
ATM Forum Document Number: ~~BTD-TEST-TM-PERF.00.03 (96-0810R6)~~BTD-TEST-TM-PERF.00.04 (96-0810R7)

Title: ATM Forum Performance Testing Specification - Baseline Text

Abstract: This baseline document includes all text related to performance testing that has been agreed so far by the ATM Forum Testing Working Group.

Source:

Raj Jain, Gojko Babic, Arjan Duresi, Justin Dolske.
The Ohio State University

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

The presentation of this contribution at the ATM Forum is sponsored by NASA.

Date: ~~September~~December 1997

Distribution: ATM Forum Technical Working Group Members (AF-TEST, AF-TM)

Notice:

This contribution has been prepared to assist the ATM Forum. It is offered to the Forum as a basis for discussion and is not a binding proposal on the part of any of the contributing organizations. The statements are subject to change in form and content after further study. Specifically, the contributors reserve the right to add to, amend or modify the statements contained herein.



The ATM Forum
Technical Committee

**ATM Forum Performance
Testing Specification**

September 1997

~~BTD-TEST-TM-PERF.00.03 (96-0810R6)~~ **December 1997**

BTD-TEST-TM-PERF.00.04 (96-0810R7)

ATM Forum Performance Testing Specifications

Version 1.0, ~~September~~ December 1997

(C) 1997 The ATM Forum. All Rights Reserved. No part of this publication may be reproduced in any form or by any means.

The information in this publication is believed to be accurate at its publication date. Such information is subject to change without notice and the ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM recommendations and/or specifications or recommendations of the ATM Forum or any committee of the ATM Forum will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in the ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

Table of Contents

1. INTRODUCTION.....	1
1.1. SCOPE.....	1
1.2. GOALS OF PERFORMANCE TESTING.....	2
1.3. NON-GOALS OF PERFORMANCE TESTING.....	3
1.4. TERMINOLOGY.....	3
1.5. ABBREVIATIONS.....	4
2. CLASSES OF APPLICATION.....	4
2.1. PERFORMANCE TESTING ABOVE THE ATM LAYER.....	4
2.2. PERFORMANCE TESTING AT THE ATM LAYER.....	5
3. PERFORMANCE METRICS.....	6
3.1. THROUGHPUT.....	6
3.1.1. <i>Definitions</i>	6
3.1.2. <i>Units</i>	7
3.1.3. <i>Statistical Variations</i>	7
3.1.4. <i>Measurement Procedures</i>	7
3.1.5. <i>Foreground Traffic</i>	8
3.1.6. <i>Background Traffic</i>	12
3.1.7. <i>Guidelines For Scaleable Test Configurations</i>	12
3.1.8. <i>Reporting results</i>	14
3.2. FRAME LATENCY.....	14
3.2.1. <i>Definition</i>	14
3.2.2. <i>Units</i>	16
3.2.3. <i>Statistical Variations</i>	16
3.2.4. <i>Measurement Procedures</i>	16
3.2.5. <i>Foreground traffic</i>	17
3.2.6. <i>Background Traffic</i>	17
3.2.7. <i>Guidelines For Scaleable Test Configurations</i>	18
3.2.8. <i>Reporting results</i>	20
3.3. THROUGHPUT FAIRNESS.....	21
3.3.1. <i>Definition</i>	21
3.3.2. <i>Units</i>	21
3.3.3. <i>Measurement procedures</i>	22
3.3.4. <i>Statistical Variations</i>	22
3.3.5. <i>Reporting Results</i>	22
3.4. FRAME LOSS RATIO.....	22
3.4.1. <i>Definition</i>	22
3.4.2. <i>Units</i>	23
3.4.3. <i>Measurement Procedures</i>	23
3.4.4. <i>Statistical Variations</i>	23
3.4.5. <i>Reporting Results</i>	23
3.5. MAXIMUM FRAME BURST SIZE (MFBS).....	23
3.5.1 <i>Definition</i>	23
3.5.2 <i>Units</i>	24
3.5.3 <i>Statistical Variations</i>	24
3.5.4 <i>Measurement Procedure and MFBS Calculation</i>	24
3.5.5 <i>Reporting Results</i>	25
3.6. CALL ESTABLISHMENT LATENCY.....	25
3.6.1. <i>Definition</i>	25
3.6.2. <i>Units</i>	26

3.6.3. <i>Configurations</i>	26
3.6.4. <i>Statistical Variations</i>	27
3.6.5. <i>Guidelines For Using This Metric</i>	27
3.7. APPLICATION GOODPUT.....	27
3.7.1. <i>Guidelines For Using This Metric</i>	28
4. REFERENCES.....	28
APPENDIX A: DEFINING FRAME LATENCY ON ATM NETWORKS.....	30
A.1. INTRODUCTION.....	30
A.2. USUAL FRAME LATENCIES AS METRICS FOR ATM SWITCH DELAY	34
A.3. MIMO LATENCY DEFINITION	37
A.4. CELL AND CONTIGUOUS FRAME LATENCY THROUGH A ZERO-DELAY SWITCH	38
A.5. LATENCY OF DISCONTINUOUS FRAMES PASSING THROUGH A ZERO-DELAY SWITCH.....	51
A.6. CALCULATION OF FILO LATENCY FOR A ZERO-DELAY SWITCH	54
A.7. EQUIVALENT MIMO LATENCY DEFINITION.....	54
A.8. MEASURING MIMO LATENCY	55
A.9. USER PERCEIVED DELAY	56
APPENDIX B: METHODOLOGY FOR IMPLEMENTING CONNECTION CONFIGURATIONS	58
B.1. INTRODUCTION.....	58
B.2. DEFINITIONS AND RULES.....	59
B.3. CONNECTION CONFIGURATIONS CHARACTERISTICS.	62
<i>B.3.1 N-to-N Straight (Single Generator)</i>	62
<i>B.3.2. N-to-N Straight (r Generators)</i>	63
<i>B.3.3. N-to-m Partial Cross (r Generators)</i>	65
B 4. ALGORITHM FOR CREATING VCC CHAINS.	68

1. Introduction

Performance testing in ATM deals with the measurement of the level of quality of a system under test (SUT) or an interface under test (IUT) under well-known conditions. The level of quality can be expressed in the form of metrics such as latency, end-to-end delay, effective throughput. Performance testing can be carried at the end-user application level (e.g., FTP, NFS), at or above the ATM layers (e.g., cell switching, signaling, etc.). Performance testing also describes in details the procedures for testing the IUTs in the form of test suites. These procedures are intended to test the SUT or IUT and do not assume or imply any specific implementation or architecture of these systems.

This document highlights the objectives of performance testing and suggests an approach for the development of the test suites.

1.1. Scope

Asynchronous Transfer Mode, as an enabling technology for the integration of services, is gaining an increasing interest and popularity. ATM networks are being progressively deployed and in most cases a smooth migration to ATM is prescribed. This means that most of the existing applications can still operate over ATM via service emulation or service interworking along with the proper adaptation of data formats. At the same time, several new applications are being developed to take full advantage of the capabilities of the ATM technology through an Application Protocol Interface (API).

While ATM provides an elegant solution to the integration of services and allows for high levels of scalability, the performance of a given application may vary substantially with the IUT or the SUT utilized. The variation in the performance is due to the complexity of the dynamic interaction between the different layers. For example, an application running with TCP/IP stacks will yield different levels of performance depending on the interaction of the TCP window flow control mechanism and the ATM network congestion control mechanism used. Hence, the following points and recommendations are made. First, ATM adopters need guidelines on the measurement of the performance of user applications over different systems. Second, some functions above the ATM layer, e.g., adaptation, signaling, constitute applications (i.e. IUTs) and as such should be considered for performance testing. Also, it is essential that these layers be implemented in compliance with the ATM Forum specifications. Third, performance testing can be executed at the ATM layer in relation to the QoS provided by the different service categories. Finally, because of the extensive list of available applications, it is preferable to group applications in generic classes. Each class of applications requires different testing environment such as metrics, test suites and traffic test patterns. It is noted that the same application, e.g., ftp, can yield different performance results depending on the underlying layers used (TCP/IP to ATM versus TCP/IP to MAC layer to ATM). Thus performance results should be compared based on the utilization of the same protocol stack.

Performance testing is related to user perceived performance of ATM technology. In other words, goodness of ATM will be measured not only by cell level performance but also by frame-level performance and performance perceived at higher layers.

Most of the quality of Service (QoS) metrics, such as cell transfer delay (CTD), cell delay variation (CDV), cell loss ratio (CLR), and so on, may or may not be reflected directly in the performance perceived by the user. For example, while comparing two switches if one gives a CLR of 0.1% and a frame loss ratio of 0.1% while the other gives a CLR 1% but a frame loss of 0.05%, the second switch will be considered superior by many users.

ATM Forum and ITU-T have standardized the definitions of ATM layer QoS metrics. ~~We need to do metrics [1, 2]. This specification does~~ the same for higher level layer performance metrics. Without a standard definition, each vendor will use their own definition of common metrics such as throughput and latency resulting in a confusion in the market place. Avoiding such a confusion will help buyers eventually leading to better sales resulting in the success of the ATM technology.

The initial work at the ATM Forum will be restricted to the native ATM layer and the adaptation layer. Any work on the performance of the higher layers is being deferred for further study.

1.2. Goals of Performance Testing

The goal of this effort is to enhance the marketability of ATM technology and equipment. Any additional criteria that helps in achieving that goal can be added later to this list.

- a. The ATM Forum shall define metrics that will help compare various ATM equipment in terms of performance.
- b. The metrics shall be such that they are independent of switch or NIC architecture.
 - (i) The same metrics shall apply to all architectures.
- c. The metrics can be used to help predict the performance of an application or to design a network configuration to meet specific performance objectives.
- d. The ATM Forum will develop a precise methodology for measuring these metrics.
 - (i) The methodology will include a set of configurations and traffic patterns that will allow vendors as well as users to conduct their own measurements.
- e. The testing shall cover all classes of service including CBR, rt-VBR, nrt-VBR, ABR, and UBR.
- f. The metrics and methodology for different service classes may be different.
- g. The testing shall cover as many protocol stacks and ATM services as possible.
 - (i) As an example, measurements for verifying the performance of services such as IP, Frame Relay and SMDS over ATM may be included.
- h. The testing shall include metrics to measure performance of network management, connection setup, and normal data transfer.
- i. The following objectives are set for ATM performance testing:
 - (i) Definition of criteria to be used to distinguish classes of applications.

- (ii) Definition of classes of applications, at or above the ATM Layer, for which performance metrics are to be provided.
- (iii) Identification of the functions at or above the ATM Layer which influence the perceived performance of a given class of applications. Example of such functions include traffic shaping, quality of service, adaptation, etc. These functions need to be measured in order to assess the performance of the applications within that class.
- (iv) Definition of common performance metrics for the assessment of the performance of all applications within a class. The metrics should reflect the effect of the functions identified in (iii).
- (v) Provision of detailed test cases for the measurement of the defined performance metrics.

1.3. Non-Goals of Performance Testing

- a. The ATM Forum is not responsible for conducting any measurements.
- b. The ATM Forum will not certify measurements.
- c. The ATM Forum will not set thresholds such that equipment performing below those thresholds are called "unsatisfactory."
- d. The ATM Forum will not establish any requirement that dictates a cost versus performance ratio.
- e. The following areas are excluded from the scope of ATM performance testing:
 - (i) Applications whose performance cannot be assessed by common implementation independent metrics. In this case the performance is tightly related to the implementation. An example of such applications is network management, whose performance behavior depends on whether it is a centralized or a distributed implementation.
 - (ii) Performance metrics which depend on the type of implementation or architecture of the SUT or the IUT.
 - (iii) Test configurations and methodologies which assume or imply a specific implementation or architecture of the SUT or the IUT.
 - (iv) Evaluation or assessment of results obtained by companies or other bodies.
 - (v) Certification of conducted measurements or of bodies conducting the measurements.

1.4. Terminology

The following definitions are used in this document:

- *Implementation Under Test (IUT)*: The part of the system that is to be tested.
- *Metric*: a variable or a function that can be measured or evaluated and which reflects quantitatively the response or the behavior of an IUT or an SUT.
- *System Under Test (SUT)*: The system in which the IUT resides.
- *Test Case*: A series of test steps needed to put an IUT into a given state to observe and describe its behavior.
- *Test Suite*: A complete set of test cases, possibly combined into nested test groups, that is necessary to perform testing for an IUT or a protocol within an IUT.

1.5. Abbreviations

ISO	International Organization for Standardization
IUT	Implementation Under Test
NP	Network Performance
NPC	Network Parameter Control
PDU	Protocol Data Unit
PVC	Permanent Virtual Circuit
QoS	Quality of Service
SUT	System Under Test
SVC	Switched Virtual Circuit
WG	Working Group

2. Classes of Application

Developing a test suite for each existing and new application can prove to be a difficult task. Instead, applications should be grouped into categories or classes. Applications in a given class have similar performance requirements and can be characterized by common performance metrics. This way, the defined performance metrics and test suites will be valid for a range of applications. Classes of application can be defined based on one or a combination of criteria. The following criteria can be used in the definition of the classes:

- (i) Time or delay requirements: real-time versus non real-time applications.
- (ii) Distance requirements: LAN versus WAN applications.
- (iii) Media type: voice, video, data, or multimedia application.
- (iv) Quality level: for example desktop video versus broadcast quality video.
- (v) ATM service category used: some applications have stringent performance requirements and can only run over a given service category. Others can run on several service categories. An ATM service category relates application aspects to network functionalities.
- (vi) Others to be determined.

2.1. Performance Testing Above the ATM Layer

Performance metrics can be measured at the user application layer, and sometimes at the transport layer and the network layer, and can give an accurate assessment of the perceived performance. Since it is difficult to cover all the existing applications and all the possible combinations of applications and underlying protocol stacks, it is desirable to classify the applications into classes. Performance metrics and performance test suites can be provided for each class of applications.

The perceived performance of a user application running over an ATM network is dependent on many parameters. It can vary substantially by changing an underlying protocol stack, the ATM service category it uses, the congestion control mechanism used in the ATM network, etc. Furthermore, there is no direct and unique relationship between the ATM Layer Quality of Service (QoS) parameters and the perceived application performance. For example, in an ATM network implementing a packet level discard congestion mechanism, applications using TCP as the transport protocol may see their effective throughput improved while the measured cell loss ratio may be relatively high. In practice, it is difficult to carry out measurements in all the layers that span the region between the ATM Layer and the user application layer given the inaccessibility of testing points. More effort needs to be invested to define the performance at these layers. These layers include adaptation, signaling, etc.

2.2. Performance Testing at the ATM Layer

The notion of application at the ATM Layer is related to the service categories provided by the ATM service architecture. The Traffic Management Specification, version 4.0 [2], specifies five service categories: CBR, rt-VBR, nrt-VBR, UBR, and ABR. Each service category defines a relation of the traffic characteristics and the Quality of Service (QoS) requirements to network behavior. There is an assessment criteria of the QoS associated with each of these parameters. These are summarized below.

QoS PERFORMANCE PARAMETER	QoS ASSESSMENT CRITERIA
Cell Error Ratio	Accuracy
Severely-Errored Cell Block Ratio	Accuracy
Cell Misinsertion Ratio	Accuracy
Cell Loss Rate	Dependability
Cell Transfer Delay	Speed
Cell Delay Variation	Speed

A few Measurement methods for the measurement of the QoS parameters are defined in Appendix C of [1] and Appendix B of [2]. However, detailed test cases and procedures, as well as test configurations are needed for both in-service and out-of-service measurement of QoS parameters. An example of test configuration for the out-of-service measurement of QoS parameters is given in [4]-Appendix A of [3].

Performance testing at the ATM Layer covers the following categories:

- (i) In-service and out-of-service measurement of the QoS performance parameters for all five service categories (or application classes in the context of performance testing): CBR, rt-VBR, nrt-VBR, UBR, and ABR. The test configurations assume a non-overloaded SUT.
- (ii) Performance of the SUT under overload conditions. In this case, the efficiency of the congestion avoidance and congestion control mechanisms of the SUT are tested.

In order to provide common performance metrics that are applicable to a wide range of SUT's and that can be uniquely interpreted, the following requirements must be satisfied:

- (i) Reference load models for the five service categories CBR, rt-VBR, nrt-VBR, UBR, and ABR, are required. Reference load models are to be defined by the Traffic Management Working Group.
- (ii) Test cases and configurations must not assume or imply any specific implementation or architecture of the SUT.

3. Performance Metrics

In the following description System Under Test (SUT) refers to an ATM switch. However, the definitions and measurement procedures are general and may be used for other devices or a network consisting of multiple switches as well.

3.1. Throughput

3.1.1. Definitions

There are three frame-level throughput metrics that are of interest to a user:

- **Loss-less throughput** - It is the maximum rate at which none of the offered frames is dropped by the SUT.
- **Peak throughput** - It is the maximum rate at which the SUT operates regardless of frames dropped. The maximum rate can actually occur when the loss is not zero.
- **Full-load throughput** - It is the rate at which the SUT operates when the input links are loaded at 100% of their capacity.

A model graph of throughput vs. input rate is shown in Figure 3.1. Level X defines the loss-less throughput, level Y defines the peak throughput and level Z defines the full-load throughput.

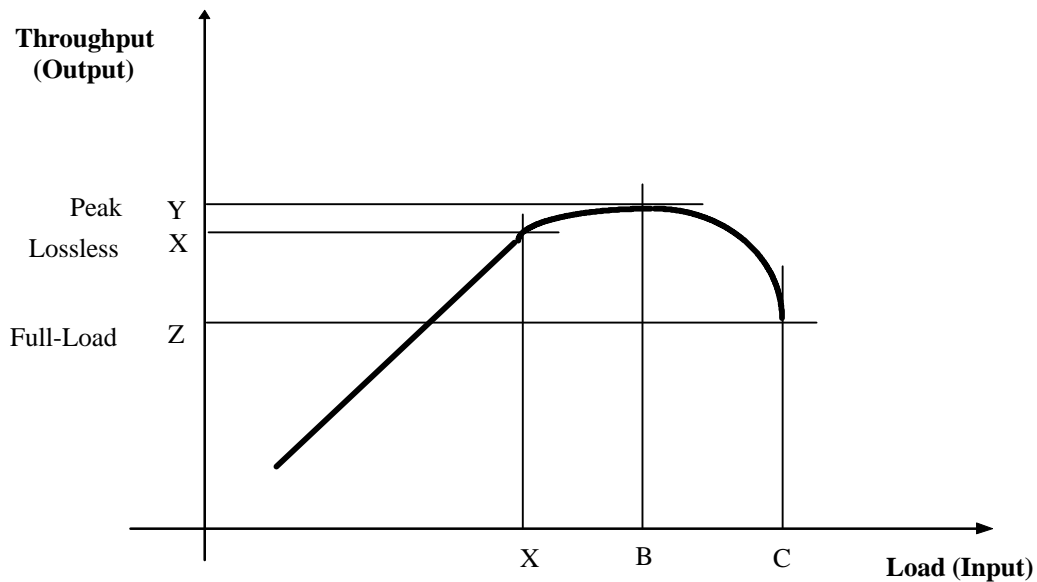


Figure 3.1: Peak, loss-less and full-load throughput

The loss-less throughput is the highest load at which the count of the output frames equals the count of the input frames. The peak throughput is the maximum throughput that can be achieved in spite of the losses. The full-load throughput is the throughput of the system at 100% load on input links. Note that the peak throughput may equal the loss-less throughput in some cases.

Only frames that are received completely without errors are included in frame-level throughput computation. Partial frames and frames with CRC errors are not included.

3.1.2. Units

Throughput should be expressed in the effective bits/sec, counting only bits from frames excluding the overhead introduced by the ATM technology and transmission systems.

This is preferred over specifying it in frames/sec or cells/sec. Frames/sec requires specifying the frame size. The throughput values in frames/sec at various frame sizes cannot be compared without first being converted into bits/sec. Cells/sec is not a good unit for frame-level performance since the cells aren't seen by the user.

3.1.3. Statistical Variations

There is no need for obtaining more than one sample for any of the three frame-level throughput metrics. Consequently, there is no need for calculation of the means and/or standard deviations of throughputs.

3.1.4. Measurement Procedures

Before starting measurements, a number of VCCs (or VPCs), henceforth referred to as “foreground VCCs”, are established through the SUT. Foreground VCCs are used to transfer only the traffic whose performance is measured. That traffic is referred as the foreground traffic. Characteristics of foreground traffic are specified in 3.1.5.

The tests can be conducted under two conditions:

- without background traffic;
- with background traffic;

Procedure without background traffic

The procedure to measure throughput in this case includes a number of test runs. A test run starts with the traffic being sent at a given input rate over the foreground VCCs with early packet discard disabled (if this feature is available in the SUT and can be turned off). The average cell transfer delay is constantly monitored. A test run ends and the foreground traffic is stopped when the average cell transfer delay has not significantly changed (not more than 5%) during a period of at least 5 minutes.

During the test run period, the total number of frames sent to the SUT and the total number of frames received from the SUT are recorded. The throughput (output rate) is computed based on the duration of a test run and the number of received frames.

If the input frame count and the output frame count are the same then the input rate is increased and the test is conducted again.

The loss-less throughput is the highest throughput at which the count of the output frames equals the count of the input frames.

The input rate is then increased even further (with early packet discard enabled, if available). Although some frames will be lost, the throughput may increase till it reaches the peak throughput value. After this point, any further increase in the input rate will result in a decrease in the throughput.

The input rate is finally increased to 100% of the link input rates and the full-load throughput is recorded.

Procedure with background traffic

Measurements of throughput with background traffic are under study.

3.1.5. Foreground Traffic

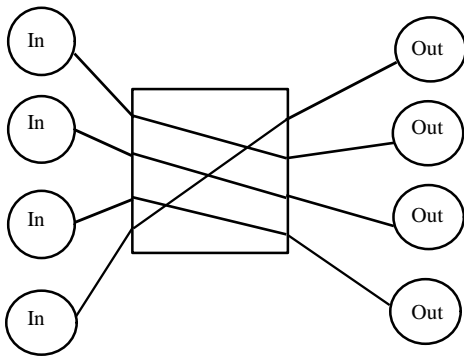
Foreground traffic is specified by the type of foreground VCCs, connection configuration, service class, arrival patterns, frame length and input rate.

