BUFFER REQUIREMENTS IN ATM NETWORKS WITH LEAKY BUCKETS

Ljiljana Trajković and **Shlomo Halfin**

Bellcore
445 South Street
Morristown, NJ 07962
ljilja@bellcore.com sh@bellcore.com

Abstract

For an ATM network with guaranteed Quality of Service, we present traffic scenarios showing that the buffer size allocation requirements at various hops through the network may grow unexpectedly high for a zero loss Quality of Service. This occurs even if one uses Peak Load Allocation and the traffic patterns entering the network are well regulated and conform to the agreed rates.

I. Introduction

High speed packet networks will support diverse traffic such as voice, data, image, and real time video. In such a heterogeneous environment, different services will demand a range of Quality of Services (QoS) [1]. Some applications require no packet loss, while others are sensitive to packet delay or to packet delay-jitter. One of the goals of a good admission control scheme for ATM networks is to guarantee contracted performance for heterogeneous services while optimally using network resources (bandwidth) [2], [3], [4], [5], [6], [7], [8], [9], [10].

An ATM network with guaranteed QoS may support Virtual Channels (VCs) of various bandwidth allocations. We assume that the sources on the outskirts of the network have pre-shaped traffic patterns that conform to the Connection Traffic Descriptor specified by the Generic Cell Rate Algorithm (GCRA) [11], [12] or Leaky (or Token) Bucket [13]. One ATM layer QoS that a network will provide is guaranteed service. This service will guarantee agreed upon low losses for customers who adhere to agreed Peak Cell Rates (PCR) defined by the GCRA. We show, by constructing possible traffic scenarios, that the buffer size allocation requirements at various hops through the network may grow unexpectedly high for a zero loss QoS even if the traffic entering the network is well regulated and conforms to the agreed PCR, and Peak Load Allocation is used.

II. Buffer Size Allocation Requirements

ATM networks consist of various Network Elements: MUXs. DMUXs, and switches, that might have input or output buffers. The buffering of arriving packets can be performed per VC (or per Virtual Path) or per link. These mechanisms will determine buffer size allocations requirements for zero packet loss. In our examples we consider switching nodes (for simplicity named MUXs and DMUXs) with output (shared) buffering per outgoing link, as shown in Figure 1(a) and (b). (Note that certain switch architecture possess both input and output buffers, as well as internal buffers designed as recirculating queues of cells that are being processed.) While packets belonging to distinct VCs can be distinguished based on their VC field in the ATM cell header, they are buffered as they arrive on a link together with packets with distinct VC fields. A different buffering strategy, such as one with per VC output buffering in a switching node, will alleviate the problem we describe, but will be more complex to implement. Furthermore, distributed buffering always requires larger buffers than a common buffering scheme.

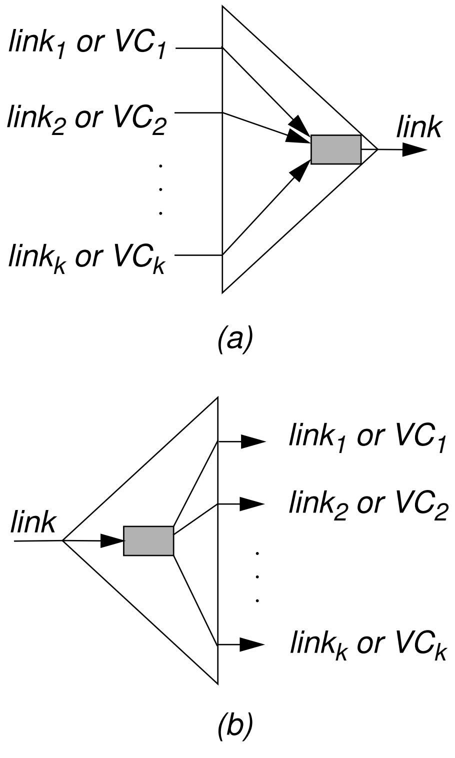

Figure 1: Buffer strategies at a service node: Output buffering per link for a (a) MUX node and (b) DMUX node.

