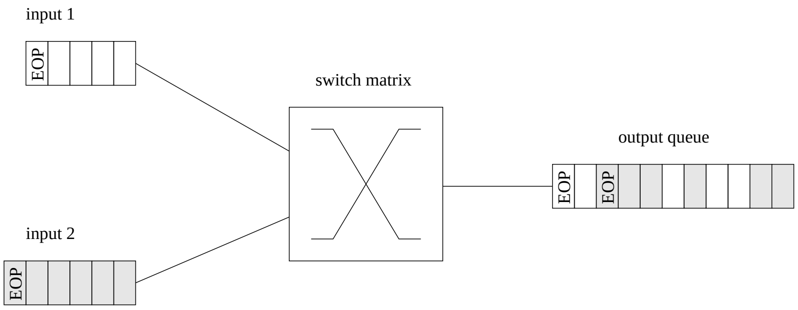

Figure 1: Cell interleaving problem.

A third possibility is to handle a mpt-to-pt connection of N senders to one receiver like N pt-to-pt connections without applying any merging. This possibility again requires order N^2 labels for the order of N^2 pt-to-pt connections. Of the above possible solutions, VC merging appears to be the method of choice as it is relatively easy to implement and yet scalable. At the ATM Forum, VC merging has been accepted and will be introduced in the PNNI v2.0 specification (expected to be finished in the autumn of 1998). The only concern is with the pseudo-reassembly required in the switches in terms of additional buffering and delay. The numerous simulations presented in the following sections are used to investigate the required additional buffer overhead for VC merging. It is also very likely that different methods of merging and nonmerging will exist simultaneously in an ATM network. Some interworking aspects of these methods are discussed in [4].

Section 2 of this paper describes our switching architecture model for VC merging and the model of the arriving traffic. In Section 3 we show our simulation setup and discuss the results of the simulations. In Section 4 we give a summary and derive some conclusions.

2 Switch and Traffic Model

2.1 Switch Model

In this paper we consider the general class of single-stage, nonblocking $N \times N$ packet switches with both input and output queuing [5, 6]. The shared output buffer is assumed to be sufficiently large so that the switch performance is close to optimal, corresponding to the pure output queuing. Cells are transferred from the head of the input queues to the shared buffer. The speed of the input and output switch ports is denoted R_S, and the speed of the outgoing links is denoted R_L. Let k denote the speed ratio of the switch speed (per port) to the outgoing link speed, i.e. $k = R_S/R_L$. Typically k is greater than one, which implies that an output queue should be provided in order to cope with the speed mismatch.

As described above, VC merging requires a certain amount of additional output buffering due to the packet pseudo-reassembly. We introduce a so-called reassembly buffer at each output port of the ATM switch. Figure 2 shows the concept of the reassembly buffer. A switch has N input ports and M sources of mpt-to-pt connections at every input port. For our simulations we assume that the same number of sources arrive at every input port. All sources are assumed to participate in mpt-to-pt connections. Therefore, at every merging point, each of the sources is associated with a distinct reassembly buffer at the output queue. This results in a maximum of $M \times N$ reassembly buffers at every output port. When the last cell of a packet with the EOP marker arrives at the reassembly buffer, all of the cells of a packet are instantly transferred into a single output buffer per output port. Physically the

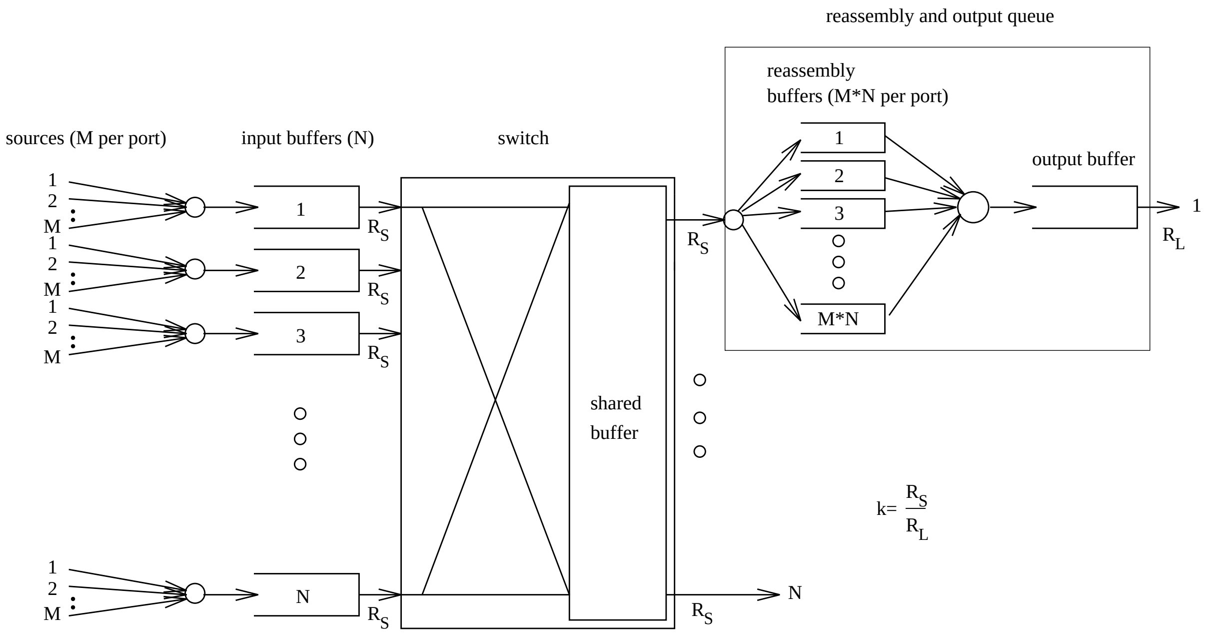

Figure 2: Switch architecture.

reassembly and the output buffers of one output port share a common memory pool. The transfer from the reassembly to the output buffer can easily be done by a pointer movement and will therefore not incur additional delay.

