UNH-IOL
SAS CONSORTIUM

System Interoperability Test Suite
Version 1.01

Technical Document

Last Updated: August 15, 2005

UNH IOL SAS Consortium
InterOperability Laboratory
University of New Hampshire

121 Technology Drive, Suite 2
Durham, NH 03824
Phone: (603) 862-3582
Fax: (603) 862-4181
http://www.iol.unh.edu/consortiums/sas

TABLE OF CONTENTS

The University of New Hampshire
InterOperability Laboratory

MODIFICATION RECORD

February 22, 2005 (Version 0.1) DRAFT RELEASE FOR INTERNAL REVIEW ONLY
David Woolf: Initial Version

February 24, 2005 (Version 0.2) DRAFT RELEASE FOR REVIEW BY SAS PLUGFEST TECHNICAL COMMITEE
David Woolf: Revised Version

March 24, 2005 (Version 0.3) DRAFT RELEASE FOR REVIEW
David Woolf: added PER Estimation test originally written by Matt Plante. Added Appendix A and B.

May 3, 2005 (Version 1.0) FINAL RELEASE
David Woolf: Added further details to test procedures.

August 15, 2005 (Version 1.01) FINAL RELEASE
Michael Davidson: Removed Research Computing Center References

ACKNOWLEDGMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals in the development of this test suite:

Andy Baldman	UNH InterOperability Laboratory
Matthew Plante	UNH InterOperability Laboratory
David Woolf	UNH InterOperability Laboratory

INTRODUCTION

The University of New Hampshire's InterOperability Laboratory (IOL) is an institution designed to improve the interoperability of standards based products by providing an environment where a product can be tested against other implementations of a standard. This particular suite of tests has been developed to help implementers evaluate the Phy Layer functionality of their Serial Attached SCSI (SAS) products.

These tests are designed to determine if a SAS product conforms to specifications defined in Clause 6 of **ISO/IEC 14776-151,** ***Serial Attached SCSI-1.1 (SAS-1.1) standard*** **T10/1601-D, Revision 7** (hereafter referred to as the "SAS Standard"). Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other SAS products. However, when combined with satisfactory operation in the IOL's interoperability test bed, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many SAS environments.

The tests contained in this document are organized in order to simplify the identification of information related to a test, and to facilitate in the actual testing process. Tests are separated into groups, primarily in order to reduce setup time in the lab environment, however the different groups typically also tend to focus on specific aspects of device functionality. A three-number, dot-notated naming system is used to catalog the tests, where the first number always indicates the specific clause of the reference standard on which the test suite is based. The second and third numbers indicate the test's group number and test number within that group, respectively. This format allows for the addition of future tests in the appropriate groups without requiring the renumbering of the subsequent tests.

The test definitions themselves are intended to provide a high-level description of the motivation, resources, procedures, and methodologies specific to each test. Formally, each test description contains the following sections:

Purpose

The purpose is a brief statement outlining what the test attempts to achieve. The test is written at the functional level.

References

This section specifies all reference material *external* to the test suite, including the specific subclauses references for the test in question, and any other references that might be helpful in understanding the test methodology and/or test results. External sources are always referenced by a bracketed number (e.g., [1]) when mentioned in the test description. Any other references in the test description that are not indicated in this manner refer to elements within the test suite document itself (e.g., "Appendix 5.A", or "Table 5.1.1-1")

Resource Requirements

The requirements section specifies the test hardware and/or software needed to perform the test. This is generally expressed in terms of minimum requirements, however in some cases specific equipment manufacturer/model information may be provided.

Last Modification

This specifies the date of the last modification to this test.

Discussion

The discussion covers the assumptions made in the design or implementation of the test, as well as known limitations. Other items specific to the test are covered here as well.

Test Setup

The setup section describes the initial configuration of the test environment. Small changes in the configuration should not be included here, and are generally covered in the test procedure section (next).

Procedure

The procedure section of the test description contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with observable results.

Observable Results

This section lists the specific observables that can be examined by the tester in order to verify that the DUT is operating properly. When multiple values for an observable are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is generally based on the successful (or unsuccessful) detection of a specific observable.