Foreground VCCs can be permanent or switched, virtual path or virtual channel connections, established between ports on the same network module on the switch, or between ports on different network modules, or between ports on different switching fabrics.

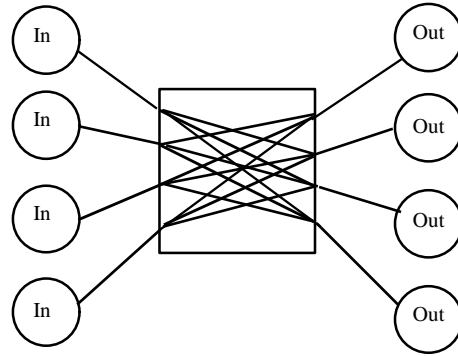
A system with n ports can be tested for the following connection configurations:

- n -to- n straight,
- n -to- $(n-1)$ full cross,
- n -to- m partial cross, $1 \leq m \leq n-1$,
- k -to-1, $1 < k < n$,
- 1-to- $(n-1)$ multicast,
- n -to- $(n-1)$ multicast.

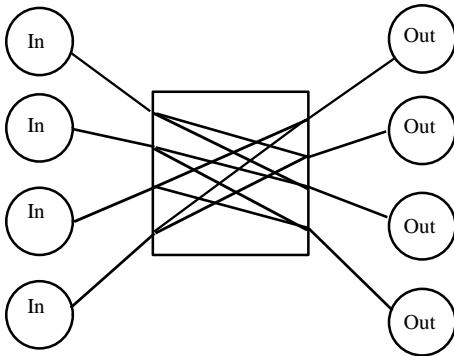
Different connection configurations are illustrated in Figure 3.2, where each configuration includes one ATM switch with four ports, with their input components shown on the left and their output components shown the right.



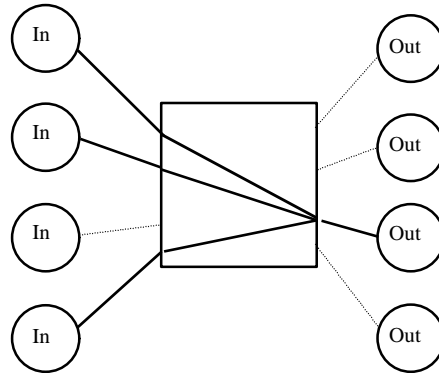
a. n-to-n straight: n VCCs; $n=4$



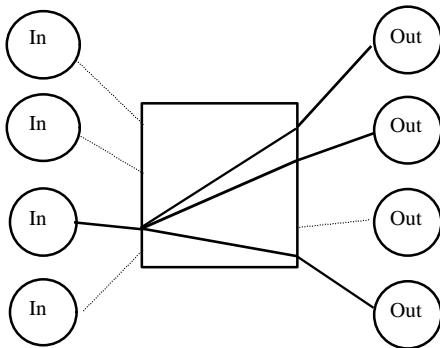
b. n-to-(n-1) full cross: $n \times (n-1)$ VCCs; $n=4$



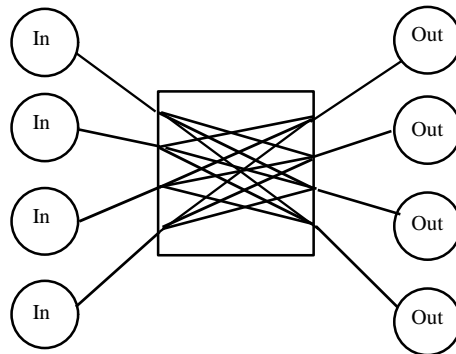
c. n-to-m partial cross: $n \times m$ VCCs; $n=4$, $m=2$



d. k-to-1: k VCCs; $k=3$



e. 1-to-(n-1) multicast: one (multicast) VCC



f. n-to-(n-1) multicast: n (multicast) VCCs; $n=4$

Figure 3.2: Connection configurations for foreground traffic

In the case of n-to-n straight, input from one port exits to another port. This represents almost no path interference among the foreground VCCs. There are n foreground VCCs. See Figure 3.2a.

In the case of n-to-(n-1) full cross, input from each port is divided equally to exit on each of the other (n-1) ports. This represents intense competition for the switching fabric by the foreground VCCs. There are $n \times (n-1)$ foreground VCCs. See Figure 3.2b.

In the case of n-to-m partial cross, input from each port is divided equally to exit on the other m ports ($1 \leq m \leq n-1$). This represents partial competition for the switching fabrics by the foreground VCCs. There are $n \times m$ foreground VCCs as shown in Figure 3.2c. Note that n-to-n straight and n-to-(n-1) full cross are special cases of n-to-m partial cross with $m=1$ and $m=n-1$, respectively.

In the case of k-to-1, input from k ($1 < k < n$) ports is destined to one output port. This stresses the output port logic. There are k foreground VCCs as shown in Figure 3.2d.

In the case of 1-to-(n-1) multicast, all foreground frames input on the one designated port are multicast to all other (n-1) ports. This tests single multicast performance of the switch. There is only one (multicast) foreground VCC as shown in Figure 3.2e.

Use of the 1-to-(n-1) multicast connection configuration for the foreground traffic is under study.

In the case of n-to-(n-1) multicast, input from each port is multicast to all other (n-1) ports. This tests multiple multicast performance of the switch. There are n (multicast) foreground VCCs. See Figure 3.2f.

Use of the n-to-(n-1) multicast connection configuration for the foreground traffic is under study.

The following service classes, arrival patterns and frame lengths for foreground traffic are used for testing:

- UBR service class: Traffic consists of equally spaced frames of fixed length. Measurements are performed at AAL payload size of 64 B, 1518 B, 9188 B and 64 kB. Variable length frames and other arrival patterns (e.g. self-similar) are under study.
- ABR and VBR service classes are under study.

The required input rate of foreground traffic is obtained by loading each link by the same fraction of its input rate. In this way, the input rate of foreground traffic can also be referred to as a fraction (percentage) of input link rates. The maximum foreground load (MFL) is defined as the sum of rates of all links in the maximum possible switch configuration. Input rate of the foreground traffic is expressed in the effective bits/sec, counting only bits from frames, excluding the overhead introduced by the ATM technology and transmission systems.

3.1.6. Background Traffic

Higher priority traffic (like VBR or CBR) can act as background traffic for experiments. Further details of measurements with background traffic using multiple service classes simultaneously are under study. Until then, all testing will be done without any background traffic.

3.1.7. Guidelines For Scaleable Test Configurations

It is obvious that testing larger systems, e.g., switches with larger number of ports, could require very extensive (and expensive) measurement equipment. Hence, we introduce scaleable test configurations for throughput measurements that require only one ATM monitor with one generator/analyzer pair. Figure 3.3 presents a simple test configuration for an ATM switch with eight ports in a 8-to-8 straight connection configuration. Figure 3.4 presents a test configuration with the same switch in an 8-to-2 partial cross connection configuration. The former configuration emulates 8 foreground VCCs, while the later emulates 16 foreground VCCs.

In both test configurations, there is one link between the ATM monitor and the switch. The other seven ports have external loopbacks. A loopback on a given port causes the frames transmitted over the output of the port to be received by the input of the same port.

The test configurations in Figure 3.3 and Figure 3.4 assume two network modules in the switch, with switch ports P0-P3 in one network module and switch ports P4-P7 in the another network module. Foreground VCCs are always established from a port in one network module to a port in the another network module. These connection configurations could be more demanding on the SUT than the cases where each VCC uses ports in the same network module. An even more demanding case could be when foreground VCCs use different fabrics of a multi-fabric switch.

Approaches similar to those in Figure 3.3 and Figure 3.4 can be used for n-to-(n-1) full cross and other types of n-to-m partial cross connection configurations, as well as for larger switches. Guidelines to set up scaleable test configurations for the k-to-1 connection configuration are under study.

It should be noted that in the proposed test configurations, because of loopbacks, only permanent VCCs or VPCs can be established.

It should also be realized that in the test configurations with loopbacks, if all link rates are not identical, it is not possible to generate foreground traffic equal to the MFL. The maximum foreground traffic load for a n-port switch in those cases equals $n \times$ lowest link rate. Only in the case when all link rates are identical is it possible to obtain MFL level. If all link rates are not identical, and the MFL level needs to be reached, it is necessary to have more than one analyzer/generator pair.

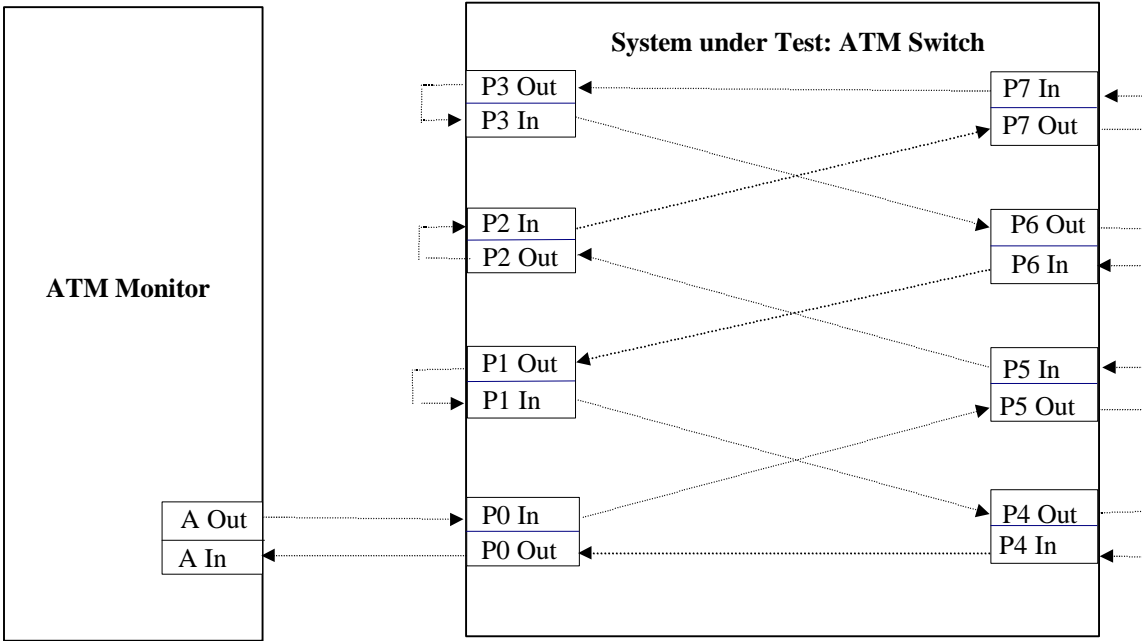


Figure 3.3: A scaleable test configuration for throughput measurements using only one generator/analyzer pair with 8-port switch and a 8-to-8 straight connection configuration

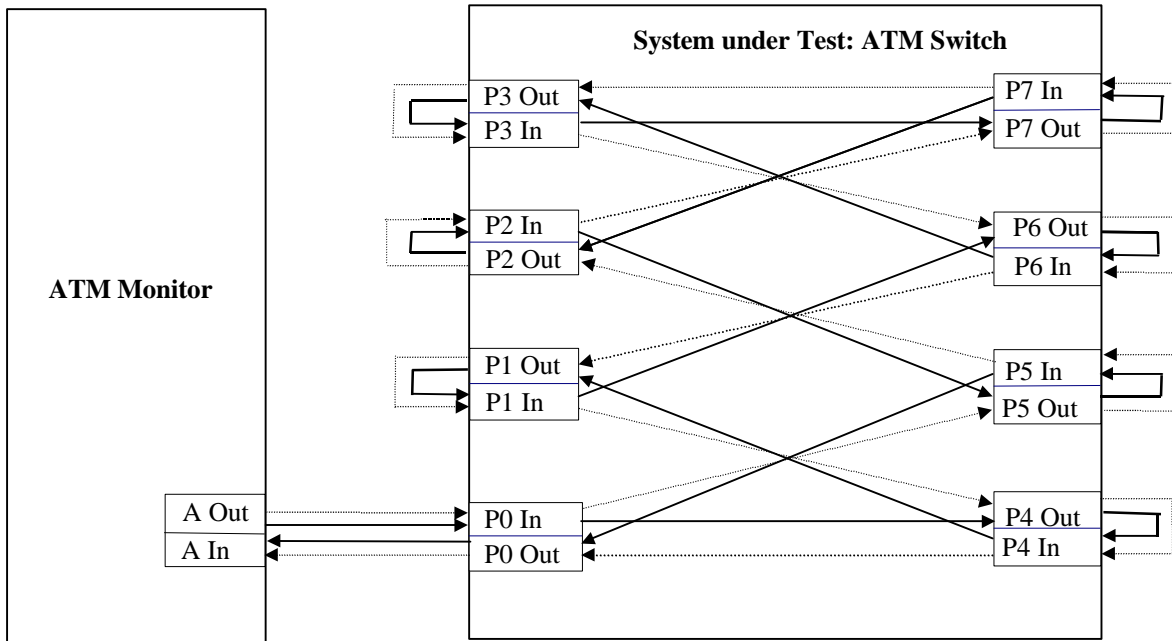


Figure 3.4: A scaleable test configuration for throughput measurements using only one generator/analyzer pair with 8-port switch and a 8-to-2 partial cross connection configuration

3.1.8. Reporting results

Results should include a detailed description of the SUT, such as the number of ports, rate of each port, number of ports per network module, number of network modules, number of network modules per fabric, number of fabrics, maximum foreground load (MFL), software version, and any other relevant information.

Values for the loss-less throughput, the peak throughput with corresponding input load, and the full-load throughput with corresponding input load (if different from MFL) are reported along with foreground (and background, if any) traffic characteristics.

The list of foreground traffic characteristics and their possible values are now provided:

- type of foreground VCCs: permanent virtual path connections, switched virtual path connections, **permanent virtual channel connections**, switch virtual channel connections;
- foreground VCCs established: between ports inside a network module, **between ports on different network modules**, between ports on different fabrics, some combination of previous cases;
- connection configuration: n-to-n straight, n-to-(n-1) full cross, **n-to-m partial cross** with $m = 2, 3, 4, \dots, n-1$, **k-to-1** with $k=2, 3, 4, 5, 6, \dots$, 1-to-(n-1) multicast, n-to-(n-1) multicast;
- service class: **UBR**, ABR, VBR;
- arrival patterns: **equally spaced frames**, self-similar, random;
- frame length: 64 B, **1518 B**, **9188 B** or 64 kB, variable;

Values in bold indicate traffic characteristics for which measurement tests must be performed and for which throughput values must be reported.

3.2. Frame Latency

3.2.1. Definition

The frame latency for a system under test is measured using a "Message-in Message-out (MIMO)" definition. Succinctly, MIMO latency is defined as follows:

$$\text{MIMO latency} = \text{FILO latency} - \text{NFOT}$$

where

- FILO latency = Time between the first-bit entry and the last-bit exit
- NFOT = Nominal Frame Output Time, defined as the time a frame needs to pass through *the zero-delay switch*, that can be calculated using the following procedure:

Initially NFOT = 0 and time t is measured from the arrival of the first bit of the first cell. For each cell with its first bit arriving at time $t \Rightarrow \text{NFOT} = \max\{t, \text{NFOT}\} + \text{CT}$.

Here CT is the larger of the cell input time or cell output time. Cell times are computed as the cell size of 424 bits divided by the respective link rates in bits per sec.

An equivalent MIMO latency definition is:

$$\text{MIMO latency} = \begin{cases} \text{LILO latency} & \text{if Input Link Rate} \leq \text{Output Link Rate} \\ \text{FILO latency} - \text{NFOT} & \text{otherwise} \end{cases}$$

where

- LILO latency = Time between the last-bit entry and the last-bit exit

Frame Latency Measurements and Calculation

To obtain MIMO latency for a given frame, the time of occurrence for the following two events need to be recorded:

- First-bit of the frame enters into the SUT,
- Last-bit of the frame exits from the SUT.

The time between the second and the first events is FILO latency. If measurement data are available at cell level, what is usually the case with contemporary ATM monitors, it can be shown that:

$$\text{FILO latency} = \text{First cell's transfer delay} + \text{First cell to last cell inter-arrival time}$$

where

- cell transfer delay (CTD) is the time between the first bit of the cell entering the switch and the last bit of the cell leaving the switch,
- cell inter-arrival time is the time between arrival from the switch of the last bit of the first cell and arrival from the switch of the last bit of the second cell.

Given the cell pattern of a frame on input, NFOT can be obtained using the procedure from its definition. Then, substituting FILO latency and NFOT in the MIMO latency formula would give the SUT delay for the given frame.

In the cases when Input Link Rate \leq Output Link Rate, MIMO latency can be obtained easier. In those cases, the time of occurrence for the following two events need to be recorded:

- Last-bit of the frame enters into the SUT,
- Last-bit of the frame exits from the SUT.

The time between the second and the first events is LILO latency. When measurement data are available at cell level, it can be shown that:

$$\text{LILO latency} = \text{Last cell's transfer delay} - \text{Cell input time}$$

and in these cases, LILO latency would give the SUT delay for the given frame.

An explanation of MIMO latency and its justification is presented in Appendix A.

3.2.2. Units

The latency should be specified in μsec .

3.2.3. Statistical Variations

For the given foreground traffic and background traffic, the required times and/or delays, needed for MIMO latency calculation, are recorded for p frames, according to the procedures described in 3.2.4. Here p is a parameter and its default (and the minimal value) is 100.

Let M_i be the MIMO latency of the i th frame. Note that MIMO latency is considered to be infinite for lost or corrupted frames. The mean and standard errors of the measurement are computed as follows:

$$\text{Mean MIMO latency} = (\sum M_i) / p$$

$$\text{Standard deviation of MIMO latency} = (\sum (M_i - \text{mean MIMO latency})^2) / (p-1)$$

$$\text{Standard error} = \text{standard deviation of MIMO latency} / p^{1/2}$$

Given the mean and the standard error, the users can compute a $100(1-a)$ -percent confidence interval as follows:

$$100(1-a)\text{-percent confidence interval} = (\text{mean} - z \times \text{standard error}, \text{mean} + z \times \text{standard error})$$

Here, z is the $(1-a/2)$ -quantile of the unit normal variate. For commonly used confidence levels, the quantile values are as follows:

Confidence	a	Quantile
90%	0.1	1.615
99%	0.01	2.346
99.9%	0.001	3.291

The value of p can be chosen differently from its default value to obtain the desired confidence level.

3.2.4. Measurement Procedures

For MIMO latency measurements, it is first necessary to establish one VCC (or VPC) used only by foreground traffic, and a number of VCCs or VPCs used only by background traffic. Then, the background traffic is generated. Characteristics of background traffic are described in section 3.2.6. When flow of the background traffic has been established, the foreground traffic is generated. Characteristics of foreground traffic are specified in section 3.2.5. After the steady

state flow of foreground traffic has been reached the required times and/or delays needed for MIMO latency calculation are recorded for p consecutive frames from the foreground traffic, while the flow of background traffic continue uninterrupted. The entire procedure is referred to as one measurement run.

3.2.5. Foreground traffic

MIMO latency depends upon several characteristics of foreground traffic. These include the type of foreground VCC, service class, arrival patterns, frame length, and input rate.

The foreground VCC can be a permanent or switched, virtual path or virtual channel connection established between ports on the same network module of the switch, or between ports on different network modules, or between ports on different switching fabrics.

For the UBR service class, the foreground traffic consists of equally spaced frames of fixed length. Measurements are performed on AAL payload sizes of 64 B, 1518 B, 9188 B and 64 kB. Variable length frames and other arrival patterns (e.g. self-similar) are under study. ABR service class is also under study.

Input rate of foreground traffic is expressed in the effective bits/sec, counting only bits from AAL payload excluding the overhead introduced by the ATM technology and transmission systems.

The first measurement run is performed at the lowest possible foreground input rate (for the given test equipment). For later measurement runs, the foreground load is increased up to the point when losses in the traffic occur or up to the full foreground load (FFL). FFL is equal to the lesser of the input and the output link rates used by the foreground VCC. Suggested input rates for the foreground traffic are: 0.5, 0.75, 0.875, 0.9375, 0.9687, ..., i.e. $1 - 2^{-k}$, $k = 1, 2, 3, 4, 5, \dots$, of FFL.

3.2.6. Background Traffic

Background traffic characteristics that affect frame latency are the type of background VCCs, connection configuration, service class, arrival patterns (if applicable), frame length (if applicable) and input rate.

Like the foreground VCC, background VCCs can be permanent or switched, virtual path or channel connections, established between ports on the same network module on the switch, or between ports on different network modules, or between ports on different switching fabrics. To avoid interference on the traffic generator/analyzer equipment, background VCCs are established in such way that they do not use the input link or the output link of the foreground VCC in the same direction.

For a SUT with w ports, the background traffic can use $(w-2)$ ports, not used by the foreground traffic, for both input and output. The port with the input link of the foreground traffic can be used as an output port for the background traffic. Similarly, the output port of the foreground traffic can be used as an input port for the background traffic. Overall, background traffic can use an equivalent of $n=w-1$ ports. The maximum background load (MBL) is defined as the sum of rates of all links, except the one used as the input link for the foreground traffic, in the maximum possible switch configuration.

A SUT with $w (=n+1)$ ports is measured for the following background traffic connection configurations:

- n-to-n straight, with n background VCCs, (Figure 3.2.a);
- n-to-($n-1$) full cross, with $n \times (n-1)$ background VCCs. (Figure 3.2.b);
- n-to- m partial cross, $1 \leq m \leq n-1$, with $n \times m$ background VCCs. (Figure 3.2.c);
- 1-to-($n-1$) multicast, with one (multicast) background VCC. (Figure 3.2.e);
- n-to-($n-1$) multicast, with n (multicast) background VCC. (Figure 3.2.d);

Use of the 1-to-($n-1$) multicast and n-to-($n-1$) multicast connection configurations for the background traffic is under study.