The simulation models for VC and VP merging are shown in Figures 3 and 4, respectively. Cells belonging to the various VCs are transferred from the head of the switch input queues in the shared reassembly buffer and, subsequently, to the corresponding output queues. It is assumed that the traffic is uniform, i.e. the destination of an arbitrary packet can, with an equal probability, be any of the output ports, and that successive packets are independent regarding their output port destinations. Owing to traffic symmetry, all of the output queues have identical behavior. Let us turn our attention to a particular output queue and study its behavior. The corresponding simulation model considers $M \times N$ sources feeding the output queue in a round-robin fashion serving the N input ports governed by the factor k. This model is also appropriate for the case where the switch fabric is capable of transferring only a limited number of cells to any given output [7].

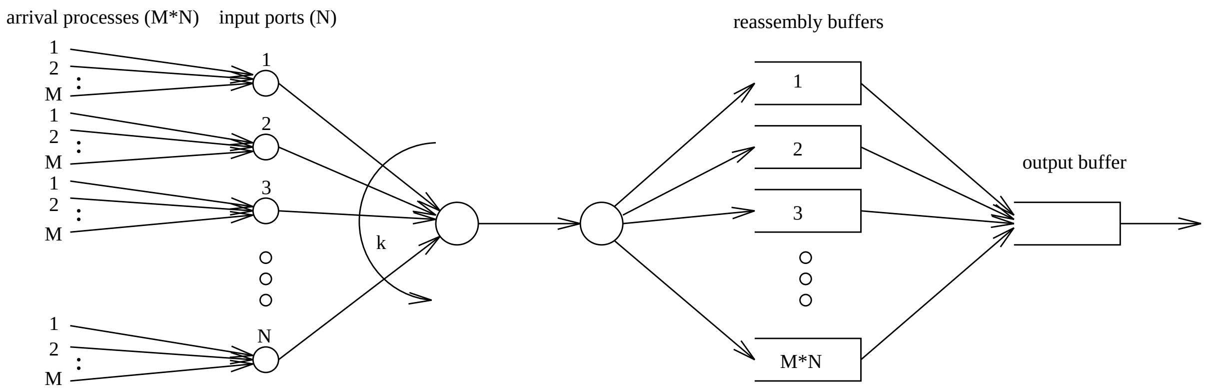

Figure 3: Our simulation model for VC merging.

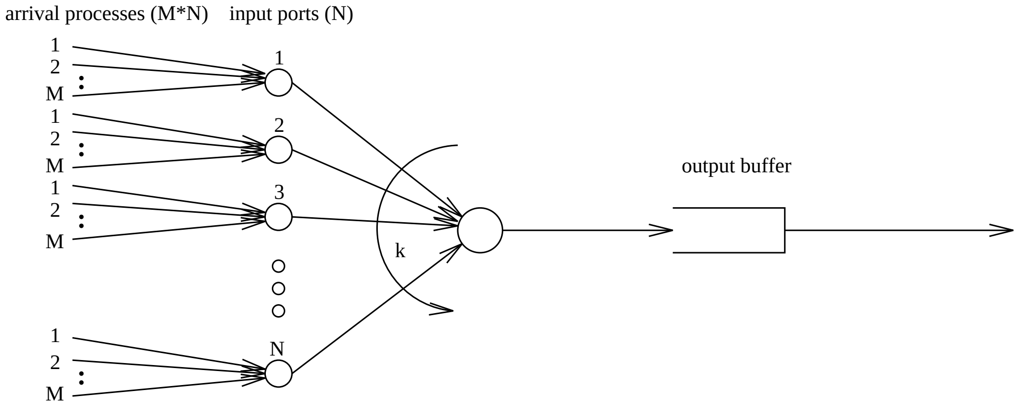

Figure 4: Our simulation model for VP merging.

2.2 Traffic Model

The traffic and simulation model we use is shown in Figures 3 and 4. We use $M \times N$ arrival processes, which correspond to the traffic destined to the output queue. Packets are assumed to arrive according to either a Poisson process (nonbursty traffic with the mean arrival rate λ) or a hyperexponential process (bursty traffic). The hyperexponential process is generated by a two-stage hyperexponential distribution. The mean values corresponding to the two stages are $0.51 * \lambda$ and $16.48 * \lambda$, respectively. The corresponding routing probabilities for the two stages are 0.97 and 0.03, respectively, so that the mean arrival rate is again equal to λ. Each packet is assumed to contain a number of cells geometrically distributed with a mean of E cells [8, 9]. We used $E = 10$, $E = 30$, or $E = 180$ cells (10 cells correspond to 472 bytes, 30 cells to 1432 bytes and 180 cells to 8632 bytes).

It is shown in [9] that the mean packet size in a core network where ATM is likely to be applied is about 289 bytes. This yields 6.2 ATM cells of data using AAL 5 with the null encapsulation method as described in [10] (additional overhead of AAL 5 is 8 bytes per packet). The dominant packet sizes in an Internet backbone are 40 or 44 bytes at about 36% of the traffic (TCP acknowledgment packets, TCP control segments such as SYN, FIN, ..., and Telnet packets carrying single characters), 552 or 576 bytes at about 25% (512 and 536 bytes of TCP implementations without path MTU discovery as the default maximum segment size (MSS) for nonlocal IP destinations, yielding a 552 or 576-byte packet size), 185 bytes at about 2.7%, and 1500 bytes at about 1.5% (Ethernet traffic). These statistics were collected on Feb 10, 1996, in FIX-West network as a sample wide-area network, and are shown in [11]. A more recent study of traffic characteristics in an Internet backbone conducted in August of 1997 is presented in [12]. It is shown that almost 50% of the traffic is 40 or 44 bytes in packet length. More prominent packet sizes are 532, 576, and 1500 bytes, each representing 15% of the traffic. Comparing the two studies we observe a shift to smaller packets of size 40 or 44 bytes and larger packets of size 1500 bytes.