Possible Problems

This section contains a description of known issues with the test procedure, which may affect test results in certain situations. It may also refer the reader to test suite appendices and/or other external sources that may provide more detail regarding these issues.

GROUP 1: POINT TO POINT LINK VERIFICATION

Overview:

This group of tests verifies the interoperability of a small SAS system including a SAS HBA, a SAS Target, and zero or one expander devices connecting the SAS HBA and the SAS Target.

Scope:

Comments and questions regarding the implementation of these tests are welcome, and may be forwarded to David Woolf, UNH InterOperability Lab (djwoolf@iol.unh.edu).

Test #1.1: Converge to Highest Supported Speed

Purpose: To determine if a device pair can establish a link and can exchange data.

References: T10 Std. SAS

Resource Requirements:

- Link monitoring facilities that are able to monitor primitives on the link.
- Local management indicators on the devices within SAS System, that indicate the state of the link as perceived by the different devices.

Last Modification: May 3, 2005

Discussion:

The SAS Standard defines a speed negotiation algorithm that is executed after completion of OOB and before the Identify Sequence. It is expected that if both devices in a device pair support a particular speed, and a reference test channel is used to connect the devices, the devices should converge to the highest mutually supported speed during execution of the speed negotiation algorithm.

This test is an interoperability test. Failure of this test does not mean that a DUT is necessarily non-conformant. It does suggest that a problem in the ability of the devices to work properly together exists and further work should be done to isolate the cause of the failure.

Test Setup: The device pair is connected through a reference test channel.

Procedure:

1. Power off both devices.
2. Power on the DUT1 and ensure that the device is initialized and all needed drivers are loaded.
3. Power on the DUT2 and ensure that the device is initialized and all needed drivers are loaded.
4. Check local management information to verify that the link is established and that the device pair converged to the highest supported speed.
5. Power off both devices.
6. Power on the DUT2 and ensure that the device is initialized and all needed drivers are loaded.
7. Power on the DUT1 and ensure that the device is initialized and all needed drivers are loaded.
8. Check local management information to verify that the link is established and that the device pair converged to the highest supported speed.
9. Power off both devices and disconnect both devices.
10. Power on the DUT1 and ensure that the device is initialized and all needed drivers are loaded.
11. Power on the DUT2 and ensure that the device is initialized and all needed drivers are loaded.
12. Connect the devices using the reference test channel.
13. Check local management information to verify that the link is established and that the device pair converged to the highest supported speed.
14. This procedure should be repeated for all physical links in the system.

Observable Results:

- Check local management information to verify that the link is established and that the device pair converged to the highest supported speed.

Possible Problems: Local management information may not be available, depending on the device. Some steps of the procedure may not be applicable to certain device pairs. For example a disk drive connected to an expander via a SAS midplane, cannot be powered apart from the midplane.

Test #1.2: Disconnect Reconnect

Purpose: To determine if a device pair can reestablish a link and can exchange data after the physical connection has been disconnected and reconnected

References: T10 Std. SAS

Resource Requirements:

- Link monitoring facilities that are able to monitor primitives on the link.
- Local management indicators on the devices within SAS System, that indicate the state of the link as perceived by the different devices.

Last Modification: May 3, 2005

Discussion:

In the field, physical connection may be broken unexpectedly. This test verifies that if a connection is broken unexpectedly, it does not require the devices in the system to be power cycled in order to reestablish a link.

This test is an interoperability test. Failure of this test does not mean that a DUT is necessarily non-conformant. It does suggest that a problem in the ability of the devices to work properly together exists and further work should be done to isolate the cause of the failure.

Test Setup:

The device pair is connected through a reference test channel. Both devices are powered on. The link has been verified using local management information. If the SAS System includes a SAS HBA and SAS Target, the SAS Target has been discovered by the SAS HBA Host System and the Host System has created a partition on the SAS Target.

Procedure:

1. Disconnect and reconnect DUT1 and DUT2. The devices should remain disconnected for 10 seconds.
2. This procedure should be repeated for all physical links in the system.