The following service classes, arrival patterns (if applicable) and frame lengths (if applicable) are used for the background traffic:

- UBR service class: Traffic consists of equally spaced frames of fixed length. Measurements are performed at AAL payload size of 64 B, 1518 B, 9188 B and 64 kB. This is a case of bursty background traffic with priority equal to or lower than that of the foreground traffic. Variable length frames and other arrival patterns (e.g. self-similar) are for further study.
- CBR service class: Traffic consists of a contiguous stream of cells at a given rate. This is a case of non-bursty background traffic with priority higher than that of the foreground traffic.
- VBR and ABR service classes are under study.

Input rate of the background traffic is expressed in the effective bits/sec, counting only bits from frames excluding the overhead introduced by the ATM technology and transmission systems.

In the cases of n-to-n straight, n-to-($n-1$) full cross and n-to- m partial cross connection configurations, measurement are performed at input rates of 0, 0.5, 0.75, 0.875, 0.9375, 0.9687, ... $(1 - 2^{-k})$, $k = 0, 1, 2, 3, 4, 5, \dots$) of MBL. The required traffic load is obtained by loading each input link by the same fraction of its input rate. In this way, the input rate of background traffic can also be expressed as a fraction (percentage) of input link rates.

3.2.7. Guidelines For Scaleable Test Configurations

Scaleable test configurations for MIMO latency measurements require only one ATM test system with two generator/analyzer pairs. Figure 3.5 presents the test configuration with an ATM switch with eight ports ($w=8$). There are two links between the ATM monitor and the switch, and they are used in one direction by the background traffic and in the another direction by the foreground traffic, as indicated. The other six ($w-2$) ports of the switch are used only by

the background traffic and they have external loopbacks. A loopback on a given port causes the frames transmitted over the output of the port to be received by the input of the same port.

Figure 3.5 shows a 7-to-7 straight connection configuration for the background traffic. The n-to-(n-1) full cross configuration and the n-to-m partial cross configurations can also be similarly implemented.

The test configuration shown assumes two network modules in the switch with ports P0-P3 in one network module and ports P4-P7 in the another network module. Here, the foreground VCC and background VCCs are established between ports in different network modules.

It should be noted that in the proposed test configurations, because of loopbacks, only permanent VCCs or VPCs can be established.

It should also be realized that in test configurations, if all link rates are not identical, it is not possible to generate background traffic (without losses) equal to MBL. The maximum background traffic input rate in those cases equals $(n-1) \times$ lowest link rate. Only in the case where all link rates are identical is it possible to obtain MBL level without losses in background traffic.

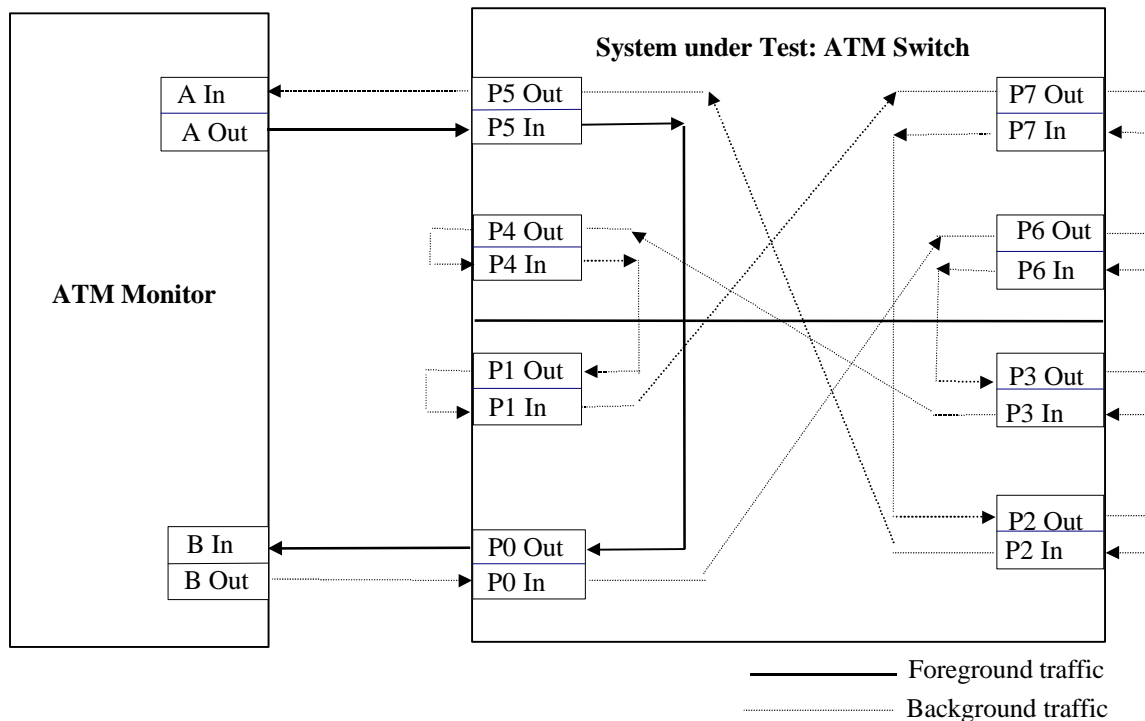


Figure 3.5: A scalable test configuration for measurements of MIMO latency using only two generator analyzer pairs with 8-port switch and 7-to-7 straight configuration for background traffic

If the link rates are different, it is possible to obtain MBL in the n-to-n straight case, but background traffic will have losses. In this case, the foreground traffic should use the lowest rate port in the switch as the input, while the highest rate port in the switch should be used as the

output. The background traffic enters the SUT through the highest rate port and passes successively through ports of decreasing speeds. At the end, the background traffic exits the switch through the lowest rate port.

3.2.8. Reporting results

Reported results should include detailed description of the SUT, such as the number of ports, rate of each port, number of ports per network module, number of network modules, number of network modules per fabric, number of fabrics, the software version and any other relevant information.

Values of the mean and the standard error of MIMO latency are reported along with values of foreground and background traffic characteristics for each measurement run.

The list of foreground and background traffic characteristics and their possible values are now provided:

Foreground traffic:

- type of foreground VCC: permanent virtual path connection, switched virtual path connection, **permanent virtual channel connection**, switch virtual channel connection;
- foreground VCC established: between ports inside a network module, **between ports on different network modules**, between ports on different switching fabrics;
- service class: **UBR**, ABR;
- arrival patterns: **equally spaced frames**, self-similar, random;
- frame length: 64 B, **1518 B**, **9188 B** or 64 kB, variable;
- full foreground load (FFL);
- input rate: **the lowest rate possible for the given test equipment**, and 0.5, **0.75**, 0.875, 0.9375, 0.9687, ..., (i.e., $1 - 2^{-k}$, $k = 1, 2, 3, 4, 5, \dots$) of FFL.

Background traffic:

- type of background VCC's: permanent virtual path connections, switched virtual path connections, **permanent virtual channel connections**, switch virtual channel connections;
- foreground VCCs established: between ports inside a network module, **between ports on different network modules**, between ports on different switching fabrics, some combination of previous cases;
- connection configuration: n-to-n straight, n-to-(n-1) full cross, **n-to-m partial cross with $m = 2, 3, 4, \dots, n-1$** , 1-to-(n-1) multicast, n-to-(n-1) multicast;
- service class: **UBR**, **CBR**, ABR, VBR;
- arrival patterns (when applicable): **equally spaced frames**, self-similar, random;
- frame length (when applicable): 64 B, 1518 B, **9188 B**, 64 kB, variable;
- maximum background load (MBL);
- input rate: **0**, 0.5, 0.75, **0.875**, 0.9375, 0.9687, ... (i.e., $1 - 2^{-k}$, $k = 0, 1, 2, 3, 4, 5, \dots$) of MBL.

Values in bold indicate traffic characteristics for which measurement tests must be performed and for which MIMO latency values must be reported.

3.3. Throughput Fairness

3.3.1. Definition

There are two throughput fairness metrics that are of interest to users:

- *Peak throughput fairness*: this is the fairness at a frame load for the peak throughput.
- *Full-load throughput fairness*: This is the fairness at a frame load for the full-load throughput.

Given n virtual circuits sharing a system (a single switch or a network of switches) and contending for the resources, throughput fairness indicates how far the actual individual allocations are from the ideal allocations. In the simplest case for a total throughput T , the ideal allocation should be T/n . We consider that in the most general case, the ideal allocation is defined by max-min allocation and that allocation is to be used.¹

If the actual measured throughputs of n virtual circuits are found to be $\{T_1, T_2, \dots, T_n\}$, where the ideal throughputs should be $\{\mathfrak{F}_1, \mathfrak{F}_2, \dots, \mathfrak{F}_n\}$, then the throughput fairness of the system under test is quantified by the "fairness index" computed as follows:

$$\text{Fairness index} = (\sum x_i)^2 / (n \times \sum x_i^2)$$

where:

- $x_i = T_i / \mathfrak{F}_i$ is the relative allocation to i th VC.

Note that fairness index is not limited to throughput. It can be applied to other metrics, such as latency. However, extreme unfairness in latency is expected show up as unfairness in throughput and vice versa. Therefore, it is not required to quantify fairness of latency.

3.3.2. Units

This fairness index is dimension-less. The units used to measure the throughput (bits/sec, cells/sec, or frames/sec) do not affect its value. In addition, the fairness index has the following desirable properties:

- It is a normalized measure that ranges between zero and one. The maximum fairness is 100% and the minimum 0%. This makes it intuitive to interpret and present.
- If all x_i 's are equal, the allocation is fair and the fairness index is one.
- If $n-k$ of n x_i 's are zero, while the remaining k x_i 's are equal and non-zero, the fairness index is k/n . Thus, a system which allocates all its capacity to 80% of VCs has a fairness index of 0.8 and so on.

¹ Other policies could be used but must be specified.

3.3.3. Measurement procedures

To measure a peak throughput fairness, the peak throughput for the given SUT has to be first obtained as described in 3.1.4. An experiment for peak throughput fairness is performed by generating the input load corresponding to the peak throughput and recording throughput for each foreground virtual circuit. The experiment is repeated p times. Here p is a parameter and its default value is 30.

To measure a full throughput fairness, the full-load throughput for the given SUT has to be first obtained as described in 3.1.4. Then experiments for full-load throughput fairness are performed similarly to peak throughput fairness experiments.

3.3.4. Statistical Variations

Let F_i be the fairness for the i th throughput experiment, then the mean fairness is computed as follows:

$$\text{Mean Fairness} = (\sum F_i) / p$$

3.3.5. Reporting Results

Values of the mean fairness for peak and lossless throughput (with indication of a number of experiments) are reported along with a detailed description of the SUT, foreground traffic characteristics, and background traffic characteristics (if any), as defined in 3.1.8.

3.4. Frame Loss Ratio

3.4.1. Definition

Frame loss ratio is defined as the fraction of frames that are not forwarded by a system under test (SUT) due to lack of resources. Partially delivered frames are considered lost.

$$\text{Frame loss ratio} = (\text{Input frame count} - \text{output frame count}) / (\text{input frame count})$$

There are two frame loss ratio metrics that are of interest to a user:

- *Peak throughput frame loss ratio*: This is the frame loss ratio at a frame load for the peak throughput.
- *Full-load throughput frame loss ratio*: This is the frame loss ratio at a frame load for the full-load throughput.

3.4.2. Units

The frame loss ratio is expressed as a fraction of input frames.

3.4.3. Measurement Procedures

The frame loss ratio metric is related to the throughput:

$$\text{Frame Loss Ratio} = (\text{Input Rate} - \text{Throughput}) / \text{Input Rate}$$

Thus, no additional experiments are required for frame loss ratios. These can be derived from tests performed for throughput measurements.

3.4.4. Statistical Variations

Since there is only one sample for any of the three frame-level throughput metrics, there is no need for calculation of the means and/or standard deviations of frame loss ratio.

3.4.5. Reporting Results

Values of the frame loss ratios for peak and lossless are reported along with a detailed description of the SUT, foreground traffic characteristics, and background traffic characteristics (if any), as defined in 3.1.8.

3.5. ~~3.5.~~ Maximum Frame Burst Size (MFBS)

3.5.1. ~~Definition~~

Maximum Frame Burst Size (MFBS) is the maximum number of frames that each of source end systems can send at the peak rate through a system under test without incurring any loss. MFBS measures the data buffering capability of the SUT and its ability to handle back-to-back frames.

Many applications and transport layer protocol drivers often present a burst of frames to AAL for transmission. For such applications, Maximum Frame Burst Size provides a ~~a~~ useful indication.

This metric is particularly relevant to UBR service category since the UBR sources are always allowed to send a burst at peak rate. ABR sources may be throttled down to a lower rate if a switch runs out of buffer.

3.5.2. Units

MFBS should be expressed in octets of AAL payload field. This is preferred over number of frames or cells. The former requires specifying the frame size and the latter is not very meaningful for a frame-level metric. Also, number of cells has to be converted to octets for use by AAL users.

~~It may be useful to indicate the frame size for which MFBS has been measured. If MFBS is found to be highly variable with frame size, a number of common AAL payload field sizes such as 64 octets, 536 octets, 1518 octets, and 9188 octets may be used (exact sizes are for further study).~~

3.5.3. Statistical Variations

~~The number of frames sent in the burst is increased successively until a loss is observed on any VC. The maximum number of frames that can be sent without loss are reported as MFBS. The tests should be repeated NRT times. The average of NRT repetitions is reported as the MFBS for the system under test.~~

3.5.4. Traffic Patterns

~~The MFBS is measured for n-to-1 traffic pattern specified in Section 3.1.4. Optionally, it can be measured for other traffic patterns also. The value obtained for n-to-1 pattern is expected to be smaller than that for other patterns.~~

3.5.5. Guidelines For Using This Metric

~~To be specified: There is no need for obtaining more than one sample for MFBS. Consequently, there is no need for calculation of the means and/or standard deviations.~~

3.5.4 Measurement Procedure and MFBS Calculation

The MFBS is measured for k-to-1 connection configuration as specified in Section 3.1.5. Thus, k VCCs (or VPCs) are established through the SUT. All k+1 links are of the same rate.

The measurement procedure may require a number of tests. Each test includes simultaneous generations of fixed length bursts of back-to-back cells through all k VCCs (or VPCs) and counting of all cells transmitted by the SUT. If there is no loss of cells, the length of bursts is increased, but if there is a loss, the length of bursts is decreased. In both case, the next test is performed with the new burst length. The procedure is finished, when the maximum cell burst size (MCBS) is found. MCBS is the maximum burst length for which there is no cell loss.

Tests are conducted without any background traffic.

Given MCBS, one can calculate the maximum integral number of back-to-back frames of a given size, which can be sent into the SUT of the given connection configuration and delivered by the SUT without any loss. This integral number then converted to octets of AAL payload field to obtain the Maximum Frame Burst Size (MFBS)

3.5.5 Reporting Results

Reported results should include detailed description of the SUT, such as the number of ports, rate of each port, number of ports per network module, number of network modules, number of network modules per fabric, number of fabrics, the software version and any other relevant information.

The value for MFBS is reported for each link rate supported by the SUT along with traffic characteristics. The list of traffic characteristics and their possible values are as follows:

- type of VCCs: permanent virtual path connections, switched virtual path connections, **permanent virtual channel connections**, switch virtual channel connections;
- VCCs established: between ports inside a network module, **between ports on different network modules**, between ports on different fabrics, some combination of previous cases;
- connection configuration: 2-to-1;
- frame length: **64 B, 1518 B, 9188 B, 64 kB**;

Values in bold indicate traffic characteristics for which measurement tests must be performed and for which MFBS values must be reported.

3.6. Call Establishment Latency

3.6.1. Definition

For short duration VCs, call establishment latency is an important part of the user perceived performance. Informally, the time between submission of a call setup request to a network and the receipt of the connect message from the network is defined as the call establishment latency. The time lost at the destination while the destination was deciding whether to accept the call is not under network control and is, therefore, not included in call setup latency (See Figure 3.6).

Thus, the sum of the latency experienced by the setup message and the resulting connect message is the call setup latency.

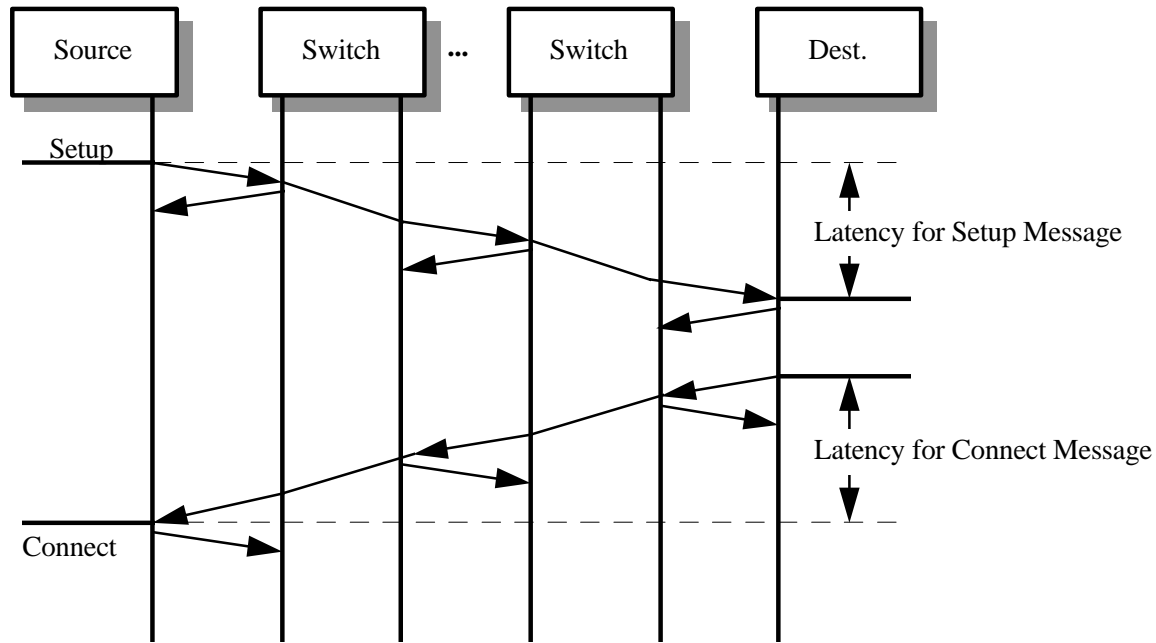


Figure 3.6: Call establishment

The main problem in measuring these latencies is that both these messages span multiple cells with intervening idle/unassigned cells. Unlike X.25, frame relay, and ISDN networks, the messages in ATM networks are not contiguous. Therefore, the MIMO latency metric defined in Section 3.2 is used². Thus,

$$\text{Call Establishment Latency} = \text{MIMO Latency for SETUP message} \\ + \text{MIMO latency for the corresponding Connect message}$$

Recall that the MIMO latency for a frame is defined as the minimum of last-bit-in-to-last-bit-out (LILO) and the difference of first-bit-in-to-last-bit-out (FILO) and normal frame output time (NFOT).

$$\text{MIMO Latency} = \text{Min}\{\text{LILO}, \text{FILO}-\text{NFOT}\}$$

3.6.2. Units

Call establishment latency is measured in units of time.

3.6.3. Configurations

The call establishment latency as defined above applies to any network of switches. In practice, it has been found that the latency depends upon the number of switches and the number of PNNI group hierarchies traversed by the call. It is expected that measurements will be conducted on

² Applies only if cells of setup and connect messages are contiguous at the input port.

multiple switches connected in a variety of ways. In all cases, the number of switches and number of PNNI group hierarchies traversed should be indicated.

The simplest configuration is that of a single switch connecting both the source and the destination end systems. Further configurations are for further study.

3.6.4. Statistical Variations

The latency measurement is repeated NRT times. Each time a different node pair is selected randomly as the source and destination end system. The average and standard error of NRT such measurements is reported. For a single n-port switch it is expected that all n ports are equally probable candidates to be source and destination end-system.

3.6.5. Guidelines For Using This Metric

To be specified.

3.7. Application Goodput

Application-goodput captures the notion of what an application sees as useful data transmission in the long term. Application-goodput is the ratio of packets(frames) received to packets(frames) transmitted over a measurement interval.

The application-goodput (AG) is defined as:

$$AG = \frac{\text{Frames Received in Measurement Interval}}{\text{Frames Transmitted in Measurement Interval}}$$

where Measurement Interval is defined as the time interval from when a frame was successfully received to when the frame sequence number has advanced by n.

Note that traditionally goodput is measured in bits per sec. However, we are interested in a non-dimensional metric and are primarily interested in characterizing the useful work derived from the expended effort rather than the actual rate of transmission. While the application-goodput is intended to be used in a single-hop mode, it does have meaningful end-to-end semantics over multiple hops.

Notes:

- This metric is useful when measured at the peak load which is characterized by varying the number of transmitted frames must be varied over a useful range from 2000 frames per second (fps) through 10000 fps at a nominal frame size of 64 bytes. Frame sizes are also varied through 64 bytes, 1518 bytes, and 9188 bytes to represent small, medium, and large frames respectively. Note that the frame sizes specified do not account for the overhead of accommodating the desired frame transmission rates over the ATM medium.
- Choose the measurement interval to be large enough to accommodate the transmission of the largest packet (frame) over the connection and small enough to track short-term excursions of the average goodput.
- It is important not to include network management frames and/or keep alive frames in the count of received frames.
- There should be no changes of frame handling buffers during the measurement.
- The results are to be reported as a table for the three different frame sizes.

3.7.1. Guidelines For Using This Metric

To be specified.

4. References

- [1] ~~ATM Forum Technical Committee, "Introduction to ATM Forum Test Specifications", at-
test-0022.00, December 1994.~~ ITU-T Recommendation I.356, "B-ISDN ATM layer cell
transfer performance".
IUT-T Study Group 13, Geneva, 1996.
- [2] ~~S. Sathaye (editor), "ATM Forum The ATM Forum, Traffic Management Specification
Version 4.0", AF-TM-95-0013R10 Straw Vote, February~~ April 1996.

Editorial note: The current text of Appendix A is outdated, because it has not been adjusted to reflect the new MIMO latency definition given in 3.2.1. It is expected that the new Appendix A will be presented at September '97 ATMF meeting.