For the future development of packet sizes, the spreading of the use of path MTU (PMTU) discovery will have a significant impact. PMTU will affect MTUs in IPv4 as proposed in [13] and even more MTUs in IPv6 over faster LANs. There will be numerous packets with possible sizes up to 64 kilobytes (max. packet format for AAL 5 is 64 kilobytes [14]). A single packet of this size involved in reassembly could alone fill the entire reassembly buffer in a switch output queue. Reference [15] gives an overview of other typical frame sizes being applied on AAL 5. These are 8 kilobytes used by the Network File System (NFS) and the 9180 bytes of IP MTU over SMDS [16] that became the default value for IP MTU over ATM AAL 5 [14].

These big packet sizes in conjunction with VC merging could induce present problems that VP merging would not encounter. On the other hand there will also be much more real-time traffic (e.g. voice) in the Internet. Real-time traffic typically produces a large amount of very small packets.

3 Simulation Setup and Results

This study concentrates on the additional buffer space required for packet pseudo-reassembly. It is conducted under loads $l = 30\%, 70\%, 90\%$, with different traffic characteristics (bursty, nonbursty), factor $k = [1, .., 16]$, $M = 1, 4, 8, 16, 32, 64$ sources, $N = 4, 16, 64, 128$ ports, and mean packet sizes of $E = 10, 30, 180$ cells. The default values for the simulations are $N = 16$ ports, $M = 1$ source per port, $E = 10$ cells, $l = 70\%$, and $k = 16$ unless specified otherwise. We made two sets of simulations, one with only one source per input port ($M = 1$) and varying number of input ports (N) contained in Subsection 3.1, and a second one with a varying number of input ports and several sources on each input port (varying M) contained in Subsection 3.2. Figures 5-13 show the VC merging buffer size (solid line) compared to the corresponding VP merging buffer size (dashed line). The results serve to compare VP and VC merging. They cannot, however, be used directly to show the required output buffer space in an ATM switch because no flow control has been taken into account. The simulations were carried out for an extremely large number of events such that 95% confidence intervals were very small.

3.1 Simulations with only One Source per Input Port

This subsection includes simulations with only one source per input port but different scenarios. First we investigated the influence of different loads l with nonbursty and bursty traffic arrival processes. We then varied the factor k at different loads and bursty traffic arrival. We also took into account larger mean packet sizes E with bursty traffic and a greater amount of input ports N with nonbursty arrival traffic. Finally we investigated the behavior of our model by varying N at extreme values of k with nonbursty traffic sources.

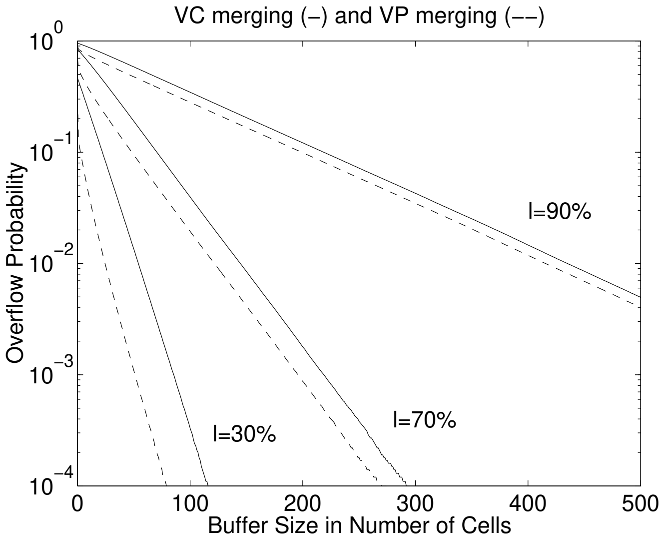

Figure 5: Simulations for l=30%, 70%, 90%, nonbursty arrival process.

Figure 5 shows the results of our simulations for $l = 30\%, 70\%, 90\%$. The difference between the solid and the dashed lines (VC and VP merging for a specific load) is about 19 to 21 cells for $l = 90\%$, about 21 to 23 cells for $l = 70\%$, and about 30 to 37 cells for $l = 30\%$ over some magnitudes of overflow probability. At high loads the output queue contains a large number of cells, which translates to long delays. Therefore, by the time the first cell of a packet is ready for transmission at the output link, the corresponding last cell has most likely arrived and, consequently, the packet pseudo-reassembly has been completed. In this case, therefore, the additional overhead due to pseudo-reassembly is almost negligible. It is also important to note that the workload of today's switches normally lies at high levels of around 70% or 80%. In contrast, at low loads, the first cell may be ready for transmission while the reassembly is in progress. In this case it has to be delayed until the reassembly process has been completed. However, owing to the low load, the number of packets under reassembly is small and, therefore, the additional buffer requirement of VC merging is minimal. The results obtained are in agreement with those presented in [9]. Furthermore, the packet delay corresponding to VC merging approaches that corresponding to VP merging as the load increases.

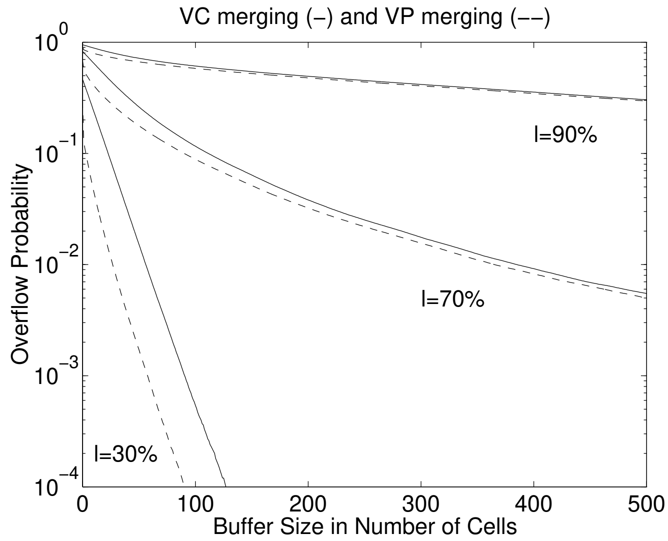

Figure 6: Simulations for l=30%, 70%, 90%, bursty arrival process.