An ATM network with guaranteed QoS may support VCs of various bandwidth allocations. Various VCs may be allocated a portion of the total ATM bandwidth (149.76 Mb/s) and may have several hops where MUX/DMUXing of ATM cells is performed. Network Elements such as MUXs and DMUXs may, therefore, process VCs of various speeds. To illustrate required buffer space per VC at each hop we consider a scenario with a MUX/DMUX scheme and multiple VCs with distinct bit rates.

Consider two types of traffic streams shown in Figure 2(a): a fast stream A arriving to a MUX point at rate 74.88 Mb/s, and five slow streams B arriving at rate 14.976 Mb/s each, as shown in Figure 2(b). We assume a FIFO or a round robin service at the input ports. We also assume that the cells belonging to the slow VCs arrive simultaneously and are served first. The cells belonging to the fast VC that arrive between the successive arrivals of the cells from slow VCs are served after them. The output stream L shown in Figure 2(b) will then have the pattern shown in Figure 3. One can see the clustering of cells belonging to the fast VC.

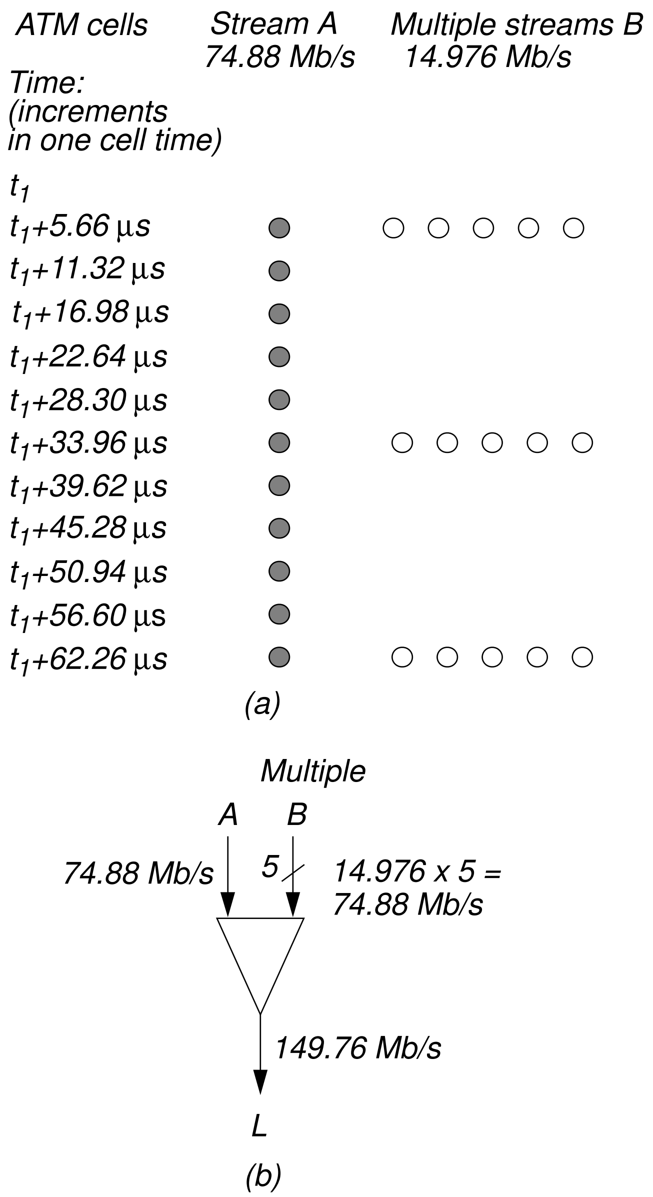

Figure 2: (a) Fast and slow ATM streams arriving to a MUX point. (b) MUX point with one A stream and multiple (five) B streams. Time scale arbitrarily starts at t_1 in increments of one cell time for stream A (5.66 μsec).