~~Appendix A: MIMO Latency~~

~~A.1. Definition~~

~~MIMO latency (Message In Message Out) is a general definition of the latency that applies to an ATM switch or a group of ATM switches and it is defined as follows:~~

$$\text{MIMO latency} = \min \{ \text{LILO latency}, \text{FILO latency} - \text{NFOT} \}$$

~~where:~~

- ~~-LILO latency = Time between the last bit entry and the last bit exit~~
- ~~-FILO latency = Time between the first bit entry and the last bit exit~~
- ~~-NFOT = Nominal Frame Output Time = FIT × Input Rate/Output Rate~~
- ~~-FIT = Frame Input Time = Time between the first bit entry and the last bit entry~~

~~Note that for contiguous frames on input:~~

$$\text{Frame Input Time} = \text{Frame Size} / \text{Input rate}$$

~~and then it follows:~~

$$\begin{aligned} \text{NFOT} &= \text{Frame Size} / \text{Input Rate} \times \text{Input Rate} / \text{Output Rate} \\ &= \text{Frame Size} / \text{Output rate} \end{aligned}$$

~~The following is an equivalent definition for MIMO Latency:~~

$$\text{MIMO latency} = \begin{cases} \text{LILO latency} & \text{if input rate} \leq \text{output rate} \\ \text{FILO latency} - \text{NFOT} & \text{if input rate} \geq \text{output rate} \end{cases}$$

~~Note that for input rate = output rate:~~

~~MIMO latency = LILO latency = FILO latency - NFOT~~ [3] ATM Forum, Introduction to ATM Forum Test Specifications, Version 1.0, December 1994.

Appendix A: Defining Frame Latency on ATM Networks

A.2.A.1. Introduction

In the rest of the Appendix we justify the MIMO latency definition. In this section, we start with a single bit case and a simple contiguous frame case. Then we systematically consider contiguous frame cases and then discontinuous frame cases in the ATM environment in Section A.3 and Section A.4, respectively. This appendix discusses delays, and the performance metrics characterizing them, that an ATM network introduces to its frames. We are concerned with delays caused by node processing, such as switching and routing, as well as queuing delays that may be introduced by the background traffic and inter-network link transmission delays. On the other hand, transmission delays introduced by input and output links of a network component should not be attributed to the component. Also, note that characteristics of traffic generators (e.g., host speeds) should not affect network performance metrics. The discussion in this Appendix applies to any network element (including switches, multiplexors, inverse-multiplexors, wires) or any combination of such network elements. Although we frequently use the term "switch," the discussion applies equally well to other network elements, whole networks, or parts of networks.

For a single bit case (see Figure A.1), the latencyIn the case of a single bit, the switch (network) delay is generally defined as the time between the instant the bit enters the system ~~to~~and the instant the bit exits from the system. Figure A.1 illustrates the single-bit latency.

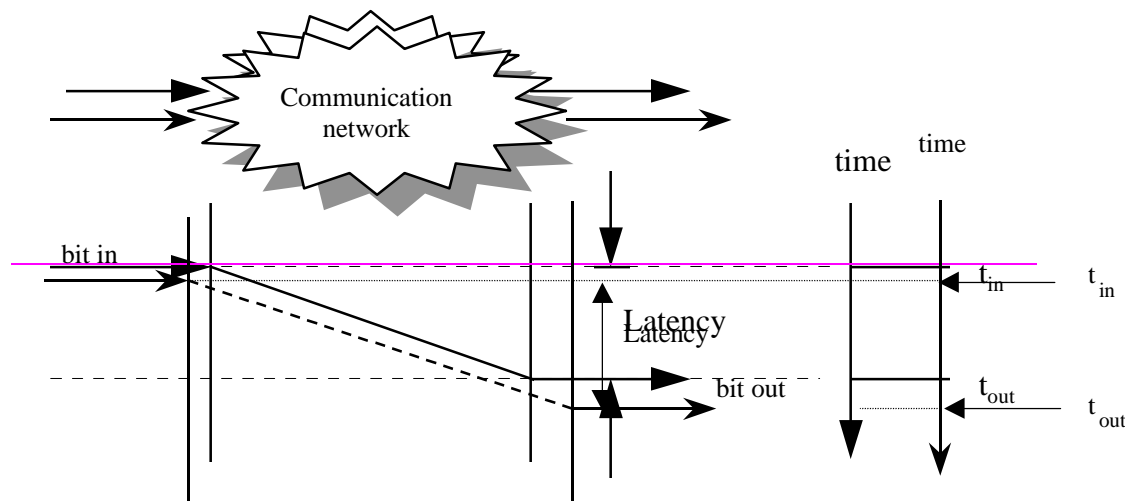


Figure A.1: Latency for a single bit

For multi-bit frames, ~~there are several possible definitions. Consider the case of contiguous frames, i.e. all bits of the frames are sent (on input) and delivered (on output) contiguously without any gap between bits. In this case, latency can be defined in the usual way to define the frame latency introduced by a switching device is to apply~~ one of the following four ~~ways:~~ definitions:

- 1. ~~FIFO~~ latency: Time between the first-bit entry and the first-bit exit
- 2. ~~LIFO~~ latency: Time between the last-bit entry and the last-bit exit
- 3. ~~FILO~~ latency: Time between the first-bit entry and the last-bit exit
- 4. ~~LIFO~~ latency: Time between the last-bit entry and the first-bit exit

~~If the input link and the output link are of the same speed and frames are contiguous (see Figure A.2), FIFO and LIFO latencies are identical. In this case FILO and LIFO latencies can be computed from FIFO (or LIFO) latency given the frame length and input rate or output rate:~~

$$\text{FILO} = \text{FIFO} + \text{Frame Size}/\text{Input rate} = \text{FIFO} + \text{Frame Size}/\text{Output rate}$$

$$\text{LIFO} = \text{FIFO} - \text{Frame Size}/\text{Input rate} = \text{FIFO} - \text{Frame Size}/\text{Output rate}$$

~~It is clear that FIFO (or LIFO) is a preferred metric in this case since it is independent of the frame length while FILO and LIFO would be different for each frame size. That is one of reasons why we shall not further consider FILO and LIFO.~~

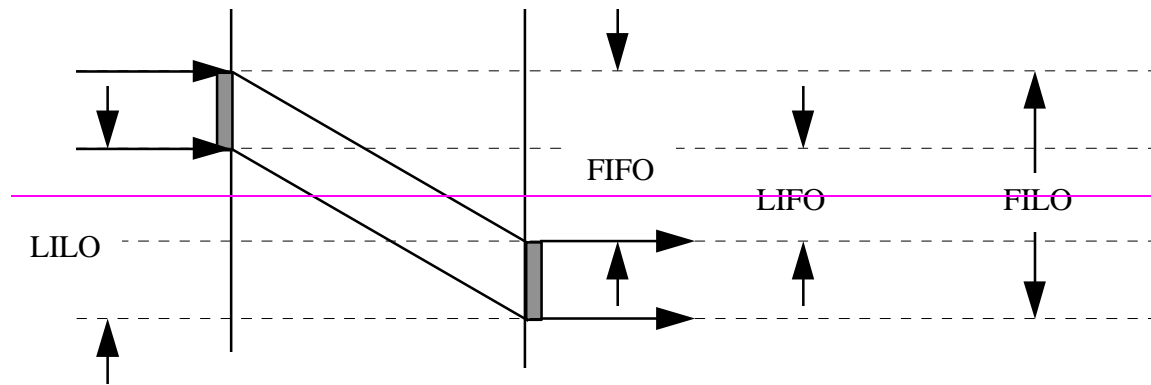


Figure A.2: Latency for multi-bit frames, input rate = output rate

Unfortunately, none of the above four metrics apply to an ATM network (or switch) latency since:

the input and output link may be of different speeds, and

the frames are not always sent in (on input) or delivered out (on output) contiguously, i.e., there may be idle times between cells of a frame either on input and/or output.

In the following, we consider first contiguous frames and then discontinuous frames in an ATM network. We compare FIFO, LIFO and MIMO metrics and show that MIMO is the correct metric in all cases while other metrics apply to some cases but give incorrect results in others.

A.3. Contiguous Frames

In this section we consider cases where frames on input as well as on output are contiguous, i.e., without any gaps between their cells. Depending upon the relative magnitude of input and output rates and the delay through the switch, there are six possible cases. These cases and the applicability of the three metrics are shown in Table A.1.

Table A.1: Applicability of Various Latency Definitions For Contiguous Frames

No.	Case	FIFO	LILO	MIMO
-----	------	------	------	------

Figure A.2 illustrates the usual frame latencies (FIFO, LILO, FILO and LIFO) in a scenario with a contiguous frame on both input and output, passing through the given communication network which has an input link rate lower than the output link rate.

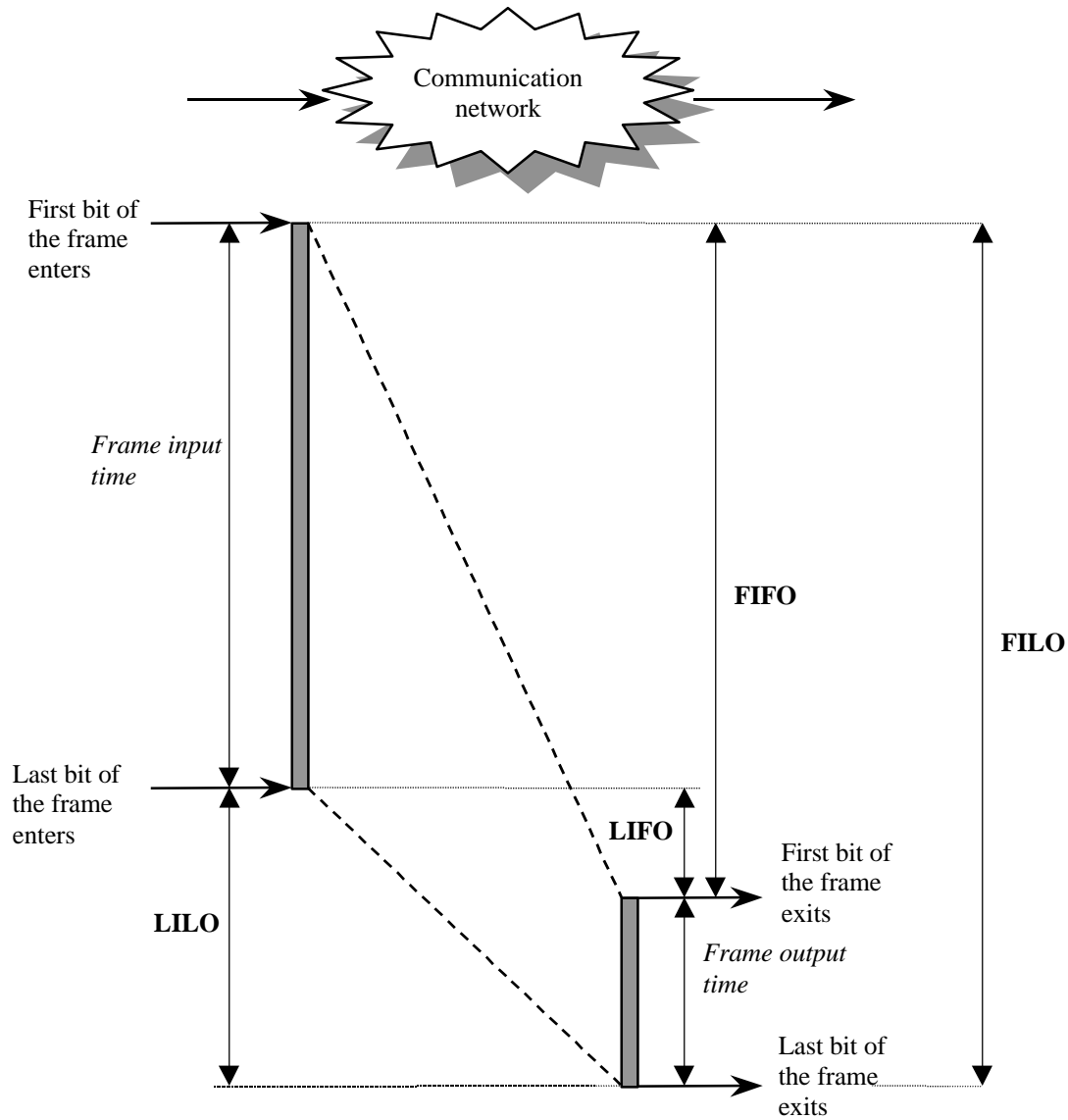


Figure A.2: Usual frame latencies

Unfortunately, as it will be shown later, none of the four above metrics is appropriate for an ATM network. In this appendix, we introduce and justify a new latency metric called "MIMO" latency. This new latency metric applies to any type of network where the frames may be contiguous or discontinuous, although our primary interest is an ATM environment. To define the MIMO latency, we introduce the concept of a "zero-delay" switch, which is in some sense

the best a switch can do. The delay of any other switch is defined as the latency over and above the delay of a zero-delay switch.

This appendix is organized as follows. In the next section, we analyze why the usual frame latencies are not appropriate in an ATM environment. We introduce the MIMO latency in Section A.3. In Section A.4, we introduce the concept of a zero-delay switch and its processing of individual cells and contiguous frames. We discuss delays introduced to discontinuous frames passing through a zero-delay switch in Section A.5. Section A.6 presents the method for calculating the FILO latency of frames passing through a zero-delay switch. An equivalent, but easier to use, definition of MIMO latency is developed in Section A.7. Section A.8 of this appendix presents derivations of expressions for MIMO latency calculation based on cell-level data. The last section discusses the user perceived delay in data communication networks.

A.2. Usual Frame Latencies as Metrics for ATM Switch Delay

An ATM switch has to deal with both contiguous and discontinuous frames. This is because ATM switches do cell-switching, i.e., an ATM switch may transmit a received cell of any frame without first waiting for other cells of that frame to arrive. Thus, frames sent and received in an ATM environment are not always contiguous. Even if the input frame is contiguous, the ATM switch may transmit discontinuous frames, i.e., it may introduce idle periods, unassigned cells and/or cells of other frames between cells of the frame.

The above factors make the usual frame latency metrics inappropriate for ATM switches. In this section, we show why LIFO, FIFO and FILO latencies are not appropriate metrics for an ATM switch. Later in this appendix, we shall show that LILO latency is an appropriate metric only in certain cases.

LIFO Latency

In [1], the delay in a packet-switching network is defined as the time between a "packet entry event" and a "packet exit event." A packet entry event is defined to occur at the time when the last bit of the frame enters a network, while a packet exit event is defined to occur when the first bit of the frame exits a network. This is equivalent to LIFO latency, which is considered as an appropriate metric for store-and-forward packet-switching networks because:

- packets are contiguous on both input and output and
- it is accepted that the transmission delay during packet input is an intrinsic delay for a store and forward device, for which the switch should not be penalized.

Newer networking devices are not necessarily store-and-forward. Some of them are cut-through devices that start emitting the frame before it is received completely. Figure A.3 illustrates the case of a frame passing through a cut-through switching device with three of the four usual latencies indicated. LIFO latency is not shown because the first bit of the frame exits before the last bit of the frame enters and the LIFO latency is negative. This is a common case with cut-through devices. Thus, LIFO latency is not a good indicator of the switch delay for any cut-

through type device, and as such it is inappropriate for an ATM environment, where cut-through forwarding of frames is the normal mode of operation.

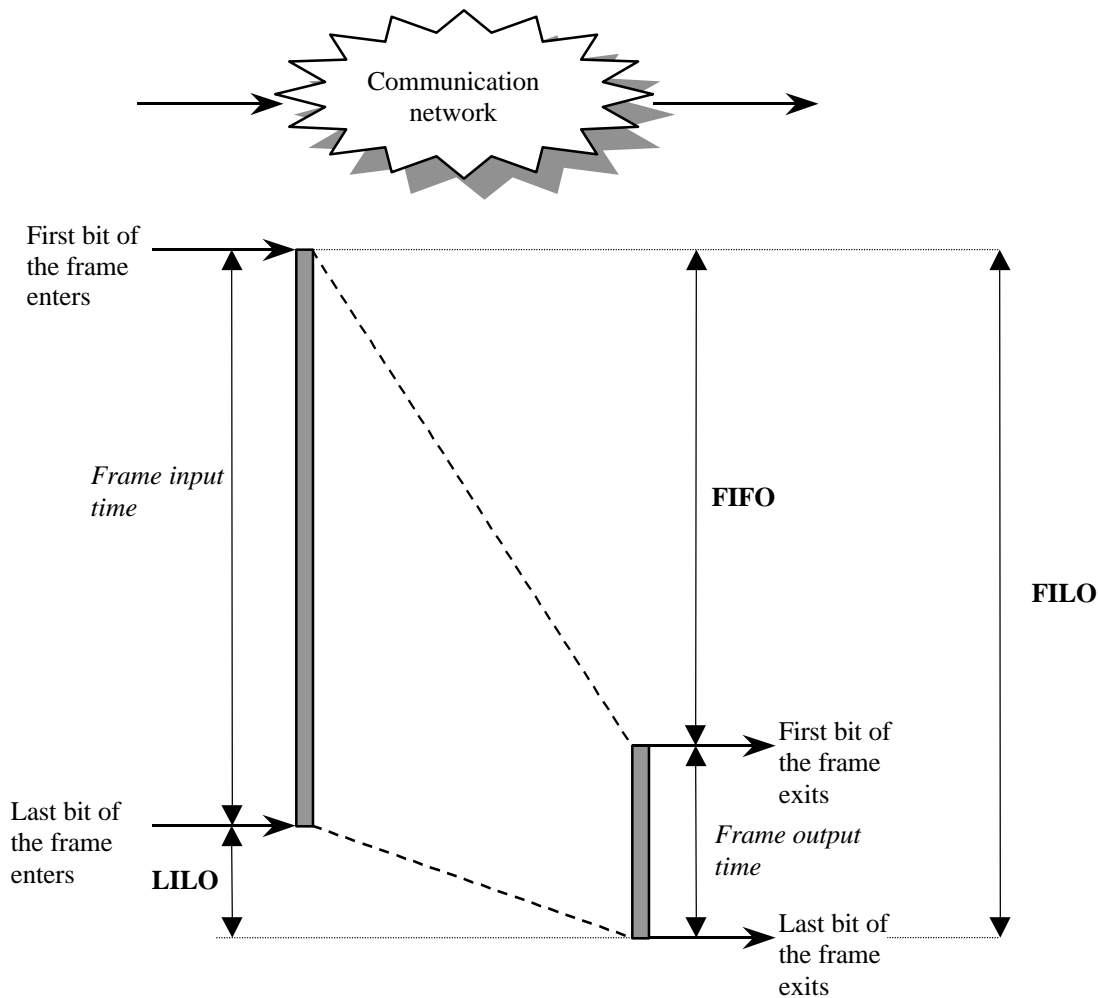


Figure A.3: Latencies of a frame passing through a cut-through switching device

FIFO Latency

It is interesting to note that [2] provides a LIFO latency definition as the delay metric for store and forward switching devices, as well as a FIFO latency definition for bit forwarding devices (i.e. cut-through switching devices). The introduction of FIFO latency as a delay metric is an attempt to avoid negative values for the delay through cut-through devices.

While FIFO latency may provide meaningful results if the frames are continuous, it may provide useless results if the frames are discontinuous. It is possible to have a very low FIFO delay while delays for the other parts of the frames are high. Again, since frames on ATM networks are generally discontinuous, FIFO latency is not a meaningful measure of frame latency. Figure A.4 illustrates this point.

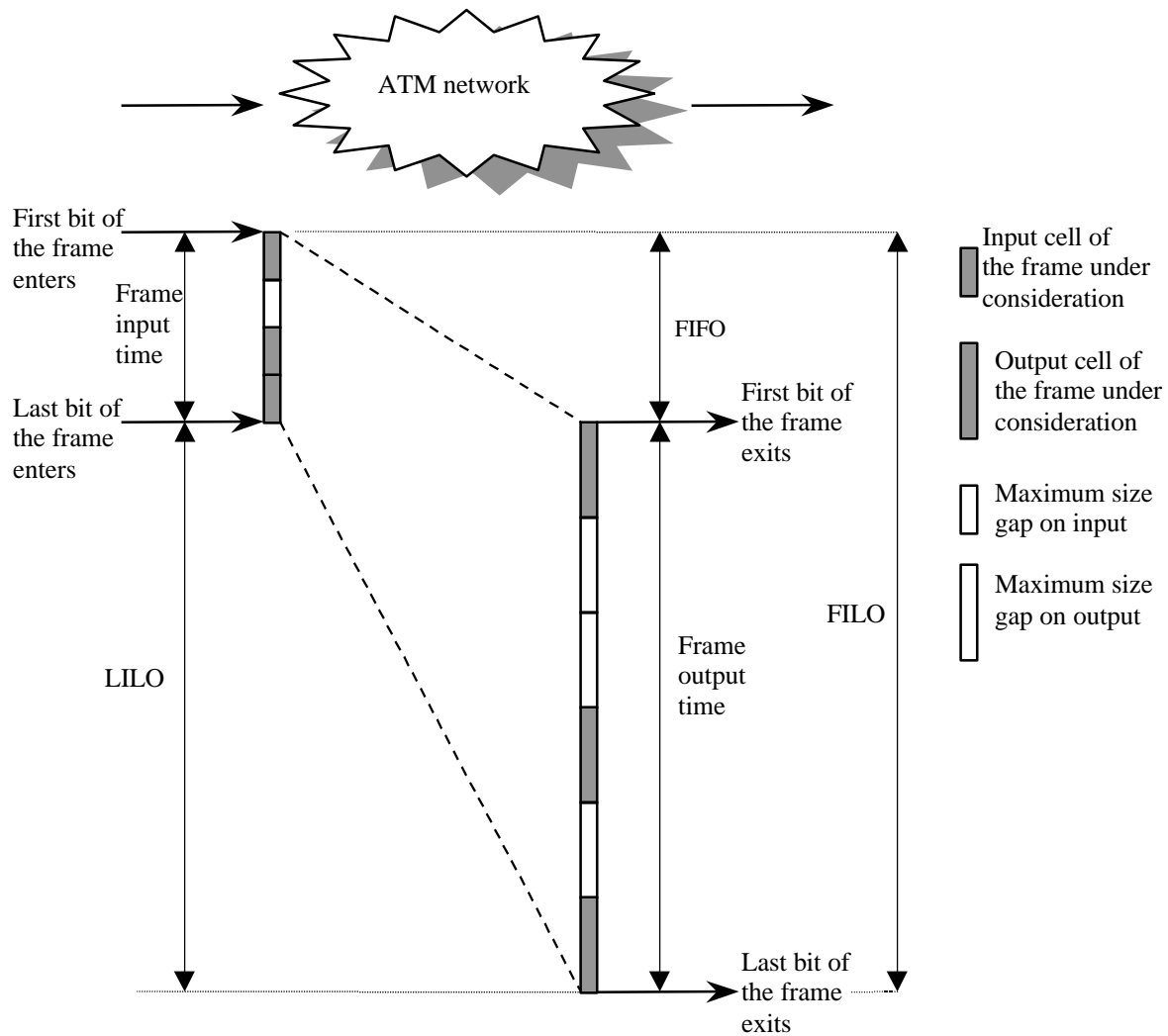


Figure A.4: Usual Latencies in an ATM Environment

In this case, the frame consists of 3 cells passing through an ATM switch with the input link rate higher than the output link rate. The frame is discontinuous on both input and output. The last cell is delayed considerably more than what FIFO latency would indicate.

