Network Working Group
Request For Comments: 2682
Category: Informational

I. Widjaja
Fujitsu Network Communications
A. Elwalid
Bell Labs, Lucent Technologies

September 1999

Performance Issues in VC-Merge Capable ATM LSRs

Status of this Memo

Copyright Notice

Abstract

VC merging allows many routes to be mapped to the same VC label, thereby providing a scalable mapping method that can support thousands of edge routers. VC merging requires reassembly buffers so that cells belonging to different packets intended for the same destination do not interleave with each other. This document investigates the impact of VC merging on the additional buffer required for the reassembly buffers and other buffers. The main result indicates that VC merging incurs a minimal overhead compared to non-VC merging in terms of additional buffering. Moreover, the overhead decreases as utilization increases, or as the traffic becomes more bursty.

1.0 Introduction

Recently some radical proposals to overhaul the legacy router architectures have been presented by several organizations, notably the Ipsilon's IP switching [1], Cisco's Tag switching [2], Toshiba's CSR [3], IBM's ARIS [4], and IETF's MPLS [5]. Although the details of their implementations vary, there is one fundamental concept that is shared by all these proposals: map the route information to short fixed-length labels so that next-hop routers can be determined by direct indexing.

Although any layer 2 switching mechanism can in principle be applied, the use of ATM switches in the backbone network is believed to be a very attractive solution since ATM hardware switches have been extensively studied and are widely available in many different

architectures. In this document, we will assume that layer 2 switching uses ATM technology. In this case, each IP packet may be segmented to multiple 53-byte cells before being switched. Traditionally, AAL 5 has been used as the encapsulation method in data communications since it is simple, efficient, and has a powerful error detection mechanism. For the ATM switch to forward incoming cells to the correct outputs, the IP route information needs to be mapped to ATM labels which are kept in the VPI or/and VCI fields. The relevant route information that is stored semi-permanently in the IP routing table contains the tuple (destination, next-hop router). The route information changes when the network state changes and this typically occurs slowly, except during transient cases. The word "destination" typically refers to the destination network (or CIDR prefix), but can be readily generalized to (destination network, QoS), (destination host, QoS), or many other granularities. In this document, the destination can mean any of the above or other possible granularities.

Several methods of mapping the route information to ATM labels exist. In the simplest form, each source-destination pair is mapped to a unique VC value at a switch. This method, called the non-VC merging case, allows the receiver to easily reassemble cells into respective packets since the VC values can be used to distinguish the senders. However, if there are n sources and destinations, each switch is potentially required to manage O(n^2) VC labels for full-meshed connectivity. For example, if there are 1,000 sources/destinations, then the size of the VC routing table is on the order of 1,000,000 entries. Clearly, this method is not scalable to large networks. In the second method called VP merging, the VP labels of cells that are intended for the same destination would be translated to the same outgoing VP value, thereby reducing VP consumption downstream. For each VP, the VC value is used to identify the sender so that the receiver can reconstruct packets even though cells from different packets are allowed to interleave. Each switch is now required to manage O(n) VP labels - a considerable saving from O(n^2). Although the number of label entries is considerably reduced, VP merging is limited to only 4,096 entries at the network-to-network interface. Moreover, VP merging requires coordination of the VC values for a given VP, which introduces more complexity. A third method, called VC merging, maps incoming VC labels for the same destination to the same outgoing VC label. This method is scalable and does not have the space constraint problem as in VP merging. With VC merging, cells for the same destination is indistinguishable at the output of a switch. Therefore, cells belonging to different packets for the same destination cannot interleave with each other, or else the receiver will not be able to reassemble the packets. With VC merging, the boundary between two adjacent packets are identified by the "End-of-Packet" (EOP) marker used by AAL 5.

It is worthy to mention that cell interleaving may be allowed if we use the AAL 3/4 Message Identifier (MID) field to identify the sender uniquely. However, this method has some serious drawbacks as: 1) the MID size may not be sufficient to identify all senders, 2) the encapsulation method is not efficient, 3) the CRC capability is not as powerful as in AAL 5, and 4) AAL 3/4 is not as widely supported as AAL 5 in data communications.

