Measurement of ATM Frame Latency

Arjan Durresi, Gojko Babic
Comp. & Inf. Science Dept., Ohio State Univ.
2015 Neil Ave, Columbus, OH 43210-1277
Email: {durresi, babic}@cis.ohio-state.edu

Raj Jain

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/

Abstract

Frame latency, as one of the main QoS parameters, quantifies also the level of quality of network devices. We proposed a new metric for ATM frame latency called MIMO (Message-In Message-Out) latency that improves upon other latency metrics commonly used for continuous frame technologies. Analysis of examples and measurement results showed that MIMO is less workload dependent and a better indicator of switch performance. MIMO can be used to measure the frame latency of a single switch or of a network of switches. We showed that the formulation of MIMO aggregation is very useful in understanding the contribution of each network element and their interconnections to the total frame latency.

1. Introduction

Frame latency is of particular interest as a QoS parameter, because the frame level is more likely to influence the application latency and performance. The performance of ATM equipment and the quality of services have been defined in terms of cell-level metrics such as cell transfer delay (CTD), cell delay variation (CDV) and cell loss ratio (CLR). However, cell-level metrics do not very often reflect the performance as experienced (or desired) by end users [1, 2, 9]. For example, in a video application it is important the latency of video-frames and it does not matter whether the cells belonging to a frame arrive back-to-back or regularly spaced.

This paper discusses delays that ATM switches introduce to frames, and the performance metrics to measure them. A frame is defined here as the ATM Adaptation Layer (AAL) protocol data unit (PDU). One problem in measuring the frame delay in ATM network is that when seen inside the network, the frames may be discontinuous with numerous gaps between the cells as well as cells of other frames. The monitoring equipment, if placed inside the host, will be affected by the performance of the host and may not accurately reflect the performance of the switch. Thus, the test probes of the monitoring equipment should be placed at the entrance and the exit of the system to be measured [1, 2].

In the paper we discuss the use of a new frame delay metric called MIMO latency and present its justification. The definition of MIMO latency uses the concept of an "ideal" switch, which behavior is the best a switch can do. The frame delay contribution introduced by a switch is thus defined as excess delay over the delay of an ideal switch.

The main goal in designing good metrics for performance testing is that the metrics should be, as much as possible, representative of real network situations. Also they should be independent of switch architecture and test workload to better measure the inherent level of quality of the switch. Based on the above criteria, we showed the advantage of using MIMO versus other metrics like FILO, LILO and LCD. Also we showed how to aggregate the MIMO of a network and explained the meaning of different contributions in the aggregated MIMO. A clear understanding of the different contributions to the total latency is very important in designing networks with latency constraints.

The authors have been involved in ATM Forum Test working groups since 1996 [1, 2, 9] and this paper presents some of their contributions to the Forum.

The next section analyzes why the usual frame latencies based on FILO, LILO and LCD latencies are not appropriate in an ATM environment. In Section 3 we explain the definition of MIMO latency. In Section 4, we compare MIMO vs. other latency metrics. In Section 5 results of test measurement are presented. In Section 6 we discuss about MIMO applicability and in Section 7 present the conclusions of our study.

2. Problem Statement

Some of alternative metrics usually used to measure

0-7695-0912-6/00 $10.00

frame latency are FILO (first-bit in to the last-bit out) latency, FIFO (first-bit in to the last-bit out), LILO (last-bit in to the last-bit out), and LIFO (last-bit in to the first-bit out). In Figure 1 is shown only FILO, but the other metrics can be easily obtained. Unfortunately none of the four above metrics is appropriate for ATM networks [1, 2]. So LIFO may result in negative values, FIFO does not reflect the expansion and compression of gabs on output, and LILO depends strongly on the workload. In this paper we will be focusing particularly on FILO, LILO and recently proposed [3, 4] LCD (last cell delay). LCD is a variation of LILO and measures the latency of a frame through a switch by measuring the last-bit-in-to-first-bit-out (LIFO) latency of the last cell of the frame.

For example, consider a frame consisting of three cells as shown in Figure 2. Let us assume that the input and output link speeds are identical and that the cell time is c. The switch introduces a delay of 2c to each cell. Consequently FILO latency (interval between the first bit in and the last bit out events of the frame) is 5c or in general for this switch behavior and for a *n* cell frame FILO will be:

$FILO = nc + delay$

So, FILO depends on the length of the testing frame and one can increase arbitrarily FILO by increasing *n*. Figure 3a and 3b shows the dependency of FILO also from the pattern of testing frame. In both Figure 3a and 3b FILO is equal to 8c, but clearly these values are misleading because the switches in Figure 3a and 3b are responsible for delays equal to c and 5c respectively.