Observable Results:

- Check local management information to verify that the link is reestablished and that the device pair converged to the highest supported speed. Allow at least 2 minutes for local management information to update after DUT1 and DUT2 are reconnected. If the system under test includes a SAS HBA in a host system and a SAS Target, use the disk management application included in the OS to verify that the disk is visible from the Host System OS. If available, a disk scan utility can be used to cause the SAS HBA to scan for attached devices again after being reconnected. If the disk scan utility is used for this, it should be used no more than 3 times. It is expected that the link be reestablished within 2 minutes.

Possible Problems: Local management information may not be available, depending on the device.

Test #1.3: Power Cycle

Purpose: To determine if a device pair can reestablish a link and can exchange data after one device has been power cycled.

References: T10 Std. SAS

Resource Requirements:

- Link monitoring facilities that are able to monitor primitives on the link.
- Local management indicators on the devices within SAS System, that indicate the state of the link as perceived by the different devices.

Last Modification: May 3, 2005

Discussion:

In the field, unexpected power loss may occur. This test verifies that if power is lost unexpectedly, the devices in the system are able reestablish a link.

This test is an interoperability test. Failure of this test does not mean that a DUT is necessarily non-conformant. It does suggest that a problem in the ability of the devices to work properly together exists and further work should be done to isolate the cause of the failure.

Test Setup:

The device pair is connected through a reference test channel. Both devices are powered on. The link has been verified using local management information. If the SAS System includes a SAS HBA and SAS Target, the SAS Target has been discovered by the SAS HBA Host System and the Host System has created a partition on the SAS Target.

Procedure:

1. Power cycle DUT1. Wait for DUT1 to reinitialize. Ensure that the device is initialized and all needed drivers are loaded.
2. Power cycle DUT2. Wait for DUT to reinitialize. Ensure that the device is initialized and all needed drivers are loaded.

Observable Results:

- For each step, check local management information to verify that the link is reestablished and that the device pair converged to the highest supported speed. Allow at least 2 minutes for local management information to update after the device has completely rebooted from the power cycle. If the system under test includes a SAS HBA in a host system and a SAS Target, use the disk management application included in the OS to verify that the disk is visible from the Host System OS. If available, a disk scan utility can be used to cause the SAS HBA to scan for attached devices again after the attached devices have completely rebooted from the power cycled. If the disk scan utility is used for this, it should be used no more than 3 times, and only after the power cycle in complete. It is expected that the link be reestablished within 2 minutes after the power cycled device has completed its boot up process.
- Repeat this procedure for all physical links in the system.

Possible Problems: Local management information may not be available, depending on the device. Some steps of the procedure may not be applicable to certain device pairs. For example a disk drive connected to an expander via a SAS midplane, cannot be powered apart from the midplane.

Test #1.4: Disconnect Reconnect – Maintain Traffic Flow

Purpose: To determine if a device pair can reestablish a link and resume data flow after one device has been disconnected.

References: T10 Std. SAS

Resource Requirements:

- Link monitoring facilities that are able to monitor primitives on the link.
- Traffic Generation facilities.
- Local management indicators on the devices within SAS System, that indicate the state of the link as perceived by the different devices.

Last Modification: May 3, 2005

Discussion:

In the field, physical connection may be broken unexpectedly. This test verifies that if a connection is broken unexpectedly, SCSI traffic will continue to flow on the link once the connection is repaired.

This test is an interoperability test. Failure of this test does not mean that a DUT is necessarily non-conformant. It does suggest that a problem in the ability of the devices to work properly together exists and further work should be done to isolate the cause of the failure.

Test Setup:

The device pair is connected through a reference test channel. Both devices are powered on. The link has been verified using local management information. The SAS System under test includes a SAS HBA and a SAS Target. The SAS Target has been discovered by the SAS HBA Host System and the Host System has created a partition on the SAS Target.

Procedure:

1. Use a large file transfer to cause SCSI Traffic on the link. Copy a file from the SAS HBA Host System to the attached SAS Target.
2. Disconnect the device pair. The devices should remain disconnected for 10 seconds.
3. Repeat this procedure for all physical links in the system.