Before VC merging with no cell interleaving can be qualified as the most promising approach, two main issues need to be addressed. First, the feasibility of an ATM switch that is capable of merging VCs needs to be investigated. Second, there is widespread concern that the additional amount of buffering required to implement VC merging is excessive and thus making the VC-merging method impractical. Through analysis and simulation, we will dispel these concerns in this document by showing that the additional buffer requirement for VC merging is minimal for most practical purposes. Other performance related issues such as additional delay due to VC merging will also be discussed.

2.0 A VC-Merge Capable MPLS Switch Architecture

In principle, the reassembly buffers can be placed at the input or output side of a switch. If they are located at the input, then the switch fabric has to transfer all cells belonging to a given packet in an atomic manner since cells are not allowed to interleave. This requires the fabric to perform frame switching which is not flexible nor desirable when multiple QoSs need to be supported. On the other hand, if the reassembly buffers are located at the output, the switch fabric can forward each cell independently as in normal ATM switching. Placing the reassembly buffers at the output makes an output-buffered ATM switch a natural choice.

We consider a generic output-buffered VC-merge capable MPLS switch with VCI translation performed at the output. Other possible architectures may also be adopted. The switch consists of a non-blocking cell switch fabric and multiple output modules (OMs), each is associated with an output port. Each arriving ATM cell is appended with two fields containing an output port number and an input port number. Based on the output port number, the switch fabric forwards each cell to the correct output port, just as in normal ATM switches. If VC merging is not implemented, then the OM consists of an output buffer. If VC merging is implemented, the OM contains a number of reassembly buffers (RBs), followed by a merging unit, and an output buffer. Each RB typically corresponds to an incoming VC value. It is important to note that each buffer is a logical buffer, and it is envisioned that there is a common pool of memory for the reassembly buffers and the output buffer.

The purpose of the RB is to ensure that cells for a given packet do not interleave with other cells that are merged to the same VC. This mechanism (called store-and-forward at the packet level) can be accomplished by storing each incoming cell for a given packet at the RB until the last cell of the packet arrives. When the last cell arrives, all cells in the packet are transferred in an atomic manner to the output buffer for transmission to the next hop. It is worth pointing out that performing a cut-through mode at the RB is not recommended since it would result in wastage of bandwidth if the subsequent cells are delayed. During the transfer of a packet to the output buffer, the incoming VCI is translated to the outgoing VCI by the merging unit. To save VC translation table space, different incoming VCIs are merged to the same outgoing VCI during the translation process if the cells are intended for the same destination. If all traffic is best-effort, full-merging where all incoming VCs destined for the same destination network are mapped to the same outgoing VC, can be implemented. However, if the traffic is composed of multiple classes, it is desirable to implement partial merging, where incoming VCs destined for the same (destination network, QoS) are mapped to the same outgoing VC.

Regardless of whether full merging or partial merging is implemented, the output buffer may consist of a single FIFO buffer or multiple buffers each corresponding to a destination network or (destination network, QoS). If a single output buffer is used, then the switch essentially tries to emulate frame switching. If multiple output buffers are used, VC merging is different from frame switching since cells of a given packet are not bound to be transmitted back-to-back. In fact, fair queueing can be implemented so that cells from their respective output buffers are served according to some QoS requirements. Note that cell-by-cell scheduling can be implemented with VC merging, whereas only packet-by-packet scheduling can be implemented with frame switching. In summary, VC merging is more flexible than frame switching and supports better QoS control.

3.0 Performance Investigation of VC Merging

This section compares the VC-merging switch and the non-VC merging switch. The non-VC merging switch is analogous to the traditional output-buffered ATM switch, whereby cells of any packets are allowed to interleave. Since each cell is a distinct unit of information, the non-VC merging switch is a work-conserving system at the cell level. On the other hand, the VC-merging switch is non-work conserving so its performance is always lower than that of the non-VC merging switch. The main objective here is to study the effect of VC merging on performance implications of MPLS switches such as additional delay, additional buffer, etc., subject to different traffic conditions.