It is possible to have one pattern of idle periods or unassigned cells (positions and a number of them) on the input of a given frame, and a completely different pattern on the output of the same frame. Note that it is also possible for a switch to remove idle periods or unassigned cells from the input, “transmitting” fewer of them on output, as we shall illustrate later.

In Figure A.4, as well as in the rest of this appendix, an unassigned cell, an idle period or a cell of another frame between cells of a given frame is indicated as a gap. In Figure A.4 the frame on input has a one-cell gap after the first cell of the frame, followed by the two remaining cells of

the frame. On output, there is a two-cell gap after the first cell and then a one-cell gap between the second and the third cell of the frame.

From Figure A.4, it can be observed that it is possible for a switch to have a small FIFO latency if the first cell of a frame is transmitted quickly. However, if the later cells are delayed considerably, the receiver is not able to assemble the frame. FIFO latency does not reflect the expansion and compression of gaps on output. This is why FIFO latency is not an appropriate delay metric for switches in the ATM environment.

FILO Latency

From any of the previous three figures it can be noted that the relationship between FILO and LILO latency is as follows:

$$\text{FILO latency} = \text{LILO latency} + \text{Frame Input Time}$$

Although FILO and LILO latencies are related (one can compute one given the other), LILO latency is a preferred metric since it is independent of frame input time. FILO latency is different for different frame input patterns. Suitability of LILO and FILO metrics under various circumstances is discussed after introducing MIMO latency in the next section.

A.3. MIMO Latency Definition

MIMO latency (Message-In Message-Out) is a performance metric that defines the delay introduced upon a frame passing through a switch (or any other network component). When applied to a single switch, the MIMO latency accounts only for delays introduced by the switch (because of switching and other processing) and is independent of the frame input time, output transmission time, and other physical layer delays introduced on the input and output links.

Succinctly, MIMO latency is defined as follows:

$$\text{MIMO latency} = \text{FILO latency} - \text{NFOT}$$

where

- NFOT (Nominal Frame Output Time) is equal to the FILO latency of a given frame passing through a zero-delay switch.

We define a zero-delay switch as a switch that handles incoming frames in such way that they are transmitted on the output link without any unnecessary time consuming processing.

The above definition implies that MIMO latency is the difference between the measured FILO latency of a frame passing through the given switch and the FILO latency of the same frame passing through a zero-delay switch. As defined, MIMO latency has the desired property of always being positive (or zero for a zero-delay switch).

The MIMO latency is not limited to switches. It applies to all types of communication devices, including repeaters, multiplexers, (store-and-forward or cut-through) bridges, routers, ATM switches, wires, or any combination of these. MIMO latency also accounts for discontinuous frames on the input and/or output. For discontinuous frames on input, gaps may include idle periods, unassigned cells and/or cells from other frames. For discontinuous frames on output, it is assumed that there are no cells from other frames inserted between the cells of the given frame, but idle periods or unassigned cells are allowed. It should be realized that the last assumption does not present a limitation for measurements in benchmarking environments.

In the following two sections, we explore the concept of a zero-delay switch in depth.

A.4. Cell and Contiguous Frame Latency Through A Zero-Delay Switch

Figure A.5 illustrates the latency that one-bit frame would experience while passing through a zero-delay switch. As expected, a zero-delay switch should start transmission on the output link as soon as the bit arrives on the input link. Thus, the latency of a single bit through a zero-delay switch is equal to zero. A wire of a zero length is one example of a zero-delay switch.

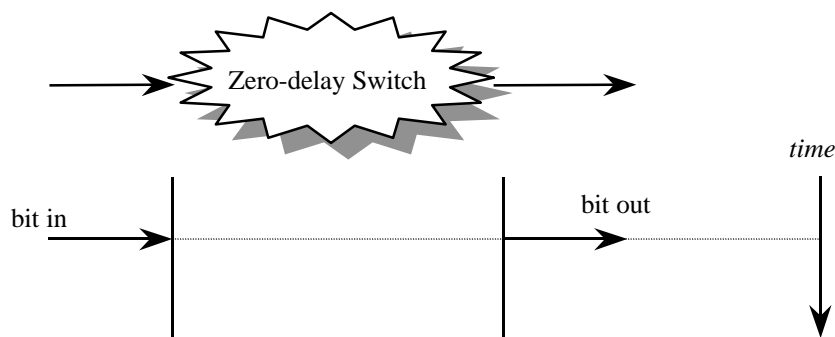


Figure A.5: Latency of one bit passing through the zero delay switch

Figure A.6 illustrates how a zero-delay switch would handle a cell consisting of multiple bits. The desired performance depends upon the relationship between the input and output link rates. In the case when the input link rate is equal to the output link rate, as presented in Figure A.6a, a zero-delay switch transmits each bit as soon as it arrives. Thus, each bit of the cell experiences zero latency in a zero-delay switch. A zero-length wire is one example of a zero-delay device.

Figure A.6b illustrates the case when the input link rate is higher than the output link rate. In this case, outputting (transmitting) a bit takes longer than inputting it. The zero-delay switch can transmit only the first bit as soon as it is received. The other bits of the cell can not be transmitted immediately as they arrive, because the transmission of all previously received bits has not yet finished. Bits at the end of the cell wait longer than bits at the beginning. Thus, a zero-delay switch in this situation should be intelligent enough to do appropriate buffering of incoming bits. A zero-length wire with a FIFO buffer is an example of a zero-delay device that can handle inputs faster than the output.

Figure 6c illustrates the case when the input link rate is lower than the output link rate. A zero-delay switch does not start transmission of the first bit immediately after it is received, but after an appropriate delay. Bits at the beginning of the cell are delayed more than bits at the end, with larger delays for slower output link rates. Only the last bit of a cell has no delay and it is transmitted immediately upon its arrival. Thus, a zero-delay switch would be intelligent enough to avoid under-runs by appropriately delaying the transmission of incoming bits. A zero-length wire with an "intelligent" FIFO buffer is an example of such a zero-delay device.

It should be realized that the illustrations in Figure A.6 apply not only to cells, but also to contiguous frames passing through a zero-delay switch.

Note that a repeater can be considered as a zero-delay switch with input link rate equal to output link rate. Thus, Figure A.6a illustrates how a repeater handles incoming frames.

Also, note that a multiplexer, with n links on input and the output link capacity equal to the sum of input link capacities, can be considered as a zero-delay switch with input link rate lower than output link rate. For a multiplexer with two input links of rates equal to one half of the output link rate, Figure A.6c illustrates how the multiplexer would handle incoming frames. Similarly, a demultiplexer can be considered as a zero-delay switch with an input-link rate higher than the output-link rate. Figure A.6b illustrates operation of a two-output demultiplexer.

Based on Figure A.6, Table 1 provides (qualitative) indications for the four usual frame latency metrics applied to a zero-delay switch. None of the latencies has a zero value in all three cases, as it should be for the latency of a frame passing through a zero-delay switch.

Table 1: Usual Latencies Applied to a Zero-Delay Switch

1aC	Input rate = Output rate, zero-delay switch	√	√	√
1bC	Input rate = Output rate, nonzero-delay switch	√	√	√
2aC	Input rate < Output rate, zero-delay switch	×	√	√
2bC	Input rate < Output rate, nonzero-delay switch	×	√	√
3aC	Input rate > Output rate, zero-delay switch	√	×	√
3bC	Input rate > Output rate, nonzero-delay switch	√	×	√

√ ⇒ The metric gives a valid result.

× ⇒ The metric gives an invalid result.

As indicated above, we consider a zero-delay switch and a nonzero-delay switch. The cases with a zero-delay switch are especially useful to verify the validity of a latency definition, because the switch delay is known in advance (equal to zero).

It should be noted that in all cases for contiguous frames on input and on output, the following relation always holds:

$$\text{FIFO} = \text{FILO} - \text{Frame size/Output rate},$$

and this relation will be used in this section.

Case 1aC: Contiguous Frames, Input rate = Output rate, Zero-Delay Switch

Figure A.1aC shows the flow in this case.

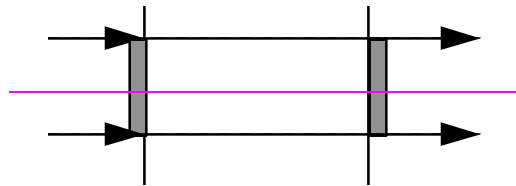


Figure A.1aC: Contiguous frames, Input rate = Output rate, Zero-delay switch

In this case, the bits appear on the output as soon as they enter on the input. Here we have:

-FIFO = 0, correct

-LILO = 0, correct

-MIMO = $\min\{\text{LILO}, \text{FILO} - \text{Frame Size/Output rate}\} = \min\{0, \text{FIFO}\} = 0$, correct

Case 1bC: Contiguous Frames, Input rate = Output rate, Nonzero-Delay Switch

Figure A.1bC shows the flow in this case.

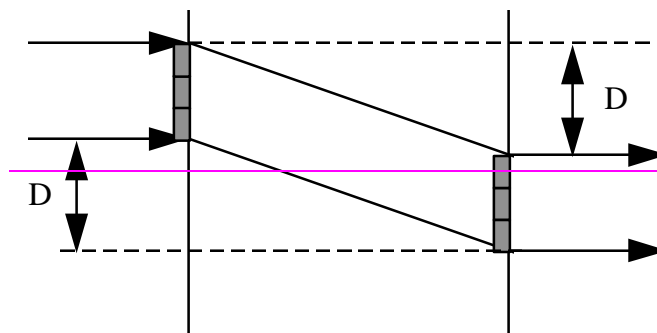


Figure A.1bC: Contiguous frames, Input rate = Output rate, Nonzero-delay switch

In this case, the switch latency D is determined by a delay of the first bit (or the last bit). Here we have:

-FIFO = D , correct

-LIFO = D, correct

-MIMO = $\min\{\text{LIFO}, \text{FIFO} - \text{Frame Size}/\text{Output Rate}\} = \min\{D, \text{FIFO}\} = D$, correct

Case 2aC: Contiguous Frames, Input rate < Output rate, Zero-Delay Switch

Figure A.2aC shows the flow in this case:

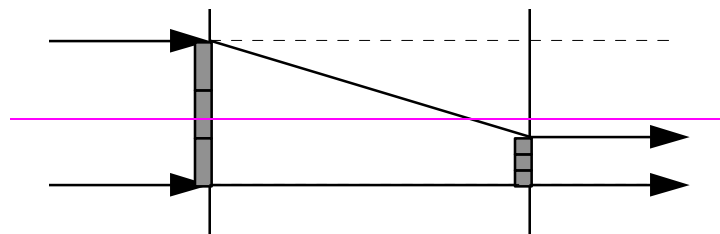


Figure A.2aC: Contiguous frames, Input rate < Output rate, Zero-delay switch

In this case, a contiguous frame on the output is possible only if the transmission of incoming bits is scheduled such that there will not be any buffer underflow until the last bit. Here we have:

-FIFO > 0, incorrect; Note that FIFO may change with changing output rate (while not changing the switch latency). So, FIFO does not correctly represent the switch latency.

-LIFO = 0, correct

-MIMO = $\min\{\text{LIFO}, \text{FIFO} - \text{Frame Size}/\text{Output Rate}\} = \min\{0, \text{FIFO}\} = 0$, correct

Case 2bC: Contiguous Frames, Input rate < Output rate, Nonzero-Delay Switch

Figure A.2bC shows the flow in this case:

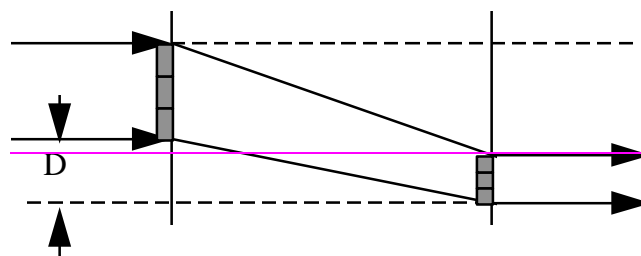


Figure A.2bC: Contiguous frames, Input rate < Output rate, Nonzero-delay switch

In this case, the switch latency D is determined by a delay of the last bit. Here we have:

-FIFO > D, incorrect; As in Case 2aC, FIFO may change with changing output rate (without changing the switch latency). So, FIFO does not correctly represent the switch latency.

-LIFO = D, correct

-MIMO = $\min\{\text{LIFO}, \text{FIFO} - \text{Frame Size}/\text{Output Rate}\} = \min\{D, \text{FIFO}\} = D$, correct

Case 3aC: Contiguous Frames, Input rate > Output rate, Zero-Delay Switch

Figure A.3aC shows the flow in this case.

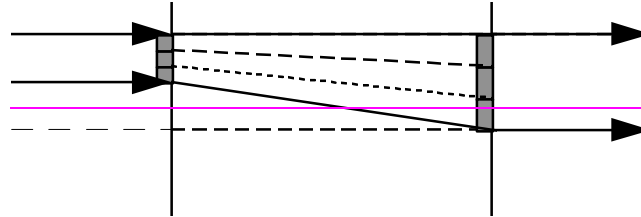


Figure A.3aC: Contiguous frames, Input rate > Output rate, Zero-delay switch

In this case, only the first bit on the input appears immediately on the output, and other bits have to be buffered, because the input rate is larger (more bits are input) than the output rate (fewer bits are output). Here we have:

- FIFO = 0, correct
- LIFO > 0, incorrect; Note that LIFO may change with changing the output rate and not changing the switch otherwise. So, LIFO does not correctly represent the switch latency.
- MIMO = $\min\{\text{LIFO}, \text{FIFO} - \text{Frame Size}/\text{Output Rate}\} = \min\{\text{LIFO}, \text{FIFO}\} = 0$, correct

Case 3bC: Contiguous Frames, Input rate > Output rate, Nonzero-Delay Switch

Figure A.3bC shows the flow in this case.

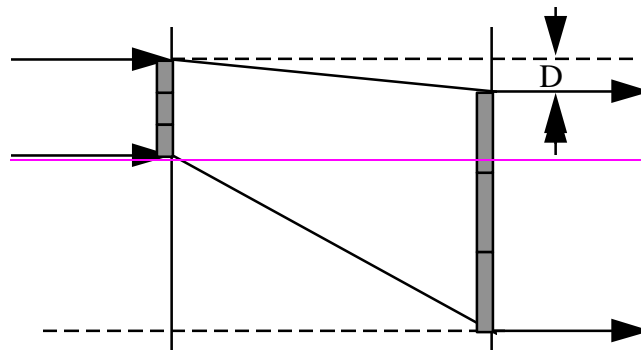


Figure A.3bC: Contiguous frames, Input rate > Output rate, nonzero-delay switch

In this case, the switch latency D is determined by a delay of the first bit. Here we have:

- FIFO = D, correct
- LIFO > D, incorrect; As in Case 3aC, LIFO may change with changing the output rate and not changing the switch otherwise. So, LIFO does not correctly represent the switch latency.
- MIMO = $\min\{\text{LIFO}, \text{FIFO} - \text{Frame Size}/\text{Output Rate}\} = \min\{\text{LIFO}, \text{FIFO}\} = D$, correct

A.4. Discontiguous Frames

In this section we consider cases where frames on input as well as on output are discontiguous, i.e. there are gaps between cells of frames. Depending upon the number of gaps on input and output, we have three possibilities:

- The number of gaps on output is same as that on input. This is the case of *no change in gaps*.
- The number of gaps on output is more than that on input. This is the case of *expansion of gaps*.
- The number of gaps on output is less than that on input. This is the case of *compression of gaps*.

It should be noted that cases with contiguous frames on input and/or output are special cases of discontiguous frames with no gaps.

The nine cases and the applicability of the three metrics (FIFO, LIFO and MIMO) to those cases are shown in Table A.2. Each case includes a case with a nonzero delay switch and (if possible) a case with a zero delay switch.

Table A.2: Applicability of Various Latency Definitions For Discontiguous Frames

No.	Case	FIFO	LIFO	MIMO
1aD	Input rate = Output rate, no change in gaps	✓	✓	✓
1bD	Input rate = Output rate, expansion of gaps	✗	✓	✓
1cD	Input rate = Output rate, compression of gaps	✗	✓	✓
2aD	Input rate < Output rate, no change in gaps	✗	✓	✓
2bD	Input rate < Output rate, expansion of gaps	✗	✓	✓
2cD	Input rate < Output rate, compression of gaps	✗	✓	✓
3aD	Input rate > Output rate, no change in gaps	✓	✗	✓
3bD	Input rate > Output rate, expansion of gaps	✗	✗	✓

3eD	Input rate > Output rate, compression of gaps	*	*	√
-----	---	---	---	---

√ ⇒ The metric gives a valid result.
 * ⇒ The metric gives an invalid result.

Case 1aD: Discontiguous Frames, Input rate = Output rate, No Changes in Gaps

Figure A.1aD shows the flow for a zero-delay switch and a nonzero-delay switch.

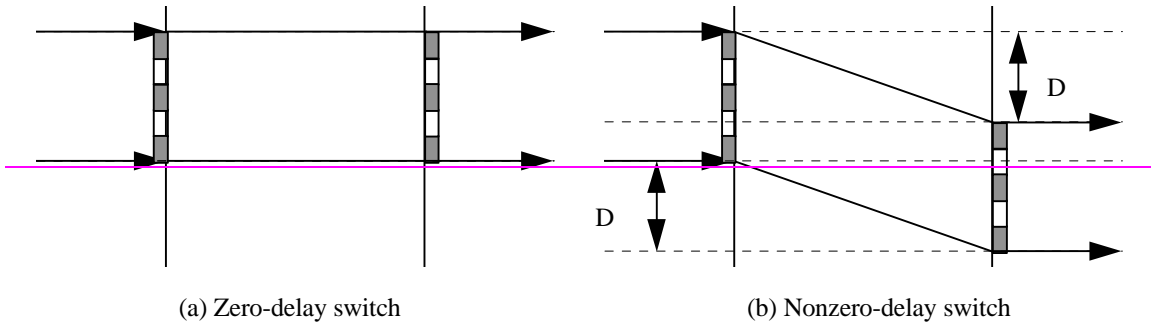


Figure A.1aD: Discontiguous frames, Input rate = Output rate, No change in gaps

This case is similar to cases 1aC and 1bC. The switch latency is determined by a delay of the first bit (or the last bit). Here we have:

- FIFO = D, correct
- LIFO = D, correct
- Input rate = Output rate
- MIMO = min {LIFO, FIFO - FIT} = {D, D} = D, correct

Case 1bD: Discontiguous Frames, Input Rate = Output Rate, Expansion of Gaps

Figure A.1bD shows the flow for a nonzero-delay switch, while a zero-delay switch with expansion of gaps is an impossible scenario.

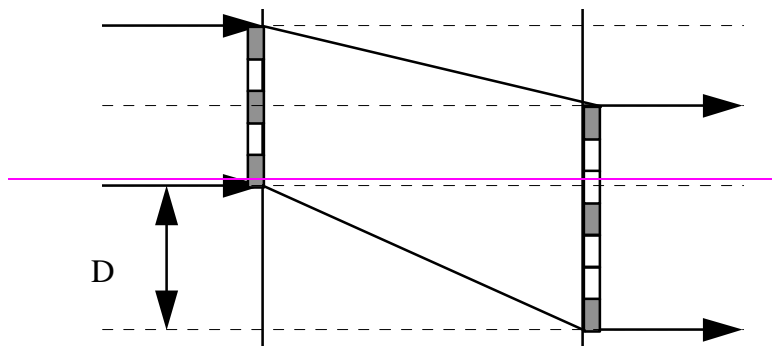


Figure A.1bD: Discontiguous frames, Input rate = Output rate, Expansion of gaps

In this case, the switch latency D is given by:

$$D = \text{first bit delay} + \text{time of additional gaps on output}$$

Here we have:

- FIFO $< D$, incorrect; FIFO is incorrect because it does not reflect expansion of gaps. Note that for a nonzero delay switch, FIFO may be zero (the case of zero delay for the first bit)
- LIFO = D , correct
- Input rate = Output rate
- MIMO = $\min \{ \text{LIFO}, \text{FILO} - \text{FIT} \} = \min \{ D, D \} = D$, correct

Case 1eD: Discontiguous Frames, Input Rate = Output Rate, Compression of Gaps

Figure A.1eD shows the flow for a zero-delay and a nonzero-delay switch with compression of gaps.

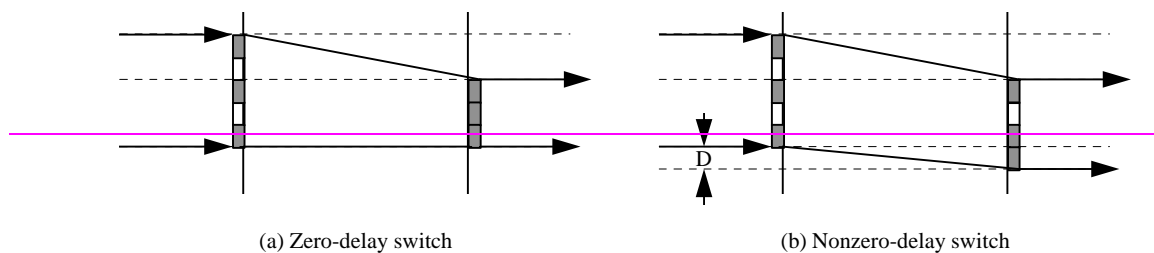


Figure A.1eD: Discontiguous frames, Input rate = Output rate, Compression of gaps

In this case, the switch latency D is given by:

$$D = \text{Last bit delay} = \text{First bit delay} - \text{Time of additional gaps on input}$$

Here we have:

- FIFO $> D$, incorrect; FIFO is incorrect because it does not reflect compression of gaps.
- LIFO = D , correct
- Input rate = Output rate
- MIMO = $\min \{ \text{LIFO}, \text{FILO} - \text{FIT} \} = \min \{ D, D \} = D$, correct

Case 2aD: Discontiguous Frames, Input Rate < Output Rate, No change in Gaps

Figure A.2aD shows the flow for a zero-delay switch and a nonzero-delay switch.