Observable Results:

- For each step, check local management information to verify that the link is reestablished and that the device pair converged to the highest supported speed. Allow at least 2 minutes for local management information to update after the device has been reconnected.
- Verify that SCSI traffic resumed automatically, without user intervention, once the link was reestablished. It is expected that traffic resume within 2 minutes.
- Verify that the file transfer completed and that the file is not corrupted by comparing the original file to the one that was transferred through the SAS System.

Possible Problems: Local management information may not be available, depending on the device.

Test #1.5: Packet Error Rate Estimation

Purpose: To determine if the DUT can exchange packets with a link partner with a bit error rate less than or equal to 10^{-12} across a SAS Test Channel.

References: T10 Std. SAS

Resource Requirements:

- Higher layer application capable of generating CJTPAT on the link.
- Local error counters, capable of counting received CRC errors, as well as received NAKs.
- Link monitoring facilities that are able to monitor primitives on the link.
- Local management indicators on each device that indicate the state of the link.
- SAS Test Channel, ideally with a channel response in accordance with the SAS TCTF specifications.

Last Modification: March 24, 2005

Discussion:
This test is designed to verify the ability of a target and initiator to exchange packets with each other. The exchange of packets should produce a packet error ratio that is low enough to meet a desired bit error ratio. The bit error ratio as specified in reference [1] is 10^{-12}. For this test, $3x10^{12}$ bits will be transmitted between the testing station and the DUT. This will ensure that the bit error ratio is less than 10^{-12} with 95% accuracy if no errors are observed. Please note that the derivation of these numbers is presented in Appendix A of this test suite.

The initiator is instructed to transmit read and write commands to the target, through the SAS channel. The SAS Line Monitor monitors the channel for NAKs. The target should generate NAKs if data is received in error. If more than 7 NAKs occur during the exchange the bit error rate criterion has not been met and the test fails. In addition to packets lost, local management information may make it possible to isolate the packet loss to either the transmit side or the receive side of the test channel relative to the DUT.

The observable in this testing process is one or more packet counters. Since a single packet contains many bits, the measurement technique does not really measure the bit error rate. However, due to the large amount of both bits and frames that are being sent, all errors will be treated equally. Some devices may have the ability to count multiple errors in a single frame, while others may only count one error per frame. For this test, both these cases are considered equivalent. This means that all errors observed in a frame will be counted towards the maximum allowed number of 1 error per frame. A device may in theory pass a test with a much higher bit error rate than that which is being measured. However, given that any one bit in error will corrupt the packet, multiple errors within a packet do not, in practice, make a difference in the number of packets that must be retransmitted on real links. Thus, a short-term clock deviation that causes a bit error rate of 5 bits in a stream of 10^{12} bits will, under most conditions, cause as many packet errors as a device with a bit error rate of 1 in 10^{12}.

This test is an interoperability test. Failure of this test does not mean that a DUT is necessarily non-conformant. It does suggest that a problem in the ability of the devices to work properly together exists and further work should be done to isolate the cause of the failure.

Test Setup:
The device pair is connected through a reference test channel. Both devices are powered on. The link has been verified using local management information. The SAS System includes a SAS HBA and SAS Target. The SAS Target has been discovered by the SAS HBA Host System and the Host System has created a partition on the SAS Target. Figure 1 shows a possible test setup:

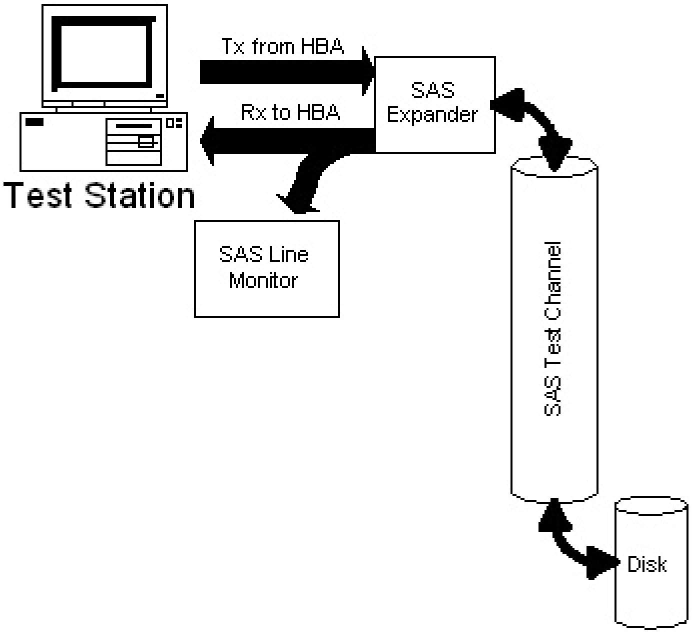

Figure 1: Possible test setup

Procedure:

1. Connect the test station, expander, test channel, and line monitor as shown in Figure 1.
2. Reset all counters that will be used to measure or monitor the exchange of packets between the DUT and the testing station. Configure software as needed.
3. Using the test station, transmit 585937500 512 byte frames (for a total of $3x10^{12}$ bits) to the link partner.
4. Observe the number of NAKs that occurred during the test period.

Observable Results:

Using the counters on the transmitter, identify the number of packets received. The difference between the number of packets transmitted and the number received is the number of lost packets. This value should be examined with other information gathered during the testing process to ensure that the failure is due to bit errors and not resource errors on the DUT or testing stations. In the ideal case all lost packets are identified on one of the testing stations or the DUT as either a CRC error, or some other type of receiver error. If the local information gathered from the DUT is reliable it is often possible to isolate the failure to either the transmitter channel or the receiver channel. It should be noted that an observed BER of greater than 10^{-12} does not necessarily imply two devices are not interoperable. Additionally, an observed BER of greater than or less than 10^{-12} does not imply compliance or non-compliance of either device.

Possible Problems:

- If errors exist within the testing station’s generation circuitry, or an with the line monitor, then the spirit of this test (BER verification of the SAS physical channel) is compromised.
- Verify the eye opening of the transmitter from the test station and the expander. Verify that the line monitor has little impact on the channel between the test station and expander.
- It is possible that some devices may not be able to properly compensate for clock tolerances with a minimum gap between the frames. It may be necessary to adjust this gap length accordingly.

GROUP 2: POINT-TO-MULTIPOINT INTEROPERABILITY

Overview:

This group of tests verifies the interoperability of a system of multiple SAS end devices connected through multiple SAS expanders.

Scope:

Comments and questions regarding the implementation of these tests are welcome, and may be forwarded to David Woolf, UNH InterOperability Lab (djwoolf@iol.unh.edu).

Test #2.1: Large System Build for Single Initiator Systems

Purpose: To determine if a SAS system consisting of a single initiator, multiple expanders, and multiple targets, can link and exchange data.

References:
[1] T10 Std. SAS

Resource Requirements:

- A reference set of stations that can be used as expanders, targets, SCSI initiators.
- Link monitoring facilities that are able to monitor primitives on the link.
- Local error counters, capable of counting received CRC errors, as well as received NAKs.
- Traffic Generator facilities, capable of generating a stressing pattern on the link (CJTPAT).
- Local management indicators on the DUT and reference set that indicate the state of the link as perceived by the different stations.

Last Modification: May 3, 2005

Discussion:

SAS is designed to allow many phys within one domain. Thus it is necessary to test how individual phys will interoperate within a large system. One difficulty in testing large systems is isolating problem devices. Hence in this test, devices are added one at a time to the system, and a series of tests are performed before another device is added. In this way problem isolation is made easier.

This test builds a large SAS system starting with one SAS Initiator, and adding a SAS expander, and then SAS targets until the expander is full. Then additional expanders are daisy chained to the first expander and the process of adding targets begins again.

Once a set of tests has been performed on an added device, a traffic stream is started from the Initiator to the added device. This traffic stream stays active throughout the rest of the test as other devices are added.

This test is an interoperability test. Failure of this test does not mean that a DUT is necessarily non-conformant. It does suggest that a problem in the ability of the devices to work properly together exists and further work should be done to isolate the cause of the failure.

Test Setup: Initiator 1, and Target 1 are connected through Expander 1. All devices are powered off.