In the simulation, the arrival process to each reassembly buffer is an independent ON-OFF process. Cells within an ON period form a single packet. During an OFF periof, the slots are idle. Note that the ON-OFF process is a general process that can model any traffic process.

3.1 Effect of Utilization on Additional Buffer Requirement

We first investigate the effect of switch utilization on the additional buffer requirement for a given overflow probability. To carry the comparison, we analyze the VC-merging and non-VC merging case when the average packet size is equal to 10 cells, using geometrically distributed packet sizes and packet interarrival times, with cells of a packet arriving contiguously (later, we consider other distributions). The results show, as expected, the VC-merging switch requires more buffers than the non-VC merging switch. When the utilization is low, there may be relatively many incomplete packets in the reassembly buffers at any given time, thus wasting storage resource. For example, when the utilization is 0.3, VC merging requires an additional storage of about 45 cells to achieve the same overflow probability. However, as the utilization increases to 0.9, the additional storage to achieve the same overflow probability drops to about 30 cells. The reason is that when traffic intensity increases, the VC-merging system becomes more work-conserving.

It is important to note that ATM switches must be dimensioned at high utilization value (in the range of 0.8-0.9) to withstand harsh traffic conditions. At the utilization of 0.9, a VC-merge ATM switch requires a buffer of size 976 cells to provide an overflow probability of 10^{-5}, whereas an non-VC merge ATM switch requires a buffer of size 946. These numbers translate the additional buffer requirement for VC merging to about 3% - hardly an additional buffering cost.

3.2 Effect of Packet Size on Additional Buffer Requirement

We now vary the average packet size to see the impact on the buffer requirement. We fix the utilization to 0.5 and use two different average packet sizes; that is, B=10 and B=30. To achieve the same overflow probability, VC merging requires an additional buffer of about 40 cells (or 4 packets) compared to non-VC merging when B=10. When B=30, the additional buffer requirement is about 90 cells (or 3 packets). As expected, the additional buffer requirement in terms of cells increases as the packet size increases. However, the additional buffer requirement is roughly constant in terms of packets.

3.3 Additional Buffer Overhead Due to Packet Reassembly

There may be some concern that VC merging may require too much buffering when the number of reassembly buffers increases, which would happen if the switch size is increased or if cells for packets going to different destinations are allowed to interleave. We will show that the concern is unfounded since buffer sharing becomes more efficient as the number of reassembly buffers increases.

To demonstrate our argument, we consider the overflow probability for VC merging for several values of reassembly buffers (N); i.e., N=4, 8, 16, 32, 64, and 128. The utilization is fixed to 0.8 for each case, and the average packet size is chosen to be 10. For a given overflow probability, the increase in buffer requirement becomes less pronounced as N increases. Beyond a certain value (N=32), the increase in buffer requirement becomes insignificant. The reason is that as N increases, the traffic gets thinned and eventually approaches a limiting process.

3.4 Effect of Interarrival time Distribution on Additional Buffer

We now turn our attention to different traffic processes. First, we use the same ON period distribution and change the OFF period distribution from geometric to hypergeometric which has a larger Square Coefficient of Variation (SCV), defined to be the ratio of the variance to the square of the mean. Here we fix the utilization at 0.5. As expected, the switch performance degrades as the SCV increases in both the VC-merging and non-VC merging cases. To achieve a buffer overflow probability of 10^{-4}, the additional buffer required is about 40 cells when SCV=1, 26 cells when SCV=1.5, and 24 cells when SCV=2.6. The result shows that VC merging becomes more work-conserving as SCV increases. In summary, as the interarrival time between packets becomes more bursty, the additional buffer requirement for VC merging diminishes.

3.5 Effect of Internet Packets on Additional Buffer Requirement

Up to now, the packet size has been modeled as a geometric distribution with a certain parameter. We modify the packet size distribution to a more realistic one for the rest of this document. Since the initial deployment of VC-merge capable ATM switches is likely to be in the core network, it is more realistic to consider the packet size distribution in the Wide Area Network. To this end, we refer to the data given in [6]. The data collected on Feb 10, 1996, in FIX-West network, is in the form of probability mass function versus packet size in bytes. Data collected at other dates closely resemble this one.