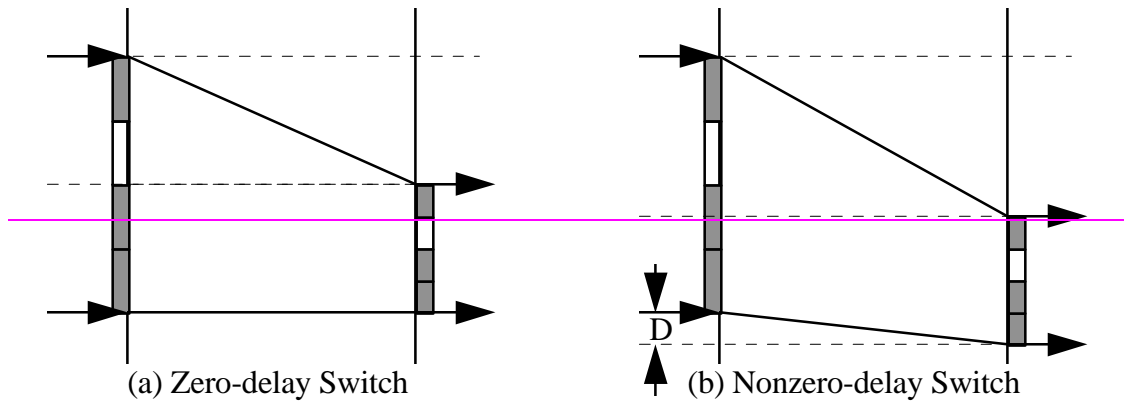


Figure A.2aD: Discontiguous frames, Input rate < Output rate, No change in gaps

This case is similar to cases 2aC and 2bC. The switch latency D is determined by a delay of the last bit. Here we have:

-FIFO > D , incorrect; FIFO may change with changing the output rate and not changing the switch otherwise. So, FIFO does not correctly represent the switch latency.

-LIFO = D , correct

-Input rate < Output rate

-FIFO - FIT * Input rate / Output rate > D

-MIMO = min {LIFO, FIFO - FIT * Input rate / Output rate} = D , correct

Case 2bD: Discontiguous Frames, Input Rate < Output Rate, Expansion of Gaps

Figure A.2bD shows the flow for a zero-delay switch and a nonzero-delay switch.

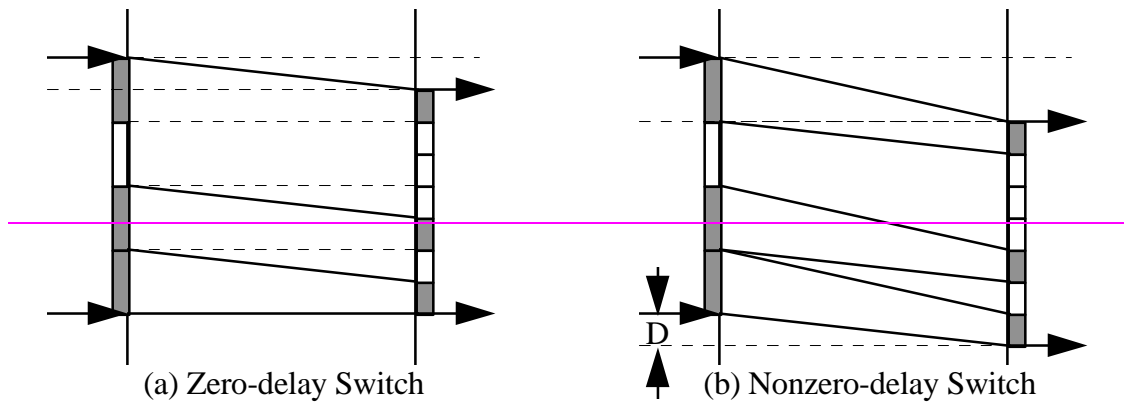


Figure A.2bD: Discontiguous frames, Input rate < Output rate, Expansion of gaps

In this case, the switch latency D is determined by a delay of the last bit. Here we have:

-FIFO is incorrect because:

a. FIFO may be affected by changing the output rate and not changing the switch (latency)

otherwise:

b. FIFO may change by changing the number of gaps on the output while the switch (latency) is unchanged. It should be noted that for this case, with the given input rate and the given number of gaps on input, it is possible to produce cases with the appropriate output rate and the appropriate number of gaps on output such that $FIFO > D$ or $FIFO < D$ or even $FIFO = D$, all without changing the switch (latency):

-LIFO = D, correct

-Input rate < Output rate

-FIFO = FIT × Input rate / Output rate > D

-MIMO = min {LIFO, FIFO = FIT × Input rate / Output rate} = D, correct

Case 2cD: Discontiguous Frames, Input Rate < Output Rate, Compression of Gaps

Figure A.2cD shows the flow for a zero-delay switch and a nonzero-delay switch.

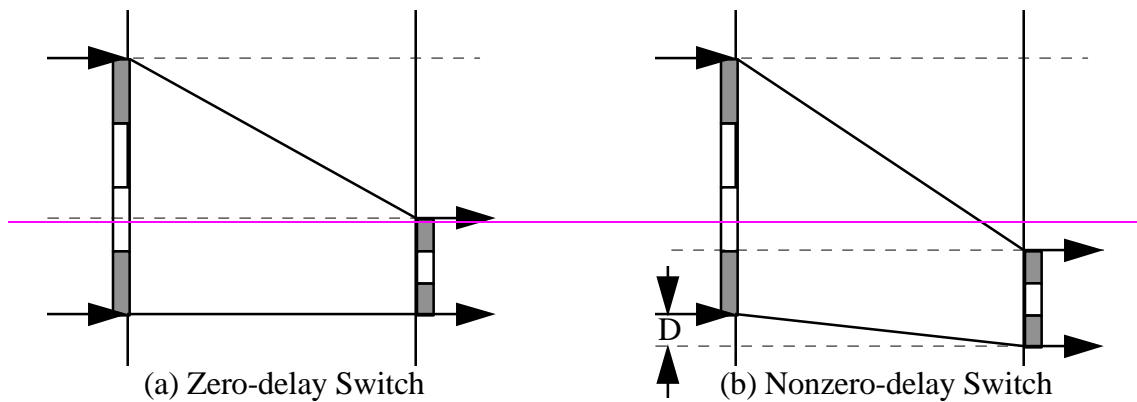


Figure A.2cD: Discontiguous frames, Input rate < Output rate, Compression of gaps

In this case, the switch latency D is determined by the last bit delay. Here we have:

-FIFO > D incorrect; FIFO may be affected by changing the output rate or/and with changing the number of gaps on the output while the switch (latency) is unchanged. So, FIFO does not correctly represent the switch latency.

-LIFO = D, correct

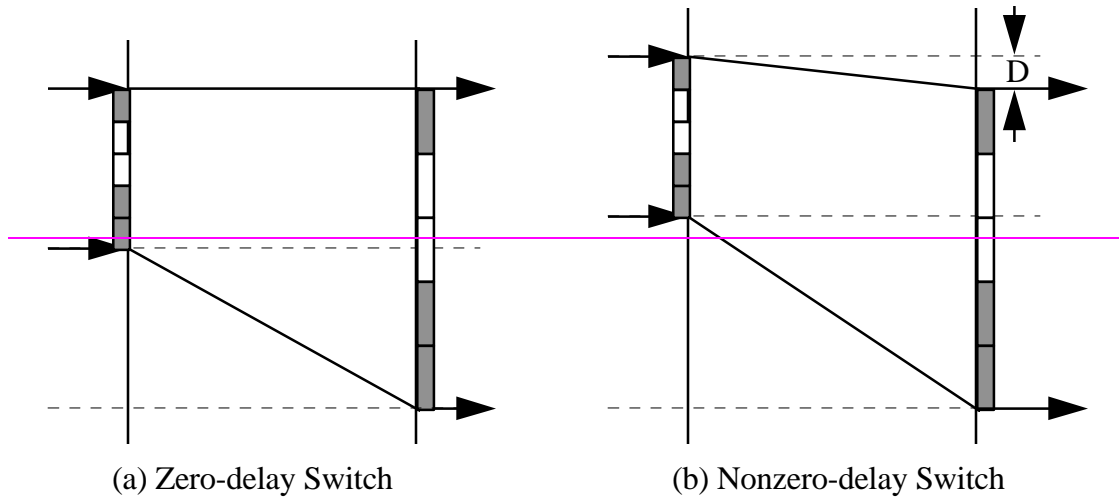
-Input rate < Output rate

-FIFO = FIT × Input rate / Output rate > D

-MIMO = min {LIFO, FIFO = FIT × Input rate / Output rate} = D, correct

Case 3aD: Discontiguous Frames, Input Rate > Output Rate, No Change in Gaps

Figure A.3aD shows the flow for a zero-delay switch and a nonzero-delay switch.



Figure

A.3aD: Discontiguous frames, Input rate > Output rate, No Change in Gaps

This case is similar to cases 3aC and 3bC. The switch latency D is determined by a delay of the first bit. Here we have:

- FIFO = D , correct
- LIFO > D , incorrect; Note that LIFO may change with changing the output rate and not changing the switch otherwise. So, LIFO does not correctly represent the switch latency.
- MIMO = $\min\{\text{LIFO}, \text{FIFO} - \text{FIT} \times \text{Input rate/Output rate}\} = \min\{\text{LIFO}, D\} = D$, correct

Case 3bD: Discontiguous Frames, Input Rate > Output Rate, Expansion of Gaps

Figure A.3bD shows the flow for a nonzero-delay switch, while a zero-delay switch with expansion of gaps is an impossible scenario.

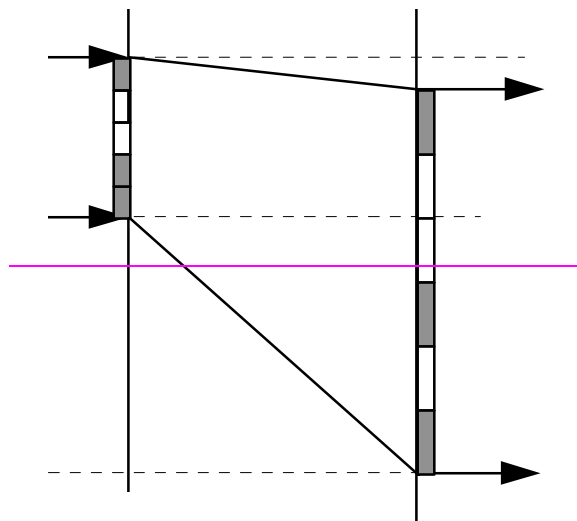


Figure A.3bD: Discontiguous frames, Input rate > Output rate, Expansion of gaps

In this case, the switch latency D is given by:

$$D = \text{first bit delay} + \text{time of additional gaps on output}$$

Here we have:

-FIFO < D , incorrect; FIFO is incorrect because it does not reflect expansion of gaps. Note for a nonzero delay switch, FIFO may be even zero (the case of a zero delay for the first bit)

-LIFO > D , incorrect; Here a similar argument applies as in Case 3aD for LIFO incorrectly being influenced by the output rate, but with the observation that LIFO correctly accounts for a time of additional gaps.

-MIMO = $\min\{\text{LIFO}, \text{FILO} - \text{FIT} \times \text{Input rate}/\text{Output rate}\} = \min\{\text{LIFO}, D\} = D$, correct

Case 3eD: Discontiguous Frames, Input Rate > Output Rate, Compression of Gaps

Figure A.3eD shows the flow for a zero-delay switch, the positive-delay switch and the speed-up switch.

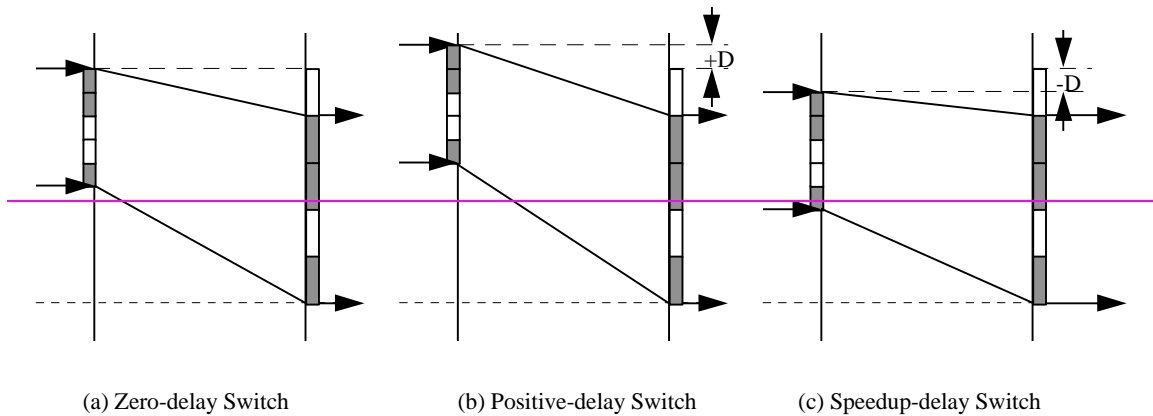


Figure A.3eD: Discontiguous frames, Input rate > Output rate, Compression of gaps

In this case the switch latency D is given by:

$$D = \text{first bit delay} - \text{time of missing gaps on output}$$

Three cases can be distinguished:

a. the case of a zero-delay switch, where:

$$\text{first bit delay} = \text{time of missing gaps on output}$$

b. the case of a positive-delay switch, where:

$$\text{first bit delay} > \text{time of missing gaps on output}$$

c. the case of a speedup-delay switch (a negative delay switch), where:

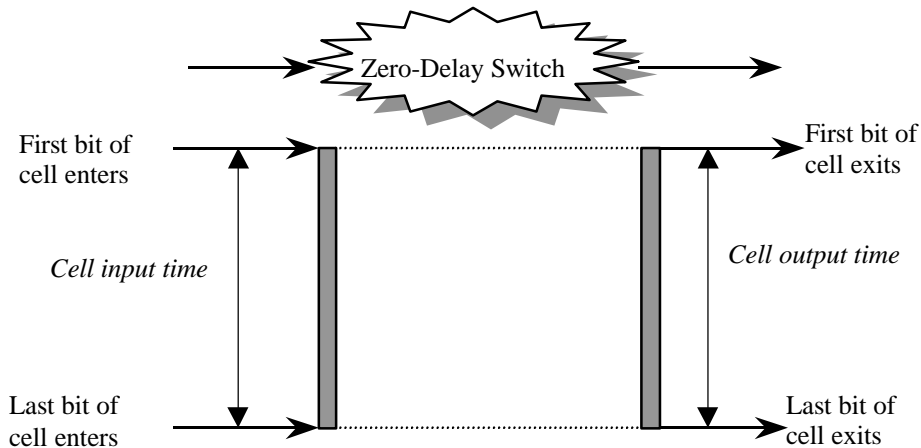
$$\text{first bit delay} < \text{time of missing gaps on output}$$

-FIFO > D, incorrect; FIFO is incorrect because it does not reflect compression of gaps. Note that, here FIFO may be zero (the case of zero delay for the first bit) while the switch latency is negative

-LIFO > D, incorrect; Here a similar argument applies as in Case 3aD for LIFO incorrectly being influenced by the output rate, but with the observation that LIFO correctly accounts for a time of missing gaps.

-MIMO = min {LIFO, FIFO - FIT × Input rate/Output rate} = min {LIFO, D} = D, correct
 In summary, MIMO latency is the only metric that applies to all cases.

	FIFO	LIFO	LIFO	FILO
Input rate = Output rate	0	0	negative	positive
Input rate > Output rate	0	positive	negative	positive
Input rate < Output rate	positive	0	negative	positive



a. Input Link Rate = Output Link Rate



A.5. Latency of Discontinuous Frames Passing Through a Zero-Delay Switch

In this section, we consider how a zero-delay switch handles discontinuous frames in an ATM environment. In particular, we are interested in FILO latency, since it is used in the MIMO latency definition.

Figure A.7 illustrates one of two possible cases of a frame passing through a zero-delay switch with an input link rate higher than the output link rate. The frame includes two cells and the input link rate is 4 times the output link rate. The two cells start arriving at time $t = 0$ and $t = 5$, respectively. A zero-delay switch will start transmitting the first cell at time $t = 0$ and finish at time $t = 4$. The second cell can be transmitted without waiting and it is finished at $t = 9$. This is how long a zero-delay switch will take to transmit this frame. Hence, FILO latency of a zero-delay switch for this frame is 9. This is the normalized frame output time (NFOT) for this input pattern. No device can transmit this frame any faster. If a device takes longer, the difference between the FILO latency of the device and NFOT is considered as the delay introduced by the device.

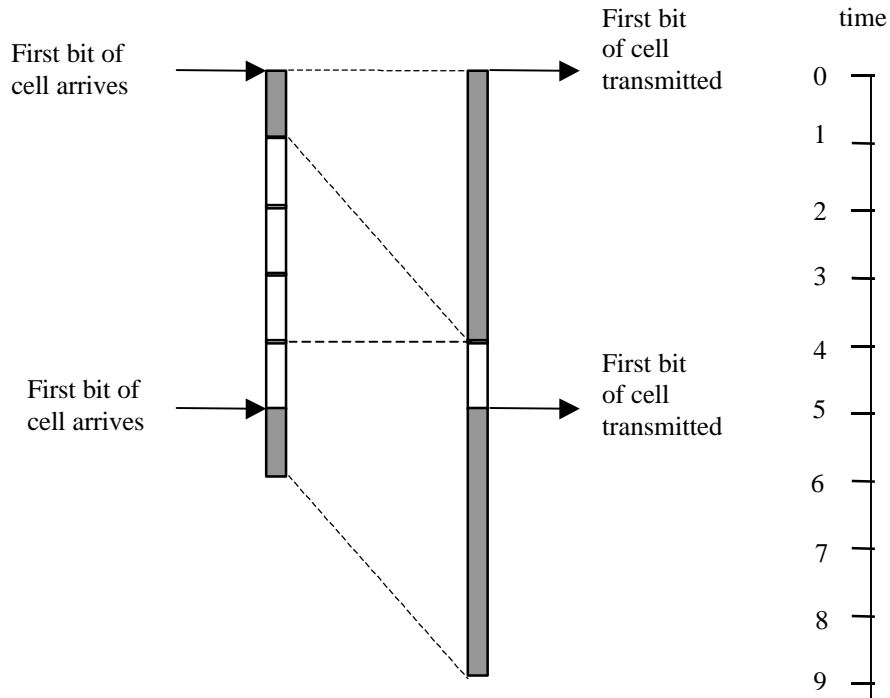


Figure A.7: Zero Delay Switch Operations, no Cell Waiting Case
(Input rate > Output Rate)

Figure A.8 shows the other possible case of a frame passing through a zero-delay switch with an input link rate higher than the output link rate. As in Figure A.7, the frame has two cells and the input link rate is 4 times the output link rate. However, the frame has a different gap pattern. The second cell arrives at time $t = 2$ and thus has to wait. A zero-delay switch will start transmitting the first cell at time $t = 0$ and finish at time $t = 4$. The second cell can be transmitted at $t = 4$ and finished at $t = 8$. Hence, FILO latency of a zero-delay switch for this frame is 8.

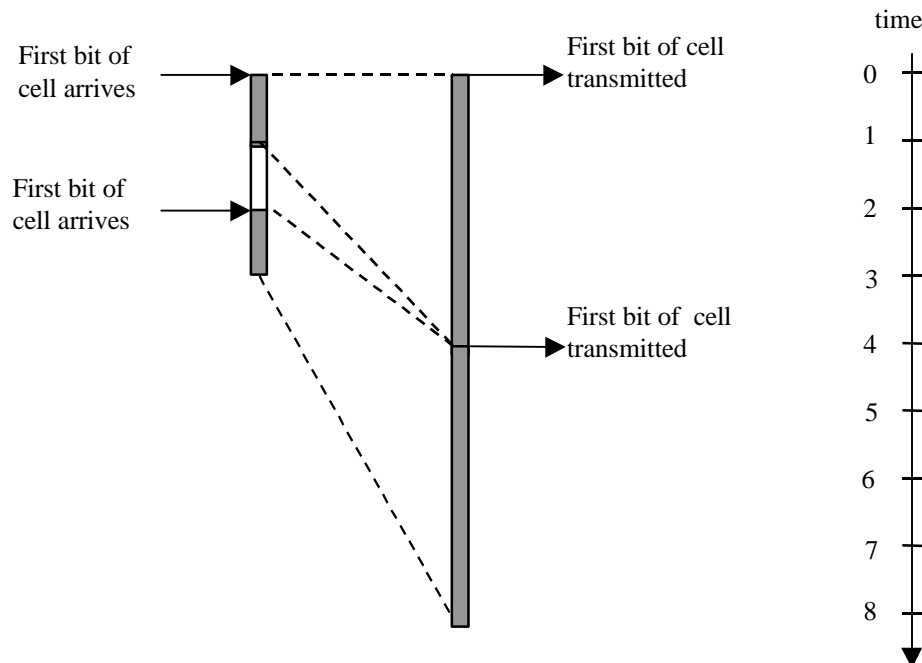


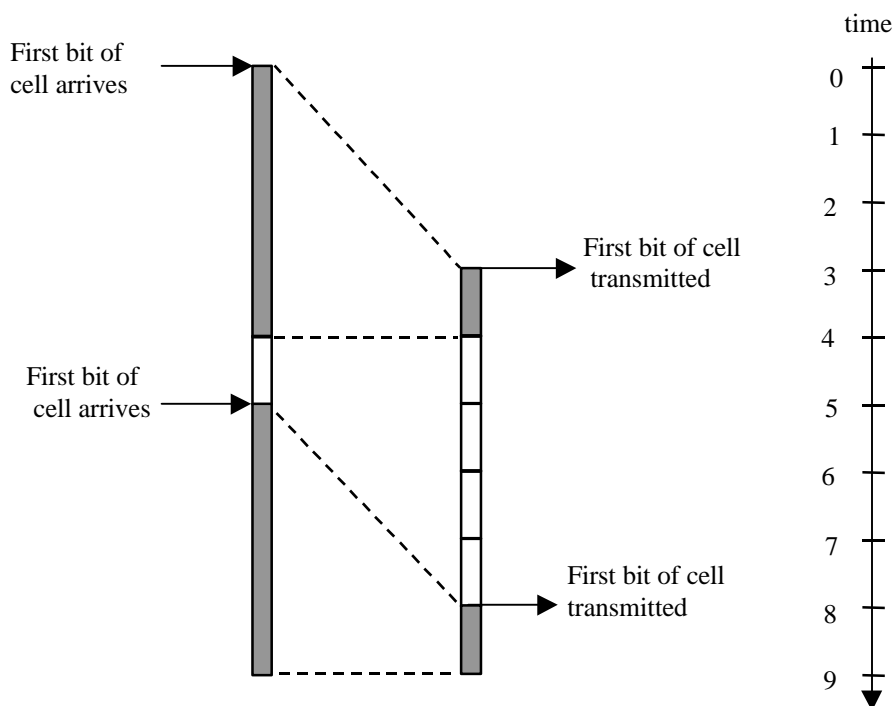
Figure A.8: Zero Delay Switch Operation, Cell Waiting Case
(Input Rate > Output Rate)

Thus, in the case when the input link rate is higher than the output link rate, it is possible that: an incoming cell can be transmitted immediately (no cell waiting case) or an incoming cell has to wait for previously received cells of the same frame to be transmitted (cell waiting case).