Stream C consisting of cells from stream A, after being DMUXed from stream L, is shown in Figure 4(a). Streams C and D have the bandwidth allocations of the original fast streams A and the total of streams B, respectively. Since we deal with a peak allocation service, the outgoing link is reserved for 149.76 Mb/s in each case, as was done at the beginning of the MUX/DMUX process. Note that due to the MUXing and DMUXing of VCs with different bandwidth allocations, the cells of stream C need to be buffered. In the example shown in Figure 4(b), up to three cells of each stream need to be stored in the MUX/DMUX point before being sent to the output link.

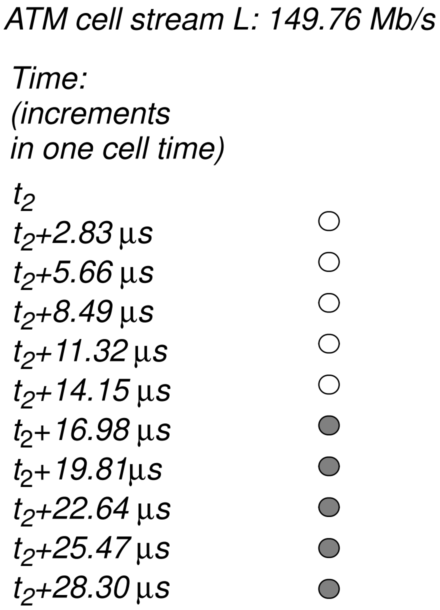

Figure 3: Traffic stream L after the MUX point. Time scale starts at t_2 in increments of one cell time for stream L (2.83 μsec).

From the given example we can conclude that the buffer size allocation requirements depend on the bandwidth ratio of MUXed and DMUXed VCs. The ATM network will support video, image, voice, and data traffic that will require vastly different bandwidths, and will, therefore, require substantial buffering for a zero loss QoS [4], [5], [14].

For example, in the case of MUXing a video application requiring 74.88 Mb/s and a voice application requiring 64 Kb/s bandwidth, we can calculate the buffer size allocation required for zero loss service. The calculation is performed by following the same argument as illustrated in the above example. When two VCs (voice and data) are MUXed and then DMUXed, the buffer size allocation requirement is equal to $74.88-(74.88)^2/(74.88+0.064)$ Mbits = 63.95 Kbits (equivalent to 150.82 ATM cells) for the VC supporting the video application.

The buffer size allocation requirements calculation for a general case of two streams involved in the previously described MUX/DMUX scenario is shown in Figure 5. The two streams have rates i_1 and i_2 respectively. After being MUXed together into a single stream of rate $i_1 + i_2$, they are DMUXed and the first stream of cells is sent with the rate o_1. For a given time interval T, the buffer size is calculated as the difference between the rate of the incoming traffic $(i_1 + i_2)$ and the rate of outgoing traffic $o_1(t - s)$. (The service time s that might be needed to serve a cell in an ATM switch was taken into account for the sake of generality.) The buffer size required to assure no loss is equal to $i_1T - o_1(m-s)$, where $m = \frac{i_1 T}{i_1+i_2}$. Note that the buffer size allocation requirement is proportional to the ratio of the fast and slow VCs.