Generally, the measured performance of a system depends upon the system as well as the workload. Some metrics are highly workload dependent while others are less dependent. A metric, which depends more on the system and less on the workload, is generally preferred particularly if the users are interested in comparing the systems and not the workloads. It turns out that FILO frame latency as defined has the undesirable property that it depends heavily on the workload.

Applying the definitions of LILO and LCD for the examples of Figure 2 and 3 we find that:

$LILO = delay$ *and* $LCD = delay - c$

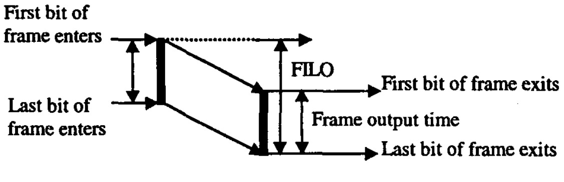


Figure 1. Frame latency metrics.

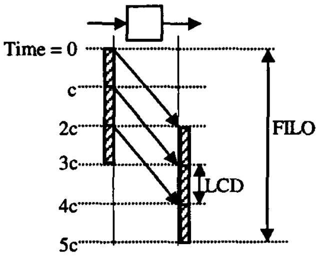


Figure 2. FILO latency

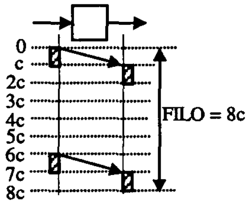


Figure 3a. Delay = c

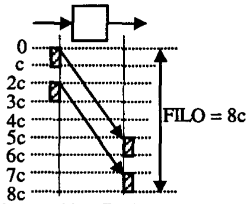


Figure 3b. Delay = 5c

So for these examples LILO and LCD depend only on the delay introduced by the switch. The problems with LILO and LCD start when the output speed of the switch is lower than its input speed. For example, consider an n-cell frame passing through a switch with input speed of *x* and output speed of *x/m*. Figure 4a shows this case for *n=3* and *m=2*. At input, each cell time is c, while at the output it is mc. In this case, the last bit of the last cell enters the switch at *nc* and the first bit of that cell exits at *(n-1)mc + delay.* The last cell's LIFO latency or LCD of the switch is *(n-1)mc + delay - nc* or *c(nm-m-n) + delay*, and LILO is equal to *cn(m-1) + delay*, where *delay* is the real delay introduced by the switch and in our example is equal to *c*. As shown in Figure 4a, for *n=3* and *m=2*, LCD is *2c* and LILO is *4c*. For *n=100*, and *m=100*, LILO and LCD will be *9901c* and *9801c* respectively. So in general for this example we have:

$LCD = c(nm - m - n) + delay$ and
$LILO = cn(m-1) + delay$

As is shown in the above relations, both LILO and LCD have an explicit dependency from the measurement

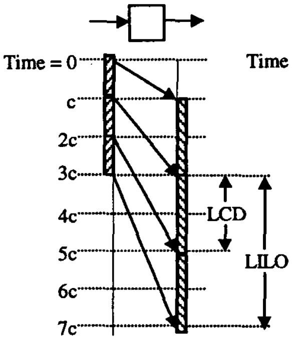


Figure 4a. LCD latency

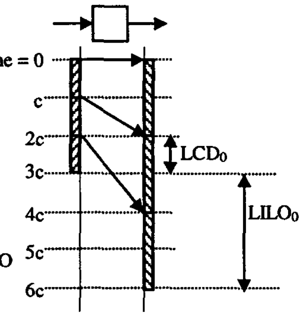


Figure 4b. Zero delay switch

load: number of frames *n* and ratio between output speed and input seed *m*.

3. MIMO definition and zero-latency switch

MIMO (Message In Message Out) latency measures the true inherent contribution of the switch to frame latency and is not affected by the arrival patterns (gaps) of the cells constituting the frame. MIMO latency is calculated for any given arrival pattern by subtracting the LILO frame latency for the pattern through the ideal switch ($LILO_0$) from the measured LILO frame latency of the switch under test gives:

$$MIMO = LILO - LILO_0 \qquad (1)$$

An ideal switch is defined as a switch that handles incoming frames in such way that they are transmitted on the output link without any unnecessary time consumption, i.e. the best any switch can do. By definition, MIMO latency for an ideal switch is zero. Hence, an ideal switch can also be called a zero-delay switch. Figure 5 shows three possible cases of behavior of an ideal switch. $LILO_0$ is zero unless input speed is higher than the output speed, shown in Figure 5c. The procedure for calculation $LILO_0$ when input speed is higher then output speed is as follows:

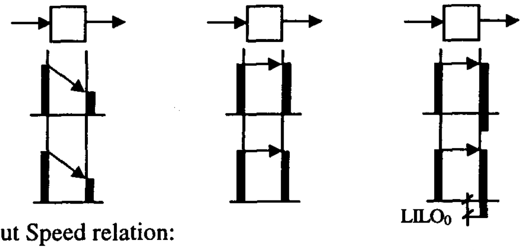
Initially $LILO_0 = 0$ and time *t* is measured from the arrival of the first bit of the first cell in a zero-delay switch.

For each cell with its first bit arriving at time *t*, update $LILO_0$ as follows:

$$LILO_0 = max\{t, LILO_0\} + COT \qquad (2)$$

Where: COT = cell output time = 424 [bits] / Output Link Rate [bits/sec]

Applying the definition of MIMO latency we find for the example of Figure 2 that MIMO is equal to 2c, and for the examples of Figure 3a and 3b it is c and 5c respectively. In each case, MIMO latency reflects the switch behavior and is not affected by the arrival pattern.



In/Out Speed relation:

a) Input < Output b) Input = Output c) Input > Output

Figure 5. Ideal Switch $LILO_0>0$ only when input link speed is higher than the output link speed (c).

$LILO_0$ is the latency that the frame would experience if is passed through an ideal switch. In other words, MIMO latency is the extra delay introduced by the switch under test, compared with its corresponding zero-delay switch. For example, let calculate MIMO for the switch shown in Figure 4a. Figure 4b shows the output pattern when the same frame is passed through an ideal switch, which sends out all bits as soon as it can and $LILO_0 = 3c$. Figure 4a shows that $LILO = 4c$, so applying the definition:

$$MIMO = LILO - LILO_0 = c$$

For other values of *n* and *m*:

$LILO_0 = n(m-1)c$

$LILO = n(m-1)c + delay$

$MIMO = LILO - LILO_0 = delay = c$

An important point, which needs to be clarified is the influence of Input/Output speeds of the switch. When we change the in/out link speed, actually we are operating with a different switch. Its measured delay may be very different from that of another switch with almost the same hardware, but different I/O speeds. The corresponding zero-delay switch is also different. For instance, in the examples given above, LILO with m=2 and m=100 are different. So are $LILO_0$s. The difference between LILO and $LILO_0$ is MIMO. MIMO definition shows that MIMO metric doesn't depend upon the measurement loads. This doesn't mean that "delay" and MIMO value do not depend for internal implementation reasons upon the measurement loads. If there is a dependence of the delay upon the measurement loads this is an issue for the designer and manufactures of the switch.

Our goal by subtracting $LILO_0$ is to subtract out the workload dependent part from the metric so that when comparing multiple switches the results are not overshadowed by the workload dependent part (for example, the part that depends upon *n* – the frame size).

If the switch would introduce no delay then this is the case an ideal switch shown in Figure 4b. In this case we have:

$delay = 0$ *and* $LCD = c(nm - m - n)$

Let name the LCD of the ideal switch LCD_0, so we have

$LCD_0 = c(nm - m - n)$

We can rewrite the expression for LCD as:

$LCD = LCD_0 + delay$

LCD_0, like $LILO_0$, is a constant factor that depends on the measurement load and has nothing to do with the delay introduced by the switch. For this reason it is desirable to remove LCD_0 from the relation used to measure the switch delay, in this case LCD. This is accomplished by MIMO latency, which definition does contain no terms explicitly depending on the measurement loads. So, MIMO latency can be obtained by LCD as:

$$MIMO = LCD - LCD_0 \tag{3}$$

4. MIMO vs. other Latency Metrics

In this Section we compare MIMO vs. other metrics based on the criteria of accountability, additivity, simplicity and negativity.

Accountability

Generally, the measured performance of a system depends upon the system as well as the workload. Some metrics are highly workload dependent while others are less dependent. A metric, which depends more on the system and less on the workload, is generally preferred particularly if the users are interested in comparing the systems and not the workloads. It turns out that the FILO, LILO and LCD frame latency as defined in [3] have the undesirable property that they depend highly on the workload. This is obvious from the example shown in Figure 2, 3 and 4. A vendor trying to sell the switch would use small frames, say, $n=2$, and claim its LCD is zero while a competing vendor will use large frames, say, n=100 and show that the same switch has large delay. In the example of Figure 4a, $LCD = c(nm - m - n) + delay$, $LILO = cn(m-1) + delay$ and $FILO = cnm + delay$ of which $c(nm - m - n)$, $cn(m-1)$ and cnm are respectively the workload dependent part and $delay = c$ is the workload-independent (or switch dependent) part.