Thus, for a given discontinuous frame, it is possible that some cells have to wait on previously received cells of the same frame, while some cells can be transmitted without waiting. Also, notice that a zero-delay switch is decreasing the size of each gap from input, with some gaps being completely removed.

Figure A.9 illustrates the only possible case of a frame passing through a zero-delay switch with an input rate lower than the output rate. Again, the frame includes two cells but the output link rate is now four times the input link rate. The two cells arrive at time $t = 0$ and $t = 5$, respectively. A zero-delay switch will start transmitting the first cell at time $t = 3$ (not at $t = 0$, in order to avoid an underrun), and finish at time $t = 4$. The second cell starts at $t = 8$ and finishes at $t = 9$. This is how long a zero-delay switch will take to transmit this frame. Hence, the FILO latency of a zero-delay switch for this frame is 9.

Note that in the case when the input rate is lower than the output rate, a cell never has to wait for completion of transmissions of previously received cells. Also, notice that in this case, a zero-delay switch does not eliminate any gaps from the input, although each gap is enlarged on output. Additionally, when back-to-back cells are received on the input, new gaps are introduced between cells on the output.



ATM Forum Technical Committee **Figure A.9: Zero-Delay Switch Operation**
(Input Rate < Output Rate)

A.6. Calculation of FILO Latency for a Zero-Delay Switch

The MIMO definition introduces NFOT as the FILO latency of a frame passing through a zero-delay switch. In this section, we explain how to obtain NFOT "on the fly," i.e., when a frame pattern is not known in advance, but cell arrival times can be obtained in real time. We define the following parameters:

- CIT = cell input time = 424[bits] / Input Link Rate [bits/sec]
- COT = cell output time = 424[bits] / Output Link Rate [bits/sec]

The procedure for NFOT calculation is as follows:

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

b. For each cell with its first bit arriving at time t, update NFOT as follows:

$$\text{NFOT} = \max\{t, \text{NFOT}\} + \text{CT}$$

where:

$$\text{CT} = \begin{cases} \text{CIT} & \text{if input link rate} \leq \text{output link rate} \\ \text{COT} & \text{if input link rate} \geq \text{output link rate} \end{cases}$$

A.7. Equivalent MIMO Latency Definition

An equivalent MIMO latency definition, which is more convenient for use in frame latency measurements and calculations when the input link rate is lower than or equal to the output link rate, can be derived as follows.

Input link rate \leq output link rate, implies that CIT \geq COT. A zero-delay switch will transmit the last bit of each cell of the frame as soon as it is received. In particular, the last bit of the frame is transmitted as soon as it is received. Thus, NFOT in these cases is equal to the frame input time:

$$\text{NFOT} = \text{Frame Input Time}$$

and,

$$\begin{aligned} \text{MIMO latency} &= \text{FILO latency} - \text{NFOT} \\ &= \text{FILO latency} - \text{Frame Input Time} \\ &= \text{LILO latency} \end{aligned}$$

Then the equivalent MIMO latency definition is:

$$\text{MIMO latency} = \begin{cases} \text{LILO latency} & \text{if Input Link Rate} \leq \text{Output Link Rate} \\ \text{FILO latency} - \text{NFOT} & \text{otherwise} \end{cases}$$

Throughout this discussion, we assume that the link rates are used in latency computation. If other rates are used, there is the potential for strange results. For example, it is possible that a carrier may offer a lower rate contract to a customer on a higher rate link. If the peak cell rate for the traffic contract is less than the link rate, and this peak cell rate is used for MIMO calculations, then the MIMO value may be negative, depending on the scheduling of cells on the link and the traffic contract. Using the link rate in MIMO calculations avoids this potential problem.

A.8. Measuring MIMO Latency

To measure MIMO latency for a frame passing through the System Under Test (SUT), the times of occurrence for the following two events need to be recorded:

- the first-bit of the frame enters into the SUT,
- the last-bit of the frame exits from the SUT.

The time between these two events is the FILO latency. NFOT can be obtained from the cell pattern of the test frame on input as explained in Section A.6. Substituting FILO latency and NFOT into the MIMO latency formula would give the SUT's delay for a given frame.

If the input link rate is lower than or equal to the output link rate, it is easier to calculate MIMO latency. In this case, the times of occurrence for the following two events need to be recorded:

- the last-bit of the frame enters into the SUT,
- the last-bit of the frame exits from the SUT.

The time between these two events is the LILO latency, which is equal to the MIMO latency for the frame. Note that the cell arrival pattern does not matter in this case.

Contemporary ATM monitors provide measurement data at the cell level. Considering that the definition of MIMO latency uses bit level data, we now describe how to calculate MIMO latency using measurements at the cell level. Standard definitions of two cell level performance metrics, which are of importance for MIMO latency calculation are:

- cell transfer delay (CTD) , defined as the time between the first bit of the cell entering the switch and the last bit of the cell leaving the switch,
- cell inter-arrival time, defined as the time between arrival of the last bit of the first cell and arrival of the last bit of the second cell.

In cases where input link rate is higher than output link rate, according to the MIMO latency definition, FILO latency has to be measured. From Figure A.10, it can be observed that:

$$\text{FILO latency} = \text{First cell's transfer delay} + \text{First cell to last cell inter-arrival time}$$

Thus, to calculate MIMO latency when the input link rate is higher than or equal to the output link rate, it is necessary to measure the transfer delay of the first cell of a frame and the inter-arrival time between the first cell and the last cell of a frame.

In cases when input link rate is lower than or equal to output link rate, it is sufficient to measure LILO latency. From Figure A.11, it can be observed that:

$$\text{LILO latency} = \text{Last cell's transfer delay} - \text{CIT}$$

Thus, to calculate MIMO latency when the input link rate is lower than or equal to the output link rate, it is necessary to measure the transfer delay of the last cell of a frame.

A.9. User Perceived Delay

It should be pointed out that MIMO latency measures only the SUT's contribution to the delay. It does not include the delay caused by components not in the SUT's control. In particular, it does not include the frame input time. However, a user using the system does have to wait while the frame is being sent to the SUT. A user typically assembles the frame and gives it to the network. The user starts waiting as soon as the first bit starts entering the system and cannot do any meaningful work until the last bit exits the network. Thus, user perceived performance is reflected by FILO latency.

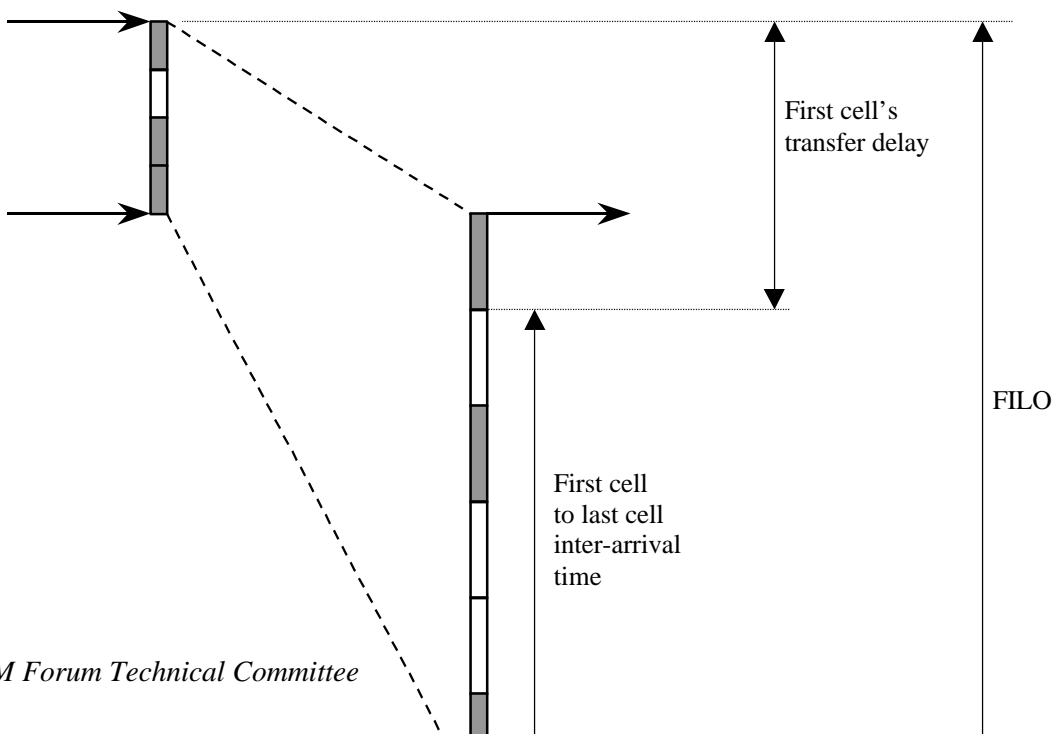


Figure A.10: FILO Latency Calculation
(Input Rate > Output Rate)

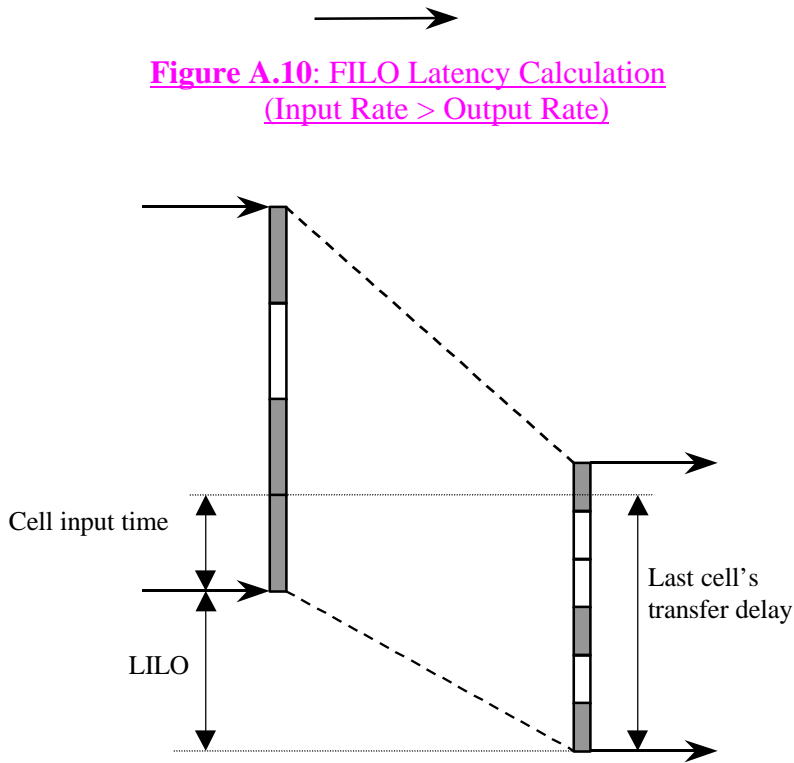
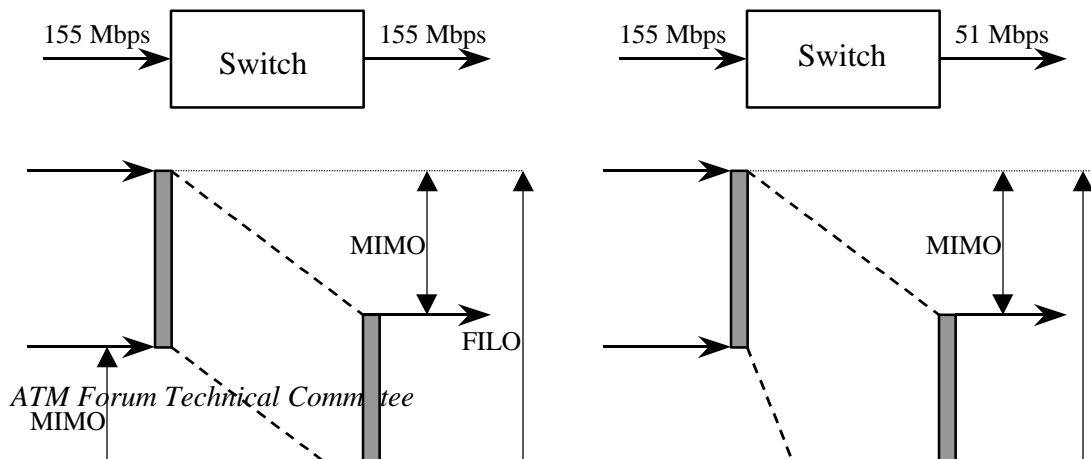


Figure A.11: LILO Latency Calculation (Input rate ≤ Output Rate)

Figure A.12 illustrates the relationships between the user perceived performance and MIMO latency in two scenarios with continuous frames. In the first scenario, the input link rate is same as the output link rate. In the second scenario, the output is slower. The switch delay, as given by MIMO latency, is same in both cases; but the user perceived delay, as given by FILO latency, is different. For the case in Figure A.12b, FILO latency is worse. It can be observed that the user perceived delay depends upon input/output link speeds. On the other hand, network delay measured by MIMO latency is independent of link speeds. The difference between those two delays is the frame latency through a zero-delay switch.



b. Input Link Rate > Output Link Rate

Figure A.12: FILO Latency as User Perceived Delay

References:

- [1] CCITT Recommendation X.135: “Speed of Service (Delay and Throughput) Performance Values for Public Data Networks when Providing International Packet Switched Service”, 1992
- [2] S. Bradner, “Benchmarking Terminology for Network Interconnection Devices”, RFC 1242
- [3] ITU-T Recommendation I.356, “B-ISDN ATM Layer Specification,” ITU-Study Group 13, Geneva, 1995

Editorial note: The current text of Appendix B includes the correction of a printing error which has been found in the algorithm pseudo-code part of this document approved at September '97 ATMF meeting.

Appendix B: Methodology for Implementing Connection Configurations

B.1. Introduction

In Sections 3.1.7 and 3.2.7 of the baseline text, a number of configurations have been presented for throughput and latency measurements. In their basic form, these configurations require traffic generators and analyzers, whose number increases as the number of ports on a switch increases. Since the test monitors are rather expensive, it is desirable to define scalable configurations that can be used with a limited number of generators. However, one problem

with scalable configurations is that there are many ways to set up the connections and the results could vary with the setup. In this appendix, a standard method for generating these configurations is defined. Thus, anyone can design a connection configuration for switches with any number of ports. Since the methodology presented here applies to any number of traffic generators, it can be used for non-scalable (basic) configurations as well.

Performance testing requires two kinds of virtual channel connections (VCCs): foreground VCCs (traffic that is measured) and background VCCs (traffic that simply interferes with the foreground traffic). The methodology for generating configurations of both types of VCCs is covered in this appendix.

The VCCs are formed by setting up connections between ports of the switch. The connection order of these ports is referred to here as a **VCC Chain**. For example, the VCC shown in Figure B.1 consists of one VCC chain passing through ports P1-P2-P3-P4-1. Another possible configuration for this "N-to-N" single generator scalable configuration would be P1-P3-P2-P4-P1. For an N-port switch, there are a total of (N-1)! possible configurations.

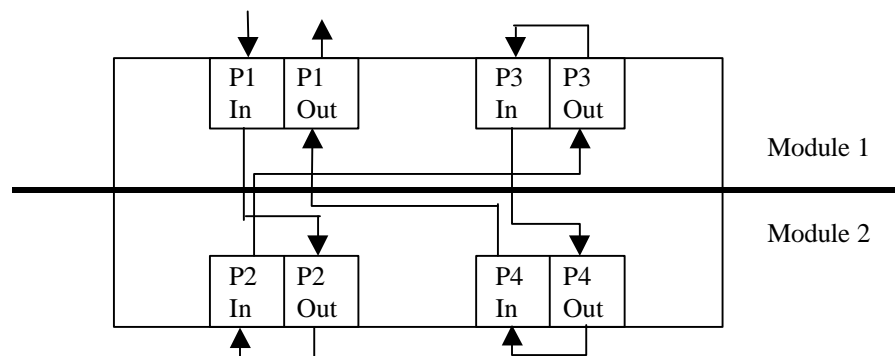


Figure B.1 One out of six possible VCC chains that can implement the 4-to-4 straight configuration with a single generator.

If the four-port switch shown in Figure B.1 consists of two modules with two ports each, the measured performance may depend upon the number of times the VCC chain passes from one module to the other and may be different for different configurations.

At the end of this appendix, the pseudocode for a computer program is presented that allows generating a standardized port order for all connection configurations. This methodology (pseudocode) generally creates VCC chains that cross the modules as often as possible while still keeping the whole process simple.

B.2. Definitions and Rules

In order to generate a standard configuration, it is first necessary to have a standard method of numbering the ports of a switch. This method is presented in this section.

Consider a switch with several modules. Each module may have a varying number of ports. In order to number these ports, the first step is to generate a schematic of modules placed one below the other. The schematic is drawn such that the modules are arranged in a decreasing order of number of ports. Then the switch ports are numbered sequentially, along the columns, starting from the top left corner of the schematic. This port numbering helps in creating VCC chains that cross modules as often as possible. The port numbers obtained this way are represented by P_i in this appendix.

Figure B.2 shows one example of port numbering. The switch consists of three modules with 8, 7, and 6 ports respectively. The first port on the first module is numbered 1 or P_1 . The first port on the second module is numbered P_2 , and so on up to P_{18} as shown in the figure. For simplicity, we also refer to P_i as port i .

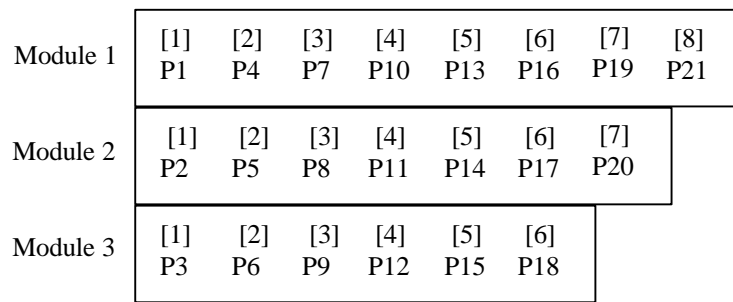


Figure B.2 Numbering of ports in a switch with different number of ports per module.

The second thing we need is a standard method of presenting connection configurations. Each VCC chain is represented by a three-dimensional matrix $C(i, j, k)$. Matrix index i represents the interconnection order among the switch ports, where the value 0 indicates the source port and the last value indicates the destination port. Index k represents the generator number and index j represents the chain number starting at that generator. One row $C(*, j, k)$ of the matrix represents a single VCC chain. For example, if the first VCC chain from generator #2 starts at source port P_1 , passes through ports $P_3, P_4, P_5, P_6, P_7, P_8$, and exits at port P_2 , the matrix C has the following entries: $C(0,1,2)=P_1, C(1, 1, 2)=P_3, C(2, 1, 2)=P_4, C(3, 1, 2)=P_5, C(4, 1, 2)=P_6, C(5, 1, 2)=P_7, C(6, 1, 2)=P_8, C(7, 1, 2)=P_2$. Figure B.3 illustrates this VCC chain. The source port and destination ports are also represented by symbols C_{in} and C_{out} . For the VCC of Figure B.3, $C_{in}=P_1, C_{out}=P_2$.

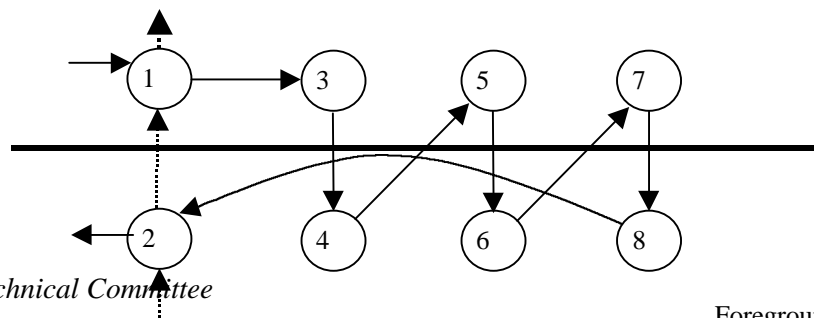


Figure B.3 A VCC chain

$NP(k)$ denotes the total number of **intermediate** ports for a VCC chain generated by generator k . Notice that the source and destination ports are not counted. In the case of Figure B.3, $NP(2) = 6$. Note that $C(NP+1, j, k)$ is always the destination port.