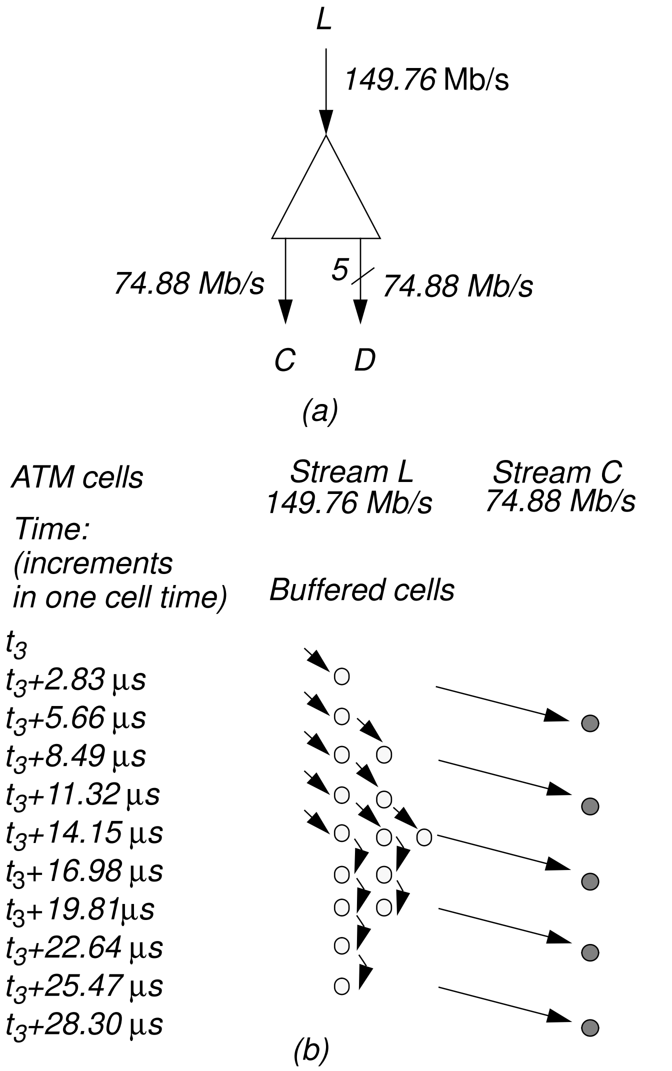

Figure 4: (a) DMUXing the streams. (b) The buffer size allocation requirement at the DMUXing point. The arrows show the buffering process of arriving cells before they can be transmitted. Time scale starts at t_3 in increments of one cell time for stream L (2.83 μsec).

One can calculate the worst case scenario with two types of traffic. If the fast VCs has allocation of 1/2 of the ATM rate, the buffer size allocation requirement is equal to 37.44 Mbits (equivalent to 88.3×10^3 ATM cells).

One may envision a worst case scenario of the case described above. ATM networks will support thousands of VCs MUXed and DMUXed together at network switching points. Consider a network where at each physical multiplexing point an ATM cell arrives simultaneously with other cells into a FIFO queue. A small example is shown in Figure 6, with larger clustering of ATM cells belonging to two distinct fast VCs A_1 and A_2. If all other cells on all links happen to be served first, the buffer size allocation requirement at all multiplexing points will be considerable and of magnitude similar to the example we presented.

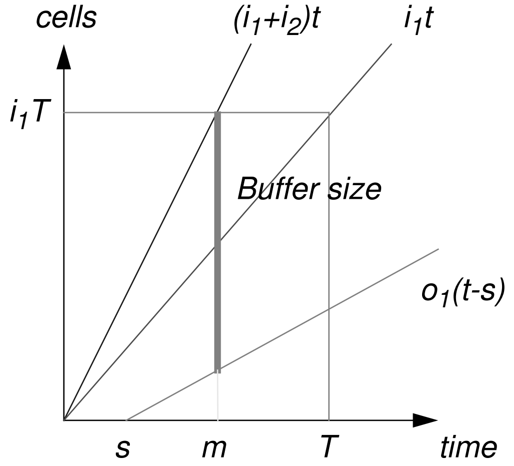

Figure 5: Buffer size estimate for time interval T. Incoming traffic rates: i_1 and i_2. Outgoing traffic rate: o_1. Instance of maximum buffer size allocation: $m = \frac{i_1 T}{i_1+i_2}$. Cell service time: s.