The dependency of FILO, LILO and LCD metrics on the input frame configuration is not a desirable feature in comparing different switches performances. However, if the workload is given and a user is interested in knowing the total delay introduced for that workload, then any of the measured latencies, including FILO, LILO, or LCD can be used as an indication of delay in an "in-service" measurement. This would be the case when the user is interested in computing total delay between the entry and exit from a given network.

On the other hand, for out-of-service performance testing, where a user wants to compare multiple networks or switches and to measure their inherent level of quality, MIMO is a better indicator of the switch performance since the workload part has been taken out.

Additivity

In [5, 6], it is shown that MIMO latency of a series of components can be computed as follows:

$$MIMO_\Sigma = \Sigma MIMO_i + \Sigma LILO_{0i} - LILO_{0\Sigma} \tag{4}$$

$MIMO_i$ is the MIMO latency of the i^{th} component, $LILO_{0i}$ is the LILO latency of the i^{th} component if it were an ideal switch, and $LILO_{0\Sigma}$ is the LILO latency of the entire series if it is replaced by a black box consisting of an ideal switch. Note that computing LILO latency of an ideal switch requires knowledge of only the i/o speeds and is trivial in most cases. If the input speed is same or slower than the output speed, the LILO latency of the ideal switch is zero.

FILO is not additive.

LILO and LCD are additive so:

$$LILO_\Sigma = \Sigma LILO_i \tag{5}$$

$$LCD_\Sigma = \Sigma LCD_i \tag{6}$$

In case of LCD, to ensure additivity, we need to use different definitions for LCD, depending whether the component is a switch or a wire. For switches, LCD is defined as the LIFO latency of the last cell of the frame. For wires, LCD is defined as the FILO latency of the last cell of the frame. Also a wire must always follow a switch and vice versa.

There are two problems with differing definitions of LCD for switches are wires. First, the LCD latency of a zero-length (or very short length) wire is c (one cell time) and not zero. The latency of a 1 km of fiber would be c+5μs and not 5μs. Second, if we put two switches back-to-back the total LCD is not the sum of individual LCDs. Figure 6 shows a concatenation of two switches, with each switch having an LCD of 0, but the LCD of the two switches combined is $2c$. In this case, in order to arrive to the right result the user should add the LCD of the intermediate wire, which is defined as the FILO latency of the last cell. Note that this definition is different from the definition of LCD of a switch. The conclusion is that the additivity of LCD is not simple, it requires the use of two different definitions for LCD, depending on the delay element.

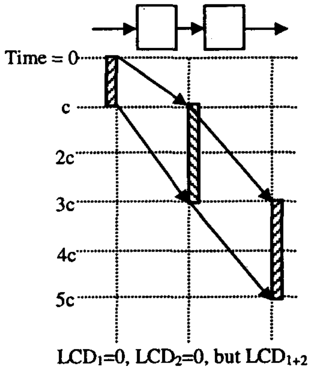
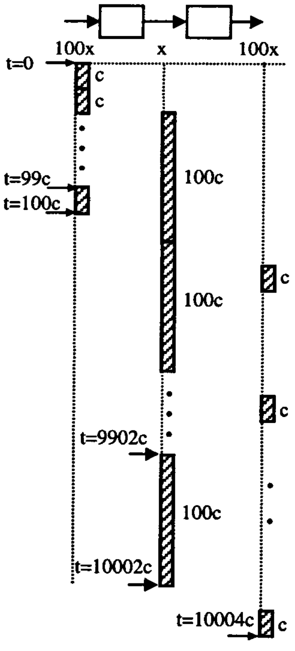


Figure 6. Two Switches back to back

Figure 7 [7] shows a concatenation of two switches. The first switch has the input speed 100 times faster than the output speed. The second switch has the output speed 100 times faster than the input speed. Both switches have the property that they unnecessarily delay cells by *2c* times. As shown, each frame has 100 cells. MIMO latency for both switches is *2c* regardless of the number of cells in the frame. LCD latency for the first switch is *9802c* and that for the second switch is only *c*, leading one to believe that the first switch is really bad. In fact, it can be made to look arbitrarily worse by simply changing the workload – number of cells in the frame.