We then made the same simulations with bursty arrival processes. We model the bursty arrival process by a hyperexponential packet arrival process as described in the previous section. The results for $l = 30\%, 70\%, 90\%$ are shown in Figure 6. We see that the buffer requirements for both VC and VP merging grow significantly for high loads. Of course flow control would alleviate this problem to some extent due to the overall load reduction. The additional buffer requirements for VC merging compared to VP merging are minimal even for the case of bursty traffic. In particular, for high loads they become negligible for the reasons given above.

Simulation results were obtained for different values of k and different loads l. By varying k we expected to see an influence on the additional buffer requirement. Surprisingly, only the extreme value $k = 1$ resulted in a big additional buffer requirement for VC merging. It is obvious that VP merging requires almost no output buffer with $k = 1$ as the speed of the switch output port (R_S) is equal to the speed of the output link (R_L). We then tried to determine the critical k for every load factor l considered. The critical value of k is defined as follows: For all values of k larger than the critical value, there is practically no distinction between VC curves and VP curves, whereas for all smaller values of k the curves start becoming distinguishable. We found that the critical k lies close to the extreme value $k = 1$. The range of the critical k

is between 1.1 and 1.3 for $l = 90\%$ and $l = 70\%$, respectively. This means that the critical k becomes larger with lower loads, but it is still far away from the values implemented in today's switches (greater than 2). To substantiate these observations we investigated the critical k for $l = 30\%$, too. In this case the critical value for k is approximately 1.5, which is still much smaller than 2 and thus confirms our theory.

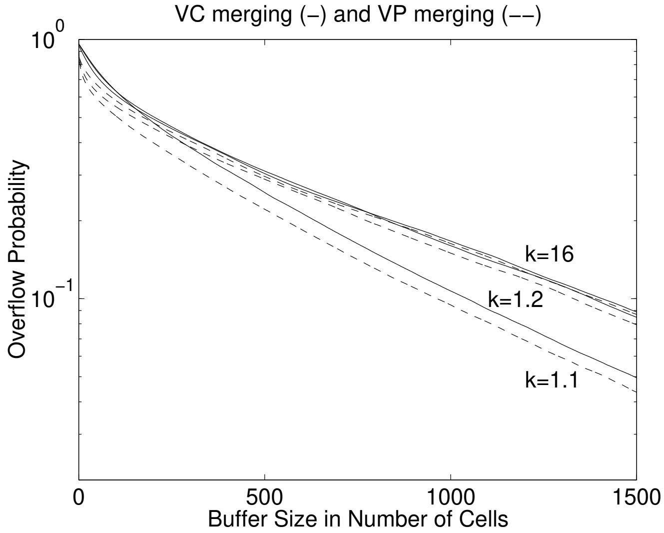

Figure 7: Simulations for l=90%, k=1.1, 1.2, 16, bursty arrival process.

Figure 7 shows the results of our simulations for $k = 1.1, 1.2, 16$ at $l = 90\%$. We observe that all of the curves for the output buffer of VC merging at different values of k lie close together. The value of $k = 1.1$ is the critical one because the corresponding curve starts to show a deviation. The same applies to the curves for VP merging. For values of k greater than the critical one, the additional buffer requirement for VC merging at low overflow probabilities is minimal. However, the difference between VC and VP merging becomes noticeable for values of k less than the critical value.

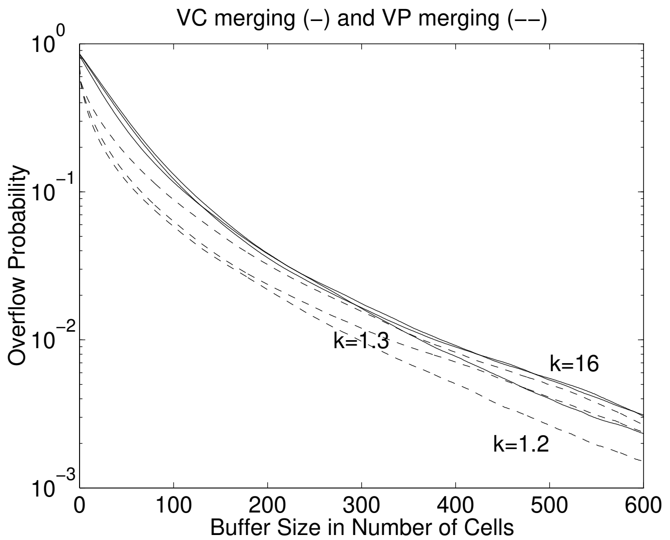

Figure 8: Simulations with l=70%, k=1.2, 1.3, 16, bursty arrival process

Figure 8 shows the results of our simulations for $k = 1.2, 1.3, 16$ at a lower load of $l = 70\%$. In this case, we observe that the different curves for VC and VP merging lie fairly close together with the exception of the curves for $k = 1.2$. This shows that the critical k is slightly larger for $l = 70\%$ (about $k = 1.2$) than for $l = 90\%$ (about $k = 1.1$). Here again, the additional buffer requirements for VC merging at low overflow probabilities become noticeable for values of k less than the critical value.