Another type of clustering might happen with cells belonging to the same VC. The example is shown in Figure 7 and illustrates clustering of cells belonging to stream A_1.

III. How to Remedy the High Buffer Size Allocation Requirements

Our example shows how the buffer size allocation requirements may grow in an ATM network with guaranteed service where the network supports VCs of various bandwidth allocations. The sources on the outskirts of the network have shaped traffic patterns with parameters that conform to a leaky bucket [13] traffic control scheme. We show that the buffer size allocation requirements at various hops through the network may grow unexpectedly high for a zero loss QoS even if the traffic entering the network is well regulated and conforms to the agreed rates.

Thus, this example implies that leaky bucket mechanism is not sufficient to provide guaranteed zero loss QoS without extensive buffers provided at all MUX/DMUX points in the network. Conforming and regular traffic patterns at the outskirts of an ATM network will not be sufficient in reducing these buffer size allocation requirements. Therefore, leaky bucket policing, without additional congestion control mechanisms, will not be able to provide guaranteed QoS. Furthermore, other admission control schemes (e.g., certain round robin schemes that do not take into account the speed of various VCs) might not be able to prevent similar high buffer size allocation requirements.

A possible remedy to high buffer size allocation requirements will be a scheduling algorithm for MUXing VCs with various speeds. Interleaving fast and slow cells in the MUXing points would prevent clustering of fast cells. This, in turn, will alleviate the need for buffering the fast cells in the DMUX network pints. One way to implement such interleaving would be a Weighted Fair Queueing [2] of cells belonging to VCs of different bandwidths.

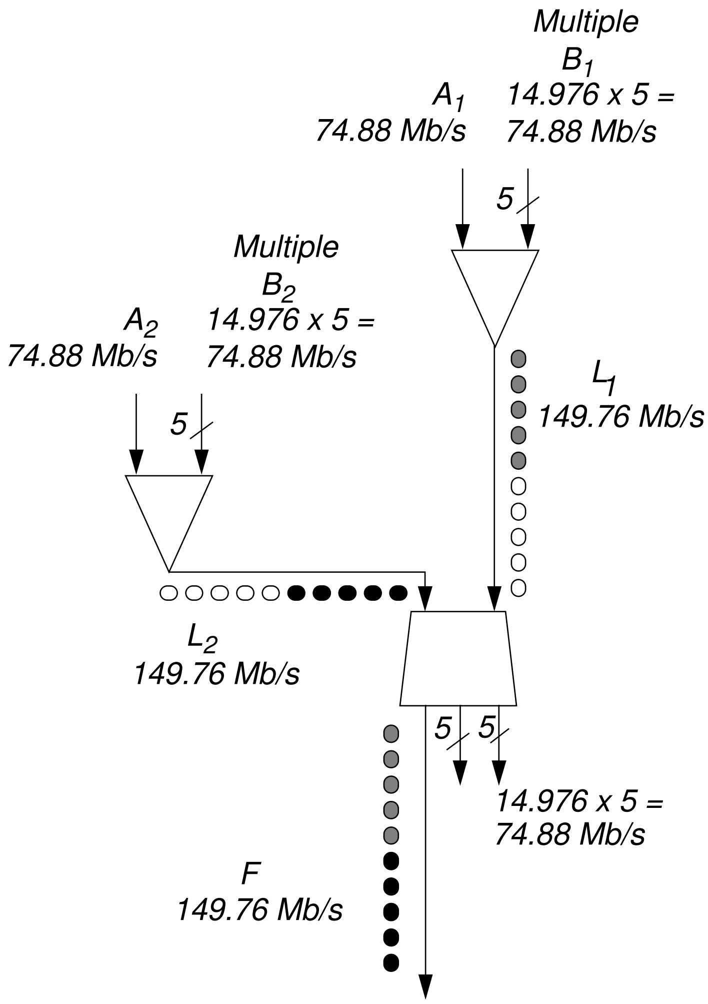

Figure 6: Clustering of cells as fast and slow ATM streams A and B arrive to MUX points and travel through links of ATM bandwidth. Traffic stream F shows clustering of cells from two separate VCs A_1 and A_2.