Another point to note from Figure 7 is that while each switch is good, the system consisting of the two switches together has a bottleneck in the middle. This bottleneck is the consequence of the mismatch among the switches link speeds. If there is no special justification this bottleneck is simply the result of bad engineering. This is reflected by the combined MIMO, which is *9904c*. If the bottleneck is improved the MIMO improves. Combined LCD in this case is 9903c. When there is a link between the switches we should apply the additivity of LCD using different definitions for LCD of the switches and that of the wire. In [7] is suggested that the first switch includes the edge links and the second switch includes what remains of end to end connection. This may rise another problem, that of arbitrary assignment of the delay to network elements. For example when the intermediate link is included in the first switch we get: $LCD_1 = 9902c$ and $LCD_2 = c$. When the intermediate link is included in the second switch the results are: $LCD_1 = 9802c$ and $LCD_2 = 101c$.



$MIMO_1 = 2c$
$MIMO_2 = 2c$
$MIMO_{1+2} = MIMO_1 + MIMO_2 +$
$+ LILO_{01} + LILO_{02} - LILO_{\Sigma 0} =$
$= 2c + 2c + 9900c + 0 - 0 = 9904c$

If there is no intermediate link:
$LCD_1 = 9802c$
$LCD_2 = c$
$LCD_{1+2} = 9903 \neq LCD_1 + LCD_2$

If the intermediate link is included in the first switch:
$LCD_1 = 9902c$
$LCD_2 = c$
$LCD_{1+2} = 9903 = LCD_1 + LCD_2$

If the intermediate link is included in the first switch:
$LCD_1 = 9802c$
$LCD_2 = 101c$
$LCD_{1+2} = 9903 = LCD_1 + LCD_2$

Figure 7. Comparing LCD vs. MIMO

Simplicity

FILO, LILO, LCD and MIMO all are simple. For MIMO, one has to know the input/output speeds of the switch and the cell arrival pattern to subtract out the workload dependent part. For LCD, one has to properly classify each black-box either as a switch or a wire and to ensure that wires and switches alternate.

In cases, where one is interested in comparing multiple systems, it is important to subtract out the workload-dependent part and so the knowledge of workload is required. However, if one is interested only in in-service allocation of delays among various components for a given workload, LCD, LILO or FILO may be used.

Non-negativity

MIMO, FILO and LILO by definition give always non-negative values. LCD can be negative if the first bit of the cell come out of a component before the last bit goes in. For example, a simple digital amplifier that takes in distorted waveform and outputs noise-free square bit pattern will have a negative LCD.

5. Test Measurements

The test configuration for the LCD, LILO, FILO and MIMO latency measurements for the case with the input link speed higher than the output link speed, is shown in Figure 8 [1, 8]. It uses a 155 Mbps UTP-5 link between the monitor port 1 and the switch port A1 and a 25 Mbps link between the monitor port 2 and the switch port D1. Figure 8 also indicates the traffic flow direction.

In this configurations:

Cell Input Time = 2.83 μs

Cell Output Time = 424[bits] / Output Link Rate = 424[bits] / 25.6 [Mbps] = 16.56 μs.

All tests are performed with 32-cell frames. One of the measurements (first test) used contiguous frames, i.e. cells of the test frame were transmitted back-to-back. In the rest of the tests, we introduce identical gaps (unassigned cells or cells of other frames) between cells of the test frame.

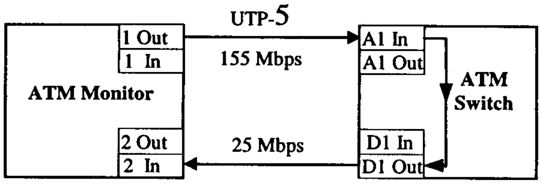


Figure 8. Test configuration to measure frame latency

Table 1 shows measurement results for eight test runs, from which LCD, LILO, FILO and MIMO latency are calculated.

Table 1: Measurement results. (All times are in μs)

Test No.	Frame Pattern	LILO	LCD	FILO	MIMO
1	No gap	385.0	368.45	563.3	33.3
2	1-cell gaps	295.8	279.22	561.8	31.8
3	2-cell gaps	209.1	192.49	562.8	32.8
4	3-cell gaps	119.8	103.26	561.3	31.3
5	4-cell gaps	44.0	27.44	560.3	30.3
6	5-cell gaps	33.6	17.01	562.8	19.9
7	6-cell gaps	35.9	19.31	652.8	22.2
8	7-cell gaps	35.6	19.01	740.3	21.9

All tests, except the first, use discontinuous frames on input, with gaps between cells of the test frame, as indicated in the second column. The third, fourth, fifth and sixth columns present measurement results for the LILO, LCD, FILO and MIMO, respectively.