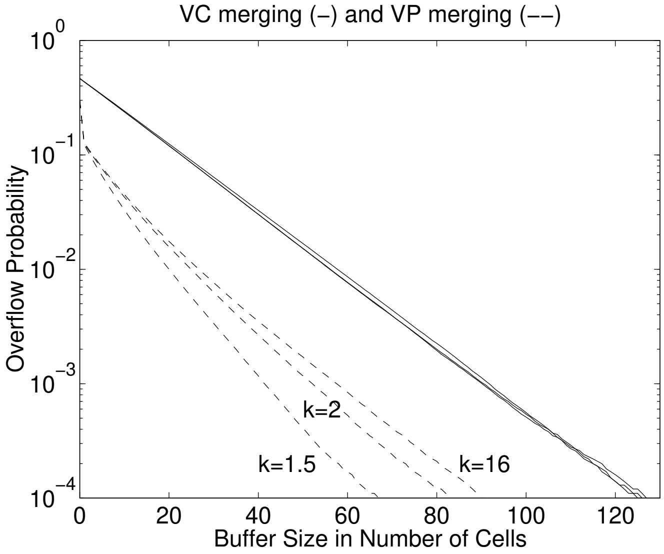

Figure 9: Simulations for l=30%, k=1.5, 2, 16, bursty arrival process.

Figure 9 shows the results of our simulations for $k = 1.5, 2, 16$ at a low load of $l = 30\%$. We observe again the similarity of the curves for VC merging over the entire range of k. The curves for VP merging vary slightly so that the additional buffer space becomes smaller for a larger k, with a critical k at about $k = 1.5$. There is a noticeable additional buffer requirement for VC merging in the entire range of values of k. Furthermore, the additional requirement increases as k decreases.

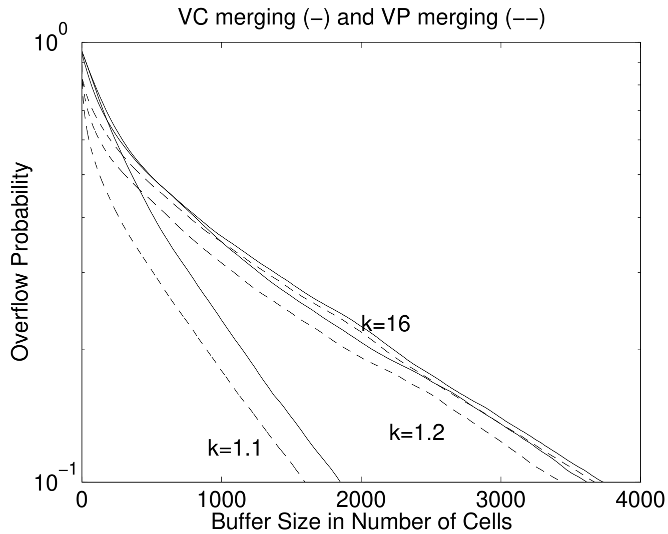

Figure 10: Simulations for l=90%, k=1.1, 1.2, 16, bursty arrival process and E=30.

We then tried to investigate the possible influence of more specific traffic characteristics such as larger packet sizes and increased numbers of sources in a mpt-to-pt connection on the additional requirements of VC merging compared to VP merging. First, we performed simulations for a larger mean packet size of the arrival process ($E = 30$). Figure 10 shows the curves for $l = 90\%$ and $k = 1.1, 1.2, 16$ with a mean packet size of $E = 30$. Compared to Figure 7 we observe a greater difference between the curves for $k = 1.1$ and for $k = 1.2$. It appears that the critical k is shifted to a value slightly larger than $k = 1.1$ (between $k = 1.1$ and $k = 1.2$). Furthermore we see that the mean packet size, which is three times larger, requires an output buffer size that is also three times larger. Moreover, the additional output buffers for VC merging are about three times larger for $E = 30$. Therefore the additional buffer requirement for VC merging appears to grow linearly with the mean packet size. This trend is also verified by our simulations for $E = 180$.

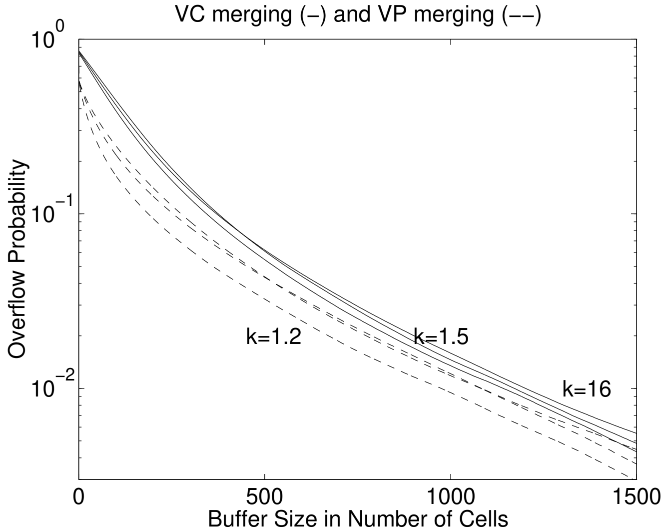

Figure 11: Simulations for l=70%, k=1.2, 1.5, 16, bursty arrival process and E=30.

Figure 11 shows the results of the same simulation for $l = 70\%$, $k = 1.2, 1.5, 16$ and an increased mean packet size of $E = 30$. Compared to Figure 8 we again observe a shift of the critical k from a value of about $k = 1.2$ to a slightly larger value. Concerning the additional buffer requirement for VC merging, the same observations were made as in the case of load $l = 90\%$. This means that, also at this load, as the packet size increases, the additional buffer requirement increases by the same factor.