IV. Conclusions

We demonstrated how the buffer size allocation requirements may grow in ATM networks with guaranteed service where the network supports VCs of various bandwidth allocations. This occurs even if one uses Peak Load Allocation and the traffic entering the network is well regulated and conforms to the agreed rates. We suggested that a scheduling scheme ensuring interleaving fast and slow cells in the MUXing points would prevent clustering of fast cells. A possible implementation for such interleaving would be a Weighted Fair Queueing of cells belonging to VCs of distinct bandwidths. Furthermore, our simulations indicate that such a scheme also distributes losses (due to finite buffers) among traffic streams of various burstiness more fairly than the FIFO scheme.

V. Acknowledgments

We thank D. Sincoskie for suggesting the problem and numerous reviewers for suggestions to improving the paper.

References

[1] J. Kurose, "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks," *Computer Communications Review,* vol. 23, no. 1, 1993, pp. 6–15.

[2] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," *Proc. ACM SIGCOMM 89,* Sept. 1989, Austin, TX, pp. 3–12, *Computer Communications Review,* vol. 19, no. 4, 1989, pp. 1–12, and *Journal of Internetworking Research and Experience*, vol. 1, Sept. 1990, pp. 3–26.

[3] D. Ferrari and D. C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Selected Areas in Communications,* vol. 8, no. 3, 1990, pp. 368–379.

[4] S. J. Golestani, "A Stop-and-Go Queueing Framework for Congestion Management," *Proc. ACM SIGCOMM 90,* Sept. 1990, Philadelphia, PA, pp. 8–18.

[5] S. J. Golestani, "A Framing Strategy for Congestion Management," *IEEE J. Selected Areas in Communications,* vol. 9, no. 7, 1991, pp. 1064–1077.

[6] D. C. Verma, H. Zhang, and D. Ferrari, "Delay Jitter Control for Real-Time Communications in a Packet Switching Network," *Proc. TriComm '91,* April 1991, Chapel Hill, NC, pp. 35–43.

[7] L. Zhang, "VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks," *Proc. SIGCOMM 90,* Sept. 1990, Philadelphia, PA, pp. 19–29.

[8] C. R. Kalmanek and H. Kanakia, "Rate Controlled Servers for Very High-Speed Networks," *Proc. IEEE GLOBECOM '90,* Dec. 1990, San Diego, CA, pp. 12–20.

[9] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks—The Single Node Case," *Proc. IEEE INFOCOM '92*, May 1992, Florence, Italy, pp. 915–924.

[10] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks—The Multiple Node Case," *Proc. IEEE INFOCOM '93*, May 1993, San Francisco, CA, pp. 521–530.

[11] CCITT, *Recommendation I.371: Traffic Control and Congestion Control in B-ISDN,* Geneva, June 1992.

[12] The ATM Forum, *ATM User-Network Interface Specification,* Version 2.4, Aug. 5, 1993.

[13] J. S. Turner, "New Directions in Communications (or Which Way to the Information Age?)," *IEEE Communications Magazine*, vol. 24, no. 10, 1986, pp. 8–15.

[14] A. I. Elwalid and D. Mitra, "Analysis and Design of Rate-Based Congestion Control of High Speed Networks, I: Stochastic Fluid Models, Access Regulation," *Queueing Systems*, vol. 9, 1991, pp. 29–64.