The distribution appears bi-modal with two big masses at 40 bytes (about a third) due to TCP acknowledgment packets, and 552 bytes (about 22 percent) due to Maximum Transmission Unit (MTU) limitations in many routers. Other prominent packet sizes include 72 bytes (about 4.1 percent), 576 bytes (about 3.6 percent), 44 bytes (about 3 percent), 185 bytes (about 2.7 percent), and 1500 bytes (about 1.5 percent) due to Ethernet MTU. The mean packet size is 257 bytes, and the variance is 84,287 bytes^2. Thus, the SCV for the Internet packet size is about 1.1.

To convert the IP packet size in bytes to ATM cells, we assume AAL 5 using null encapsulation where the additional overhead in AAL 5 is 8 bytes long [7]. Using the null encapsulation technique, the average packet size is about 6.2 ATM cells.

We examine the buffer overflow probability against the buffer size using the Internet packet size distribution. The OFF period is assumed to have a geometric distribution. Again, we find that the same behavior as before, except that the buffer requirement drops with Internet packets due to smaller average packet size.

3.6 Effect of Correlated Interarrival Times on Additional Buffer Requirement

To model correlated interarrival times, we use the DAR(p) process (discrete autoregressive process of order p) [8], which has been used to accurately model video traffic (Star Wars movie) in [9]. The DAR(p) process is a p-th order (lag-p) discrete-time Markov chain. The state of the process at time n depends explicitly on the states at times (n-1), ..., (n-p).

We examine the overflow probability for the case where the interarrival time between packets is geometric and independent, and the case where the interarrival time is geometric and correlated to the previous one with coefficient of correlation equal to 0.9. The empirical distribution of the Internet packet size from the last section is used. The utilization is fixed to 0.5 in each case. Although, the overflow probability increases as p increases, the additional amount of buffering actually decreases for VC merging as p, or equivalently the correlation, increases. One can easily conclude that higher-order correlation or long-range dependence, which occurs in self-similar traffic, will result in similar qualitative performance.

3.7 Slow Sources

The discussions up to now have assumed that cells within a packet arrive back-to-back. When traffic shaping is implemented, adjacent cells within the same packet would typically be spaced by idle slots. We call such sources as "slow sources". Adjacent cells within the same packet may also be perturbed and spaced as these cells travel downstream due to the merging and splitting of cells at preceding nodes.

Here, we assume that each source transmits at the rate of r_s ($0 < r_s < 1$), in units of link speed, to the ATM switch. To capture the merging and splitting of cells as they travel in the network, we will also assume that the cell interarrival time within a packet is randomly perturbed. To model this perturbation, we stretch the original ON period by $1/r_s$, and flip a Bernoulli coin with parameter r_s during the stretched ON period. In other words, a slot would contain a cell with probability r_s, and would be idle with probability $1-r_s$ during the ON period. By doing so, the average packet size remains the same as r_s is varied. We simulated slow sources on the VC-merge ATM switch using the Internet packet size distribution with $r_s=1$ and $r_s=0.2$. The packet interarrival time is assumed to be geometrically distributed. Reducing the source rate in general reduces the stresses on the ATM switches since the traffic becomes smoother. With VC merging, slow sources also have the effect of increasing the reassembly time. At utilization of 0.5, the reassembly time is more dominant and causes the slow source (with $r_s=0.2$) to require more buffering than the fast source (with $r_s=1$). At utilization of 0.8, the smoother traffic is more dominant and causes the slow source (with $r_s=0.2$) to require less buffering than the fast source (with $r_s=1$). This result again has practical consequences in ATM switch design where buffer dimensioning is performed at reasonably high utilization. In this situation, slow sources only help.