For latency measurements, the foreground traffic involves only two ports, one for input and the other for output. To design the VCC chain for this traffic, the operator may simply chose any two ports, referred to as C_{Fin} and C_{Fout} respectively. Here, F in the subscript signifies "foreground." In order to avoid interference with the foreground traffic, the background VCC chains may or may not use C_{Fin} and C_{Fout} . If the background traffic does use these ports then it should only be in the directions opposite to that used by the foreground traffic. Figure B.4 shows a schematic representation of connection configuration for latency measurement of a 8-port 2-module switch. The foreground traffic uses ports P2 and P1 as the source and destination ports, respectively. So, $C_{Fin}=P2$ and $C_{Fout}=P1$. The background traffic also uses these ports but in the opposite direction. Therefore, for the background traffic: $C_{in} = C(0,1,2)=P1$ and $C_{out}=C(0,1,2)=P2$. The background traffic can use the six remaining ports in both directions. Incidentally, Figure B.3 shown earlier shows the VCC chain representation of this same configuration. From now on, we only show VCC chain representations for all configurations. It is straight forward to generate the schematic representations from it.

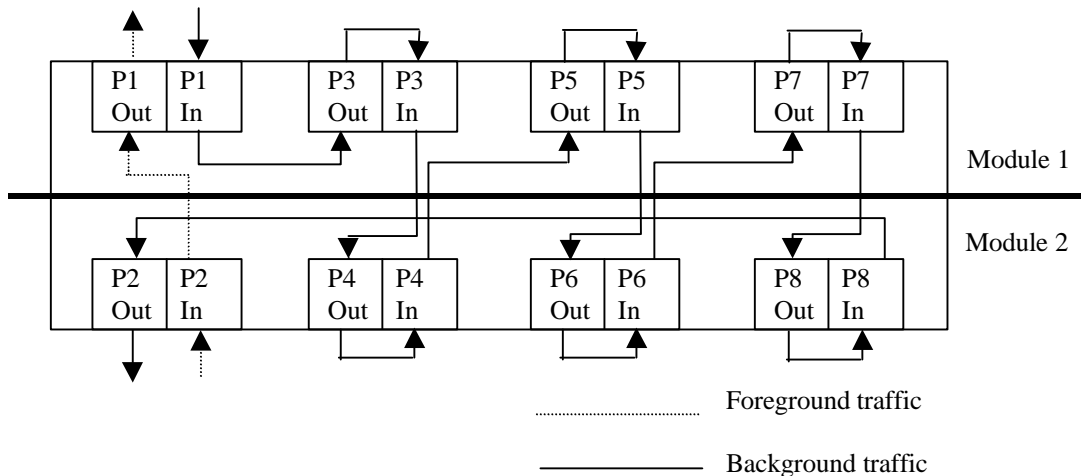


Figure B.4. A 7-to-7 straight configuration with one-generator for the background traffic.

B.3. Connection Configurations Characteristics.

In this section we analyze several of the configurations for throughput and latency measurements and show how scalable version of them can be obtained using the algorithm given in Section B.3. The algorithm consists of three simple rules:

1. The chains generally go from port i to port i+1 unless the port has already been fully used by other chains.
2. After generating jth chain, j+1 chain can be generated simply by adding 1 to each port index of the jth chain.
3. If there are multiple generators, each generator uses a contiguous subset of the switch ports as source ports. Each generator needs as many source ports as the number of VCC chains starting from it.

B.3.1 N-to-N Straight (Single Generator)

This configuration is used for throughput as well as latency measurements. The scalable versions can be obtained as follows:

a) Throughput measurements: For these tests, we need only a single chain starting from a single generator, i.e., k=1 and j=1. The chain starts from one port, goes through all other ports and exits from the starting port. Therefore, NP(1) is equal to N-1. C_{in} and C_{out} coincide and any port P_x could be selected to be the input/output port

Figure B.5 illustrates this case for the 2-module 8-port switch. Figure B.5a shows how to number the switch ports. Figure B.5b presents the VCC chain representation of the configuration, using C_{in}=C(0,1,1) = C_{out} = C(8,1,1) =P1.

The application of the algorithm is simple. The ports C(i,1,1) in the VCC chain are selected in numerically increasing order. The ports are included in VCC chain if they are not already used up. After reaching Nth port, the index (i) starts again from the beginning (from i=1).

Module 1	[1] P1	[2] P3	[3] P5	[4] P7
Module 2	[1] P2	[2] P4	[3] P6	[4] P8

Figure B.5a. Port numbering of a switch with 2 modules and 4 ports on each module. The numbers in brackets indicate the port numbers in the module.

For C_{in}=C_{out}=P1, the VCC chain is:P1-P2-P3-P4-P5-P6-P7-P8-P1

If we chose P3 as the source and destination then the VCC chain will be: P3-P4-P5-P6-P7-P8-P1-P2-P3

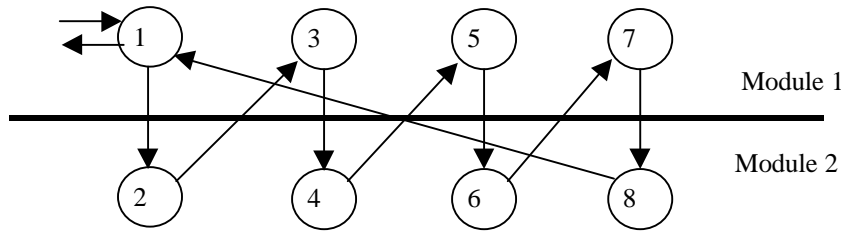


Figure B.5b. The 8-to-8 straight configuration with one generator.

Note that in both cases, the VCC chains cross the modules at every hop.

b) Latency Measurements: First, let us consider the case in which the background traffic does use the source/destination ports of the foreground traffic (but in the opposite direction). The background traffic passes through all other ports. Therefore, $NP(1)$ is equal to $N-2$. C_{in} and C_{out} for the background coincide respectively with P_{Fout} and P_{Fin} .

If $P_{Fin}=P2$ and $P_{Fout}=1$, the foreground chain is P2-P1 and the background chain is P1-P2-P3-P4-P5-P6-P7-P8-P2. This connection configuration was presented earlier in Figures B.3 and B.4.

Now, let us consider the case in which the background traffic does not use the source/destination ports of the foreground. In this case, $NP(1)$ is equal to $N-3$. C_{in} and C_{out} coincide and could be selected any of the switch ports P_x except P_{Fout} and P_{Fin} . For example, the foreground could use the chain P2-P1 and background could use P3-P4-P5-P6-P7-P8-P3. Figure B.6 illustrates this case.

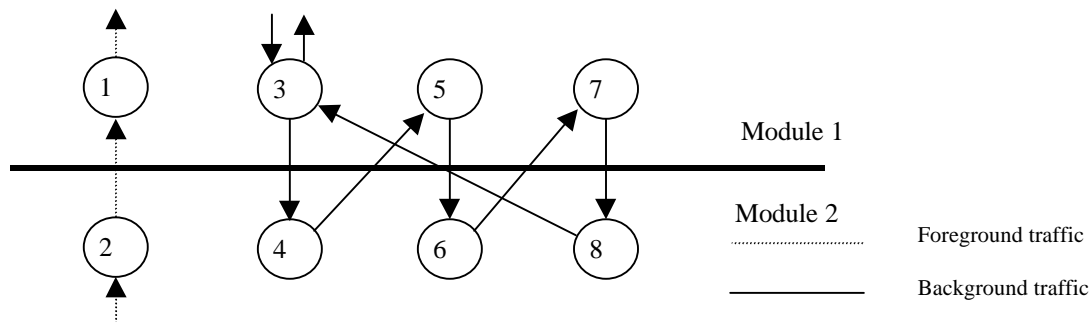


Figure B.6 Implementation of the 6-to-6 straight configuration with one generator.

B.3.2. N-to-N Straight (r Generators)

This configuration implements the N-to-N straight configuration with r generators.

a) Throughput Measurements: Each generator has one VCC chain. In all there are r VCC chains. Of the N ports, r ports are used as source/destination of these chains. The remaining ports are divided among the generators as evenly as possible.

Let $p = \text{mod}(N-r, r)$

- For the first p VCC chains, the number of intermediate ports NP is equal to the quotient of $(N-r)/r$ plus 1, i.e., $\lfloor (N-r)/r \rfloor + 1$
- For the remaining (r-p) VCC chains, NP is equal to the quotient of $(N-r)/r$, or $\lfloor (N-r)/r \rfloor$
- For all VCC chains, the source/destination ports coincide and may be selected from any of the switch ports P_x not selected by other VCC chains as source or destination.

As an example, consider the 8-port switch again. With $r=3$ generators, p equals $\text{mod}(8-3, 3) = 2$. So, the first two VCC chains have $NP = \lfloor (8-3)/3 \rfloor + 1 = 2$ intermediate ports, and the last chain has $NP = \lfloor (8-3)/3 \rfloor = 1$.

Figures B.7 illustrates the implementation of the VCC chains for this case. First we select the source and destination ports:

Port 1 is the source and destination for the first chain, so $C(0,1,1)=P1, C(3, 1,1)=P1$

Port 2 is the source and destination for the second chain, so $C(0,1,2)=P2, C(3,1,2)=P2$

Port 3 is the source and destination for the third chain, so $C(0,1,3)=P3, C(2, 1,3)=P3$.

These selections have been made to avoid any overlap.

Then we apply the algorithm. Let us start with the VCC chain having port 1 as the source. The next port available is P4, so $C(1,1,1)=P4$, then $C(2,1,1)=P5$. This VCC chain has two intermediate ports, for this reason it is now complete. Now we continue with the VCC chain starting at port 2. The next available port is port 6 (because 4 and 5 are fully occupied by the previous VCC chain). So $C(1,1,2)=P6$, and then $C(2,1,2)=P7$. Similarly, $C(1,1,3)=P8$. This VCC chain has only one intermediate port. The VCC chain implementation is complete.

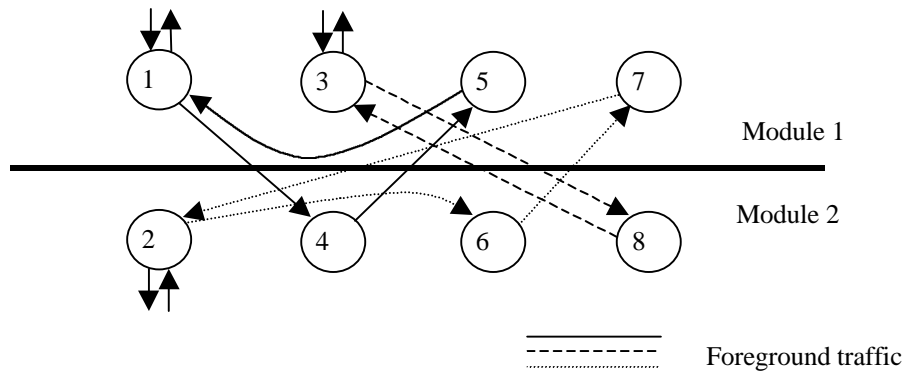


Figure B.7 Implementation of the 8-to-8 straight configuration with 3 generators.

b) Latency Measurements: Consider the case with the background traffic using the foreground ports in the opposite direction. The remaining N-1 ports are evenly divided among the r background VCC chains.

Let $p = \text{mod}(N-r-1, r)$

- For the first p VCC chains NP is equal to the quotient of $(N-r-1)/r$ plus 1, i.e., $\lfloor (N-r-1)/r \rfloor + 1$
- For the remaining $(r-p)$ VCC chains, NP is equal to the quotient of $(N-r-1)/r$, or $\lfloor (N-r-1)/r \rfloor$
- For one of VCC chains, C_{in} and C_{out} coincide with P_{Fout} and P_{Fin} respectively.
- For the other VCC chains, C_{in} and C_{out} coincide and could be selected from any of the switch ports P_x not selected by other VCCs

Figure B.8 illustrates an example for this case. Ports 1 and 2 are used by the foreground traffic as destination and source ports, respectively.

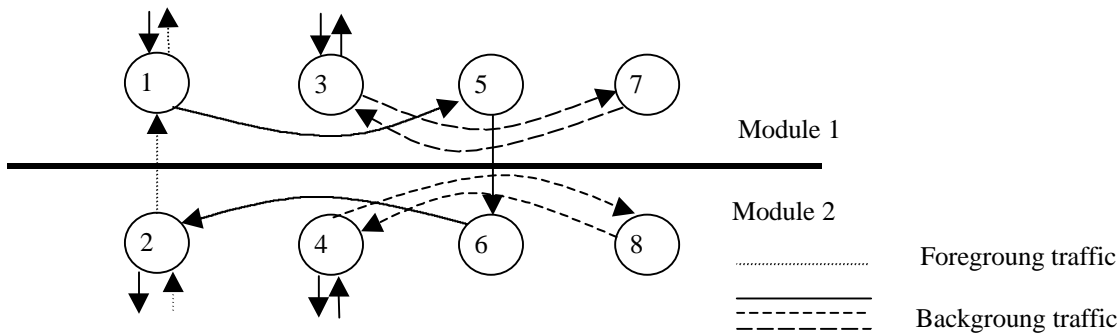


Figure B.8 Implementation of the 7-to-7 straight configuration with 3 generators for background traffic in latency measurement.

Ports 1 and 2 will be used as source and destination ports (respectively) by one of the background VCC chains. The other two generators will use port 3 and 4 as the source and destination ports, respectively. For the first VCC chain, $NP = 2$ and for the other two VCC chains $NP = 1$. The chains are: P1-P5-P6-P2, P3-P7-P3, and P4-P8-P4. Note that the first chain goes from P1 to P5 since P2, P3, P4 since have already been assigned to other chains.

The configuration for the case when the background traffic does not share the ports with the foreground can be generated by the above procedure by considering the switch having only N-2 ports

B.3.3. N-to-m Partial Cross (r Generators)

This is a generalization of N-to-m Partial cross with 1 generator presented in the baseline. The discussion here applies for $r=1$. Also, by appropriately setting r , one can obtain non-scalable (basic) configurations.

a) Throughput Measurements: This configuration has $m \cdot r$ VCC chains originating from r , where each generator originates m VCC chains. Each has a load of $1/m^{\text{th}}$ of the generator. Each intermediate node has exactly m of these streams flowing through it. Again, the ports are evenly

divided among the chains. However, since each chain uses only a part of the port's capacity, the ports can be used by other chains even from other generators as well.

Let $p = \text{mod}(N-r, r)$

- For the first p VCC chains, the number of intermediate ports NP is equal to the quotient of $(N-r)/r$ plus 1, i.e., $\lfloor (N-r)/r \rfloor + 1$
- For the remaining $(r-p)$ VCC chains, NP is equal to the quotient of $(N-r)/r$, or $\lfloor (N-r)/r \rfloor$
- For all m VCC chains, source/destination ports coincide and may be selected from any of the switch ports P_x not selected by other VCC chains.

Figure B.9 illustrates the case of 8-to-2 partial cross with 2 generators.

In this case, $p = \text{mod}(8-2, 2) = 0$. So, the VCC chains of both generators have $\lfloor (8-2)/2 \rfloor = 3$ intermediate ports.

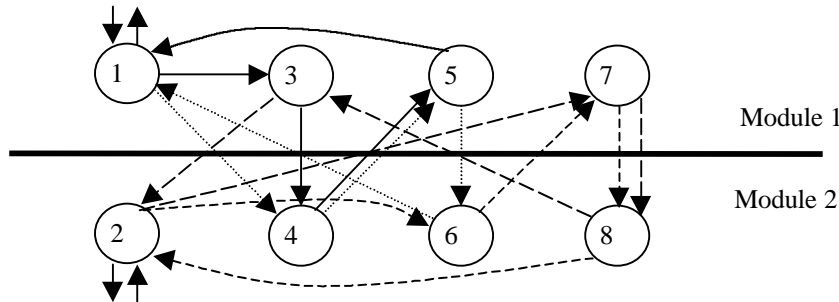


Figure B.9 Implementation of 8-to-2 partial cross configuration with 2 generators for foreground traffic

Both of the VCC chains of the first generator start and end at port 1, so:
 $C(0,1,1)=C(0,2,1)=C(4,1,1)=C(4,2,1)=P1$.

Similarly for the two VCC chains of the other generator:
 $C(0,1,2)=C(0,2,2)=C(4,1,2)=C(4,2,2)=P2$.

First we divide the remaining ports among the two generators. The first generator gets P3, P4, and P5. The second generator gets P6, P7, and P8.

The first chain of the first generator is simply P1-P3-P4-P5-P1.

The first chain of the second generator is P2-P6-P7-P8-P2. The second chain from the first generator is obtained by shifting the intermediate ports of the first chain. Therefore, the chain is P1-P4-P5-P6-P1. Note that this chain is sharing port P6 of the other generator since each chain uses only half the capacity. The second chain of the second generator is again obtained by shifting: P2-P7-P8-P3-P2. Note that, shifting P8 would have produced P1 but P1 is being fully used. The next port P2 is also being fully used. So P3 is used.

b) Latency measurements: Again we consider only the case of background traffic sharing the foreground ports in the opposite direction. Excluding the foreground port, the remaining $N-1$ ports are evenly divided among the r generators.

Let $p = \text{mod}(N-r-1, r)$

- For all VCCs of the first **p** generators NP is equal to the quotient of (N-r)/r plus 1, i.e., $\lfloor (N-r)/r \rfloor + 1$
- For all VCCs of the remaining (**r-p**) generators, NP is equal to the quotient of (N-r)/r, or $\lfloor (N-r)/r \rfloor$
- For all **m** VCCs of only one generator, source and destination coincide with P_{Fout} and P_{Fin} respectively.
- For all **m** VCCs of all other generators, source and destination coincide and could be selected from any of the switch ports P_x not selected by other generators.

An example of this case is shown in Figure B.10. In this case, N=8, r=2. So $p = \text{mod}(8-2-1, 2) = 1$, so NP(1)=3 and NP(2)=2.

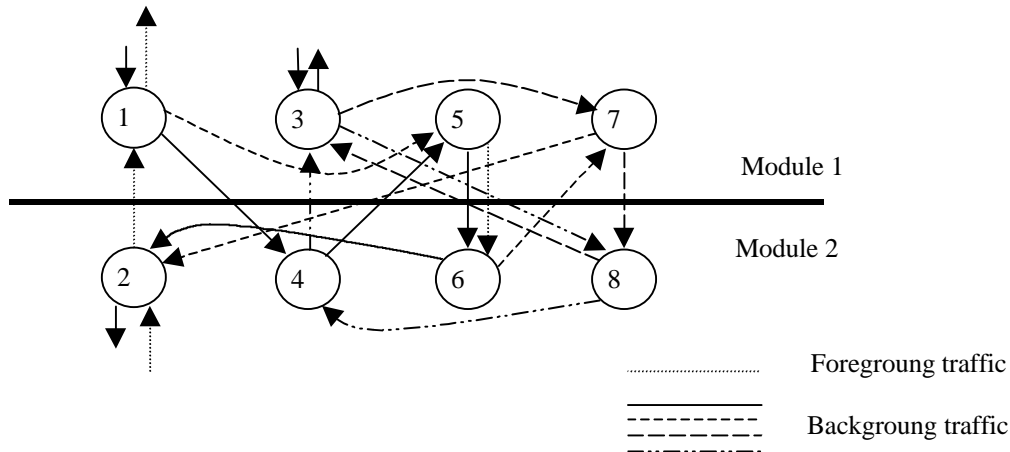


Figure B.10 Implementation of 7-to-2 partial cross configuration with 2 generators for background traffic in latency measurements.

The VCC chains of the first generator will use ports 1 and 2 in opposite directions of the foreground traffic. The VCC chains of the second generator will use port 3 as the source and destination. The chains of the first generator are: P1-P4-P5-P6-P2, P1-P5-P6-P7-P2. The chains of the second generator are: P3-P7-P8-P3, P3-P8-P4-P3.

Table B.1 summarizes the values for number of intermediate ports in various configurations of this section B.2. These values are used in the pseudocode of Section B.3.

	<u>N-to-N straight (Single generator)</u>	<u>N-to-N straight (r Generators)</u>	<u>N-to-m Partial Cross (Single Generator)</u>	<u>N-to-m Partial Cross (r Generators)</u>
<u>Number of Intermediate ports NP</u>	<u>See B.2.1.</u> a) <u>N-1</u> b) <u>N-2</u> c) <u>N-3</u>	<u>See B.2.2.</u> <u>$\lfloor (N-r)/r \rfloor + 1$</u> or <u>$\lfloor (N-r)/r \rfloor$</u>	<u>See B.2.3.</u> a) <u>N-1</u> b) <u>N-2</u>	<u>See B.2.3.</u> <u>$\lfloor (N-r)/r \rfloor + 1$</u> or <u>$\lfloor (N-r)/r \rfloor$</u>

Table B.1 Parameter values used in the algorithm to creating VCC chains for different configurations.

B 4. Algorithm for creating VCC Chains.

The algorithm for creating VCC chains for different connection configuration is based on the definitions given in section **B.1.** and the characteristics specified in section **B.2** and summarized in Table B.1.

- NP(k) denotes the number of intermediate ports for the VCC chains of the kth generator. These values are specified in **B.2.**
- P(f) denotes the fth port of the switch
- C(i, j, k) denotes the ith intermediate port of the jth VCC chain of the kth generator
- The function mod*(x, N) is equal to mod(x, N) except for the cases where mod(x, N) is equal to zero, where the function is equal to N

f = 1;

for (k = 1 to r, step 1)

{ if(k>1)ff=0;

for(q = mod*(1 + $\sum_{d=1}^{k-1} NP(d), N$), to, q => 1, step -1)

{ f=f+1;

while P(f) is source or destination{ f = f+1;}

} end for q

}end if(k>1)

for (j = 1 to m, step 1)

{

if (r is equal to 1 and j > 1) {ff = mod*(f+1, N);}

if (r>1 and j>1) {ff=C(2,j-1,k);}

for (i = 1 to NP(k), step 1)

{

while (P(f) is source or destination or is full)

ff = mod*(f+1, N);}

C(i, j, k) = P(f);f = mod*(f+1, N);

} end for i

} end for j

} end for k.