Analyzing the results shown in Table 1 is clear that LILO, LCD and FILO are heavily dependent on the input frame configuration. On the other hand MIMO reflects only the delay introduced inherently by the switch itself. In this case the variation of LCD, LILO and FILO could be misleading about the performance of the switch under test. The switch latency is higher in the first 5 tests due to cell queuing. In the last three tests, the gap between the cells is large and there is no queuing. MIMO latency clearly reflects this effect

6. MIMO applicability

Based on this analysis we can conclude that MIMO and FILO, LCD or LILO have advantages and disadvantages. These metrics can be seen as complimentary because they are more appropriate in different applications. MIMO metric is better suited for comparing performance of different switches in out-of-service measurements, where it can precisely measure the delay introduced by the switch independent of the input frame. LCD, FILO or LILO can be used for in-service measurements and for user perceived delay.

If LCD is used in place of LILO, for a network of switches we will have the following expressions:

$$MIMO_{\Sigma} = \Sigma LCD_i - LCD_{0\Sigma} \quad (7)$$

where $LCD_{0\Sigma}$ is the LCD of the entire series if it is replaced by a black box consisting of an ideal switch. This relationship applies for both switches and wires. That is, LCD can be FILO or LIFO. Furthermore the expression:

$$MIMO_{\Sigma} = \Sigma MIMO_i + \Sigma LCD_{0i} - LCD_{0\Sigma} \quad (8)$$

can be used to relate the delay of the network to the delays of individual switches measured in performance testing by $MIMO_i$ and to mismatch between input and output link speeds given by LCD_{0i}.

7. Conclusions

We analyzed the advantages of using MIMO instead of other latency metrics like FILO, LILO or LCD in performance testing. In particular MIMO is better suited for out-of-service measurements where the user is interested in evaluating the delay introduced inherently by the switch and/or comparing multiple switches independent of the workload. LCD, FILO or LILO can be used for in-service measurement where the user is interested in total delay for a given workload.

8. References

[1] Arjan Durresi, Raj Jain, Gojko Babic, "Experience with ATM Switch Performance Testing", *Proceedings IEEE International Conference on Networks (ICON'99)*, Brisbane, Australia, September 28 - October 1, 1999, pp.143-150, http://www.cis.ohio-state.edu/~durresi/papers/perfexp.html

[2] Gojko Babic, Raj Jain, Arjan Durresi, "ATM Performance Testing and QoS Management" in F. Golshani, Ed., *The IEC ATM Handbook*, International Engineering Consortium, Chicago, IL, 2000, in press. http://www.cis.ohio-state.edu/~jain/papers/exp_book.html

[3] K. Glossbrenner, F. Kaudel, "ATM Layer Reference Events for Monitoring Frames," *ATM Forum contribution* 99-0282, April 1999.

[4] K. Glossbrenner, F. Kaudel, "ATM Layer Reference Events for Monitoring Frames," *Contribution to ANSI Committee T1A1.3*, May 3,1999

[5] Arjan Durresi, Raj Jain, and Gojko Babic, "Frame Delay Through ATM Switches and networks: MIMO Latency and its aggregation," *Contribution to ANSI Committee T1A1.3/99-019*, May 3,1999, http://www.cis.ohio-state.edu/~durresi/ansi/0a13019.html/

[6] Arjan Durresi, Raj Jain, and Gojko Babic, "Aggregation of MIMO Latency", *ATM Forum contribution 99-0243*, April1999, http: //www.cis.ohio-state.edu/ ~durresi/ atmf/a99-243.htm.

[7] K. Glossbrenner, F. Kaudel, "Analysis of MIMO," *Contribution to ANSI Committee T1A1-3/99-035*, July 14, 1999

[8] Arjan Durresi, Raj Jain, Gojko Babic, and Bruce Northcote "Methodology for Implementing Scalable Test Configurations in ATM Switches, " *Proceedings IEEE International Conference on Computer Communication and Networks (ICCCN'99)*, Boston, MA, October 11-13, 1999, pp. 628-633, http://www.cis.ohio-state.edu/~durresi/papers/scalab.html/

[9] ATM Forum Performance Testing Specification, *AF-TEST-TM-0131.000*, October, 1999