3.8 Packet Delay

It is of interest to see the impact of cell reassembly on packet delay. Here we consider the delay at one node only; end-to-end delays are subject of ongoing work. We define the delay of a packet as the time between the arrival of the first cell of a packet at the switch and the departure of the last cell of the same packet. We study the average packet delay as a function of utilization for both VC-merging and non-VC merging switches for the case $r_s=1$ (back-to-back cells in a packet). Again, the Internet packet size distribution is used to adopt the more realistic scenario. The interarrival time of packets is geometrically distributed. Although the difference in the worst-case delay between VC-merging and non-VC merging can be theoretically very large, we consistently observe that the difference in average

delays of the two systems to be consistently about one average packet time for a wide range of utilization. The difference is due to the average time needed to reassemble a packet.

To see the effect of cell spacing in a packet, we again simulate the average packet delay for r_s=0.2. We observe that the difference in average delays of VC merging and non-VC merging increases to a few packet times (approximately 20 cells at high utilization). It should be noted that when a VC-merge capable ATM switch reassembles packets, in effect it performs the task that the receiver has to do otherwise. From practical point-of-view, an increase in 20 cells translates to about 60 micro seconds at OC-3 link speed. This additional delay should be insignificant for most applications.

4.0 Security Considerations

There are no security considerations directly related to this document since the document is concerned with the performance implications of VC merging. There are also no known security considerations as a result of the proposed modification of a legacy ATM LSR to incorporate VC merging.

5.0 Discussion

This document has investigated the impacts of VC merging on the performance of an ATM LSR. We experimented with various traffic processes to understand the detailed behavior of VC-merge capable ATM LSRs. Our main finding indicates that VC merging incurs a minimal overhead compared to non-VC merging in terms of additional buffering. Moreover, the overhead decreases as utilization increases, or as the traffic becomes more bursty. This fact has important practical consequences since switches are dimensioned for high utilization and stressful traffic conditions. We have considered the case where the output buffer uses a FIFO scheduling. However, based on our investigation on slow sources, we believe that fair queueing will not introduce a significant impact on the additional amount of buffering. Others may wish to investigate this further.

6.0 Acknowledgement

The authors thank Debasis Mitra for his penetrating questions during the internal talks and discussions.

7.0 References

[1] P. Newman, Tom Lyon and G. Minshall, "Flow Labelled IP: Connectionless ATM Under IP", in Proceedings of INFOCOM'96, San-Francisco, April 1996.

[2] Rekhter,Y., Davie, B., Katz, D., Rosen, E. and G. Swallow, "Cisco Systems' Tag Switching Architecture Overview", RFC 2105, February 1997.

[3] Katsube, Y., Nagami, K. and H. Esaki, "Toshiba's Router Architecture Extensions for ATM: Overview", RFC 2098, February 1997.

[4] A. Viswanathan, N. Feldman, R. Boivie and R. Woundy, "ARIS: Aggregate Route-Based IP Switching", Work in Progress.

[5] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow and A. Viswanathan, "A Framework for Multiprotocol Label Switching", Work in Progress.

[6] WAN Packet Size Distribution, http://www.nlanr.net/NA/Learn/packetsizes.html.

[7] Heinanen, J., "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 1483, July 1993.

[8] P. Jacobs and P. Lewis, "Discrete Time Series Generated by Mixtures III: Autoregressive Processes (DAR(p))", Technical Report NPS55-78-022, Naval Postgraduate School, 1978.

[9] B.K. Ryu and A. Elwalid, "The Importance of Long-Range Dependence of VBR Video Traffic in ATM Traffic Engineering", ACM SigComm'96, Stanford, CA, pp. 3-14, August 1996.

Authors' Addresses

Indra Widjaja
Fujitsu Network Communications
Two Blue Hill Plaza
Pearl River, NY 10965, USA

Phone: 914 731-2244
EMail: indra.widjaja@fnc.fujitsu.com

Anwar Elwalid
Bell Labs, Lucent Technologies
600 Mountain Ave, Rm 2C-324
Murray Hill, NJ 07974, USA

Phone: 908 582-7589
EMail: anwar@lucent.com

9. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.