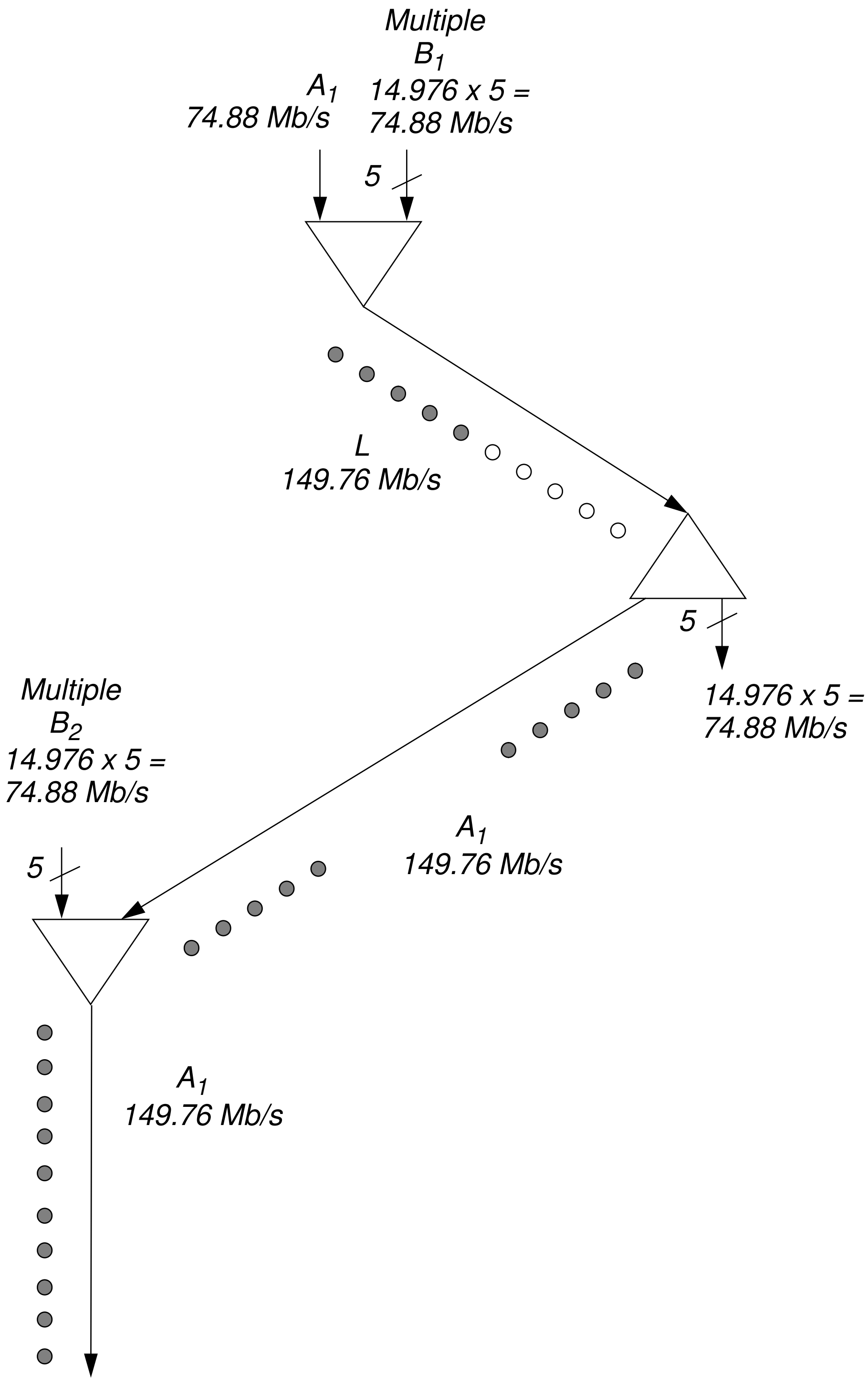

Figure 7: Clustering of cells as fast and slow ATM streams A_1 and B_1 arrive to MUX points and travel through links of ATM bandwidth. Traffic stream A_1 shows clustering of cells after being MUXed with cells from stream B_2.