Finally we performed simulations for a larger N ($N = 64, 128$) to assess the influence of a large number of input ports and corresponding sources associated with one mpt-to-pt connection on the additional buffer requirements of VC merging due to reassembly. An increased number of sources could translate to an increased degree of reassembly. This again would lead to a significantly larger required buffer space for reassembly than for nonreassembly. Figure 12 shows the results for the simulations for $N = 16, 64$ input ports each with $M = 1$ sources, factor $k = 16$ and nonbursty traffic. The results obtained also apply in the case of nonbursty traffic. This is explained by Palm–Khintchine's theorem [17, p. 156], which states that summing up a large number of iid processes (for instance hyperexponential processes as used for our bursty traffic) results in a process of Poisson type (our nonbursty traffic). As our simulation has 64 sources, each with an iid process for the arrival traffic, we are able to apply this theorem and to simulate nonbursty arrival traffic. All of the corresponding curves in Figure 13 lie close together. Concerning VP merging, as N increases, the corresponding curves converge because the aggregated arrival process tends to a Poisson one. For VC merging, the buffer requirement does not increase with the number of sources. This is because increasing the number of sources translates to decreasing the arrival rate per source such that the load at

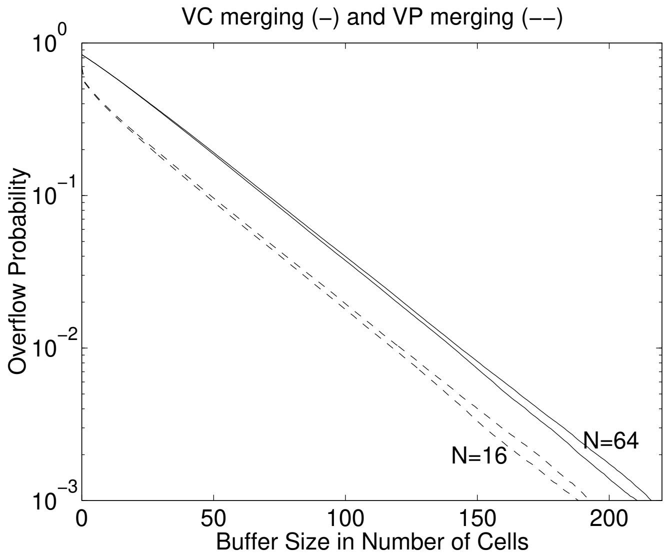

Figure 12: Simulations for N=16, 64, l=70%, k=16, nonbursty arrival process.

the output link remains constant. This shows that our previous simulation results hold also for a larger scenario with a larger number of senders.

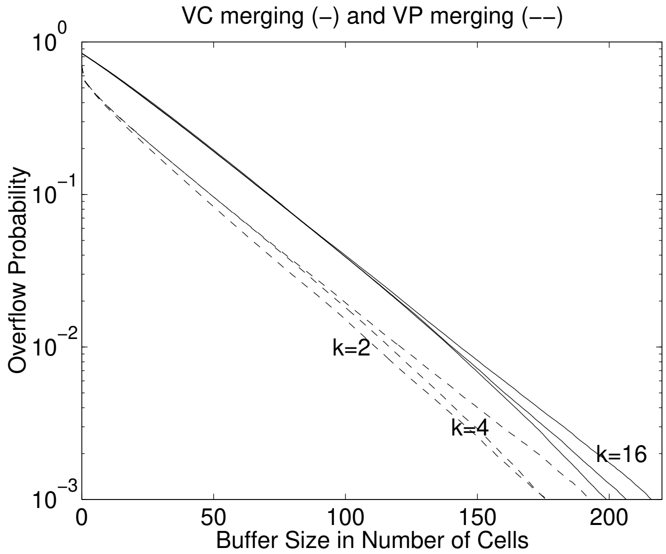

Figure 13: Simulations for N=64, l=70%, k=2, 4, 16, nonbursty arrival process.

We have investigated the impact of varying k given large values of N. Previously we found a critical k of about 1.1 to 1.3 at $N = 16$. Figure 13 shows the results of the simulations with $N = 64$ and $k = 2, 4, 16$ with nonbursty traffic. The different curves for VC and VP merging again lie close together and we see no significant difference between the curves belonging to the values $k = 2$ and $k = 16$. Consequently the critical value of k is smaller than 2. Once again, for values of k greater than the critical value, the additional buffer requirement for VC merging does not increase.

We finally investigated the influence of different values of N at extreme values of k ($k = 1, 16$). We observed that varying N does not have an impact on VP merging because the load at the output port does not change. We also know from the previous simulations that the curves for VC merging will be very similar for $k = 16$. Figure 14 depicts the behavior of the VC

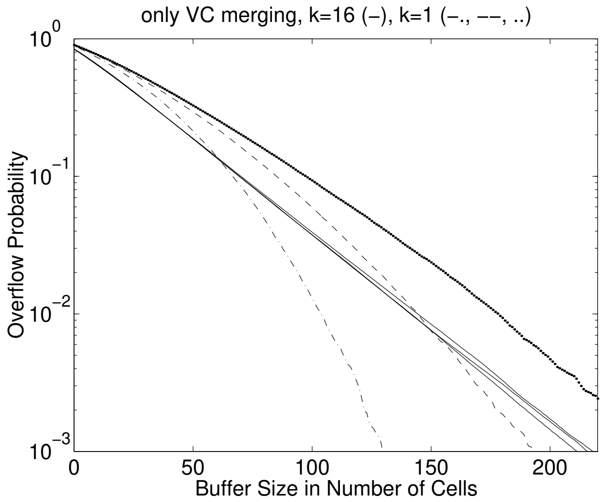

Figure 14: Simulations for N=4 (dashed-dotted), 16 (dashed), 128 (dotted), l=70%, k=1, 16, nonbursty arrival process, only VC merging.