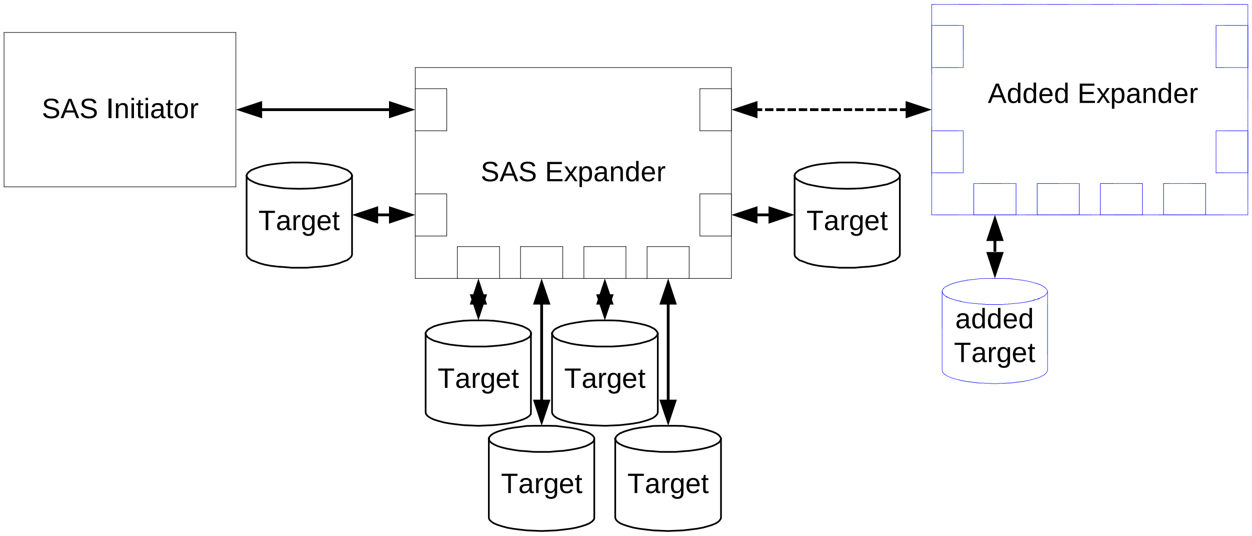

Figure 1: Large System Build ready for additional expander.

Procedure:

1. Power on Expander 1.
2. Power on Target 1.
    a. Using local management information verify that Target 1 is visible from Expander 1.
3. Power on Initiator 1.
    a. Using local management information verify that Initiator 1 is visible from Expander 1.
    b. Using local management information verify that Target 1 is visible from Initiator 1.
4. Create a partition on Target 1.
5. Using local traffic generation facilities, begin a traffic stream from Initiator 1 to Target 1.
6. Disconnect then reconnect Target 1 from the system.
    a. Verify that SCSI traffic resumed when the Target was reconnected.
7. Disconnect then reconnect Initiator 1 from the system.
    a. Verify that SCSI traffic resumed when the Initiator was reconnected.
8. Using local traffic generation facilities, generate a data stream containing a stressing pattern (CJTPAT) on the link.
    a. Using local error counters verify that bit errors are not occurring.
    b. If RX CRC errors are detected, verify that the number of errors observed correlates with the number of received NAKs indicated by the Link Partner.
9. Using local traffic generation facilities, begin a traffic stream (not CJTPAT) from Initiator 1 to Target 1.
10. Connect and power on an additional target device to the expander.
11. Repeat steps 2, 3b, 4-8 for the newly added Target. Continue with additional targets until only 1 port on Expander 1 is not connected. For each added target verify that traffic streams started in previous executions of step 8 are still active.
12. Connect a new expander (Expander 2) to the one open port on Expander 1. Power on the Expander 2.
13. Connect and power on an additional target device to the new expander.
14. Repeat steps 2, 3b, 4-8 for the newly added Target. Continue with additional targets until only 1 port on the new expander is not connected.
15. Continue to add expanders and targets, repeating steps 11-14.

Observable Results:

- As devices are added to the system use local management information to verify that added devices are visible from all other devices in the system.
- Verify SCSI traffic resumes when devices are disconnected and reconnected from the system.
- Verify that minimal errors occur