merging output buffer for simulations with $N = 4, 16, 128$ and extreme values of k. The three solid lines representing the simulations with different values of N but constant $k = 16$ show almost no difference. But we do observe a difference for the curves with $k = 1$. The dashed-dotted line corresponds to $N = 4$, the dashed line to $N = 16$, and the dotted line to $N = 128$. For each value of N, the two curves corresponding to $k = 1$ and $k = 16$ clearly identify two distinct regions. In the first region, the probability that the queue length exceeds a certain value is greater for $k = 1$ than for $k = 16$, whereas in the second region the probability is smaller for $k = 1$ than for $k = 16$. This behavior can be explained by noting that low values of k result in a slow filling process of the output port queues. Consequently, because of the large gap between successively arriving cells belonging to one packet, there is always a small number of cells in the process of being reassembled and, therefore, the reassembly buffer occupancy is greater than the one corresponding to higher values of k. For a higher value of k, the amount of cells in the process of being reassembled is rather large and, therefore, the reassembly buffer occupancy is greater than the one corresponding to smaller values of k. Note also that the size of the reassembly buffers is a function of N, but only for low values of k (k below the critical $k = 1.3$). For values of k higher than the critical one, the size of the reassembly buffers does not depend on N nor on k. The phenomenon of the two distinct regions appears also on the curves depicted in Figures 7, 8, and 10 with a bursty arrival process. For small buffer sizes (less than about 100 cells), the curves for VC merging corresponding to values of k smaller than the critical one have a greater overflow probability than those corresponding to high values of k. After the crossover, this tendency becomes reversed for large buffer sizes.

3.2 Simulations with Multiple Sources per Input Port

The second set of our simulations investigates the effect of varying the amount of sources at one input port. We investigated the relationship between the number of sources at one input port (M) and the size of the reassembly buffers.

First, we made simulations with a constant product of $N \times M = 128$. Our aim was to observe the difference between the two extreme cases; the case of having numerous sources producing traffic at one input port and the case of having the same amount of sources but each one arriving on a different input port. For VP merging there is no difference because the load l at the output port is fixed. Figure 15 shows the results for simulations with (N, M)=(4,

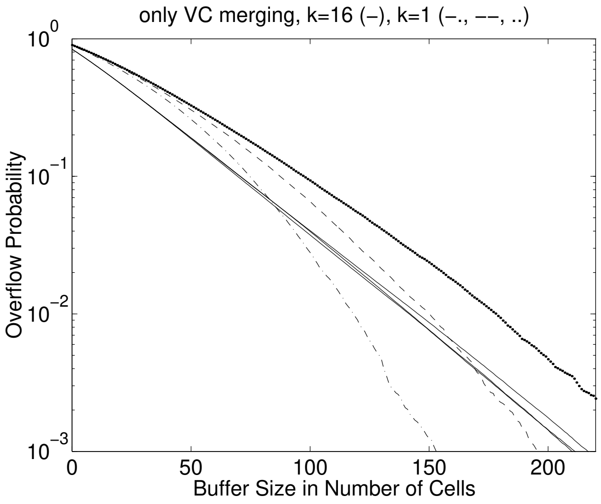

Figure 15: Simulations for $(N,\ M)$=(4, 32) (dashed-dotted), (16, 8) (dashed), (128, 1) (dotted), l=70%, k=1, 16, nonbursty arrival process, only VC merging.

32), (16, 8), and (128, 1) and extreme values of $k = 1$ and $k = 16$. Again, the dashed-dotted line corresponds to (N, M)=(4, 32), the dashed line to (N, M)=(16, 8), and the dotted line to (N, M)=(128, 1), all with $k = 1$. There is no difference between the three solid lines of $k = 16$. This figure shows almost the same behavior as Figure 14 with the same simulations (varying N of 4, 16, and 128) but constant $M = 1$. This means that the size of the reassembly buffer is only a function of N but not of M.

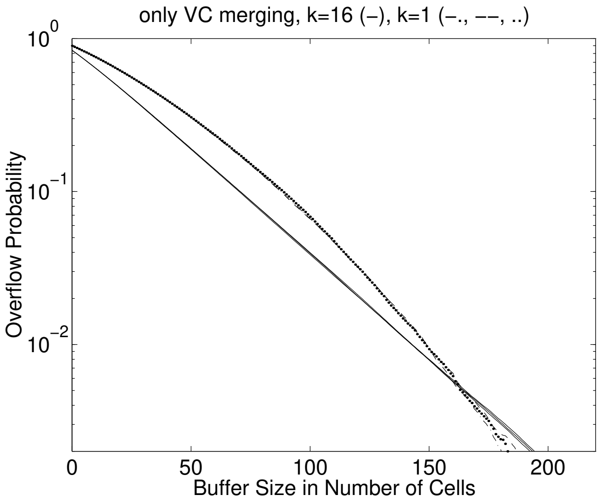

Figure 16: Simulations for N=16, M=4 (dashed-dotted), 16 (dashed), 64 (dotted), l=70%, k=1, 16, nonbursty arrival process, only VC merging.

To verify this observation we made simulations with constant $N = 4$ and $N = 16$ and varying $M = 4, 16, 32, 64$, again with the extreme values of $k = 1$ and $k = 16$. Figure 16 depicts the results for a constant N ($N = 16$) and varying M ($M = 4, 16, 64$). The solid lines for the different simulations with $k = 16$ again show no difference. But this time also the dashed-dotted, dashed, and dotted lines representing simulations with different values of M at $k = 1$ show no difference. This implies that for $N = 16$ the size of the reassembly buffer is not a function of the number of sources per input port. This is because the sources produce a constant load for the output link. The load at every input port is $1/N$ of the load at the output port. Consequently, as N increases, the load at an input port decreases. With this low load per input port the probability of cell interleaving at the input port is also very low for values of $N > 4$. Therefore, there is never more than one packet of any input port involved in reassembly at the output port. Also note that the above observations are independent of the number of sources per input port (M), therefore, the required buffer size does not depend on M, it only depends on the value of N (for $N > 4$).

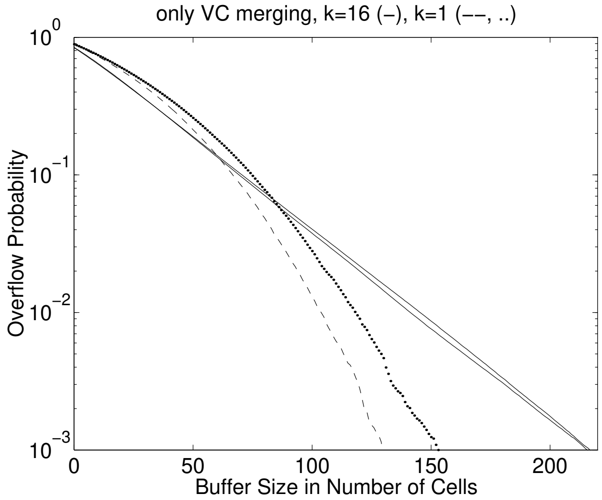

Figure 17: Simulations for N=4, M=1 (dashed), 32 (dotted), l=70%, k=1, 16, nonbursty arrival process, only VC merging.

Let us now turn our attention to small values of N. Simulations were conducted with $N = 4$ and varying M ($M = 1, 32$). Figure 17 depicts the corresponding curves. We observe the behavior we expected. The curves for $M = 1$ and $M = 32$ show no difference for $k = 16$ but they show a difference for $k = 1$ because of the small number of input ports. For a small number of input ports, the load at every input port is high enough for there to be a significant probability of cell interleaving at the input port. This interleaving gives rise to a different cell arrival process, which leads to different curves for small values of k (k smaller than the critical $k = 1.3$). Therefore, the size of the reassembly buffers at the output port is a function of M but only for small values of N and small values of k (k smaller than the critical value).

4 Summary and Conclusions

VC merging is likely to become the method of choice to implement mpt-to-pt connections in ATM networks. Because of the cell interleaving problem created by VC merging, pseudo-reassembly has to be performed in the merging points. The effect of pseudo-reassembly has been investigated assuming an output queue switch architecture. The results obtained demonstrate that, at high loads and for arbitrary arrival processes, the implementation of VC merging in the switches will not require much additional buffer at the output queues of the switches. In contrast, at low loads, additional buffer is required but this is minimal. Furthermore, it was found that the additional buffer requirement for VC merging is proportional to the average packet size. Consequently, large packet sizes can result in large reassembly buffer requirements. We further investigated the effect of the speed ratio between switch output port and output link and came to the conclusion that for sufficiently large speed ratio values ($k > 2$) the output buffer requirement for VP and VC merging remain the same, respectively. We found a critical k that grows with decreasing utilization and with increasing mean packet sizes of the arrival traffic. But it always remains between 1.1 and 1.3 for high utilization of 70% and 90%. Finally we found that if N is greater than 4, and k is greater than the critical value, and the utilization is high (higher than about 60%), the reassembly buffer size requirement for VC merging is independent of the number of sources at every input port (M) as well as the number of input ports (N).

References

[1] J. Calvignac, C. Basso, P. Droz, D. Dykeman: Dynamic IDentifier Assignment (DIDA) for Merged ATM Connections, ATM Forum, 97-0504, July 1997.

[2] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, A. Viswanathan: A Framework for Multiprotocol Label Switching, Internet Draft <draft-ietf-mpls-framework-02.txt>, November 1997.

[3] J. Heinanen: Multipoint-to-Point VCs, ATM Forum, 97-0261, April-May 1997.

[4] I. Widjaja, S. Wright, A. Chatterjee: Interworking of VP-Merge, VC-Merge, and Non-Merge ATM Switches in a Multipoint-to-Point Environment, ATM Forum, 97-0748, September 1997.

[5] I. Iliadis, W.E. Denzel: Analysis of Packet Switches with Input and Output Queuing, *IEEE Transactions on Communications*, Vol. 41, No. 5, 1993, pp. 731-740.

[6] W.E. Denzel, A.P.J. Engbersen, I. Iliadis: A Flexible Shared-Buffer Switch for ATM at Gb/s Rates, *Comp. Networks ISDN Systems*, Vol. 27, No. 4, 1995, pp. 611-624.

[7] Y. Oie, M. Murata, K. Kubota, H. Miyahara: Effect of Speedup in Nonblocking Packet Switch. Proc. Int'l Conf. on Communications "ICC '89," Boston, MA (USA), June 1989, pp. 410-414.

[8] H.J. Cha, D.E. Smith: A Shared-Memory Virtual Channel Queue for ATM Broadband Terminal Adaptors, *Int'l J. Digital and Analog Communication Systems*, Vol. 5, 1992, pp. 29-37.

[9] I. Widjaja, A.I. Elwalid: Performance Issues in VC-Merge Capable Switches for IP Over ATM, ATM Forum, 97-0675, July 1997.

[10] J. Heinanen:Multiprotocol Encapsulation Over ATM Adaption Layer 5, RFC 1483, July 1993.

[11] http://www.nlanr.net/NA/Learn/packetsizes.html

[12] K. Thompson, G.J. Miller, R. Wilder: Wide-Area Internet Traffic Patterns and Characteristics, *IEEE Network* , November/December 1997.

[13] J. Mogul, S. Deering: Path MTU Discovery, RFC 1191, November 1990.

[14] M. Laubach: Classical IP and ARP Over ATM, RFC 1577, January 1994.

[15] R. Atkinson: Default IP MTU for Use Over ATM AAL5, RFC 1626, May 1994.

[16] D. Piscitello, J. Lawrence: The Transmission of IP Datagrams Over the SMDS Service, RFC 1209, March 1991.

[17] D.P. Heyman, M.J. Sobel: *Stochastic Models in Operations Research*, Vol. 1, McGraw-Hill, New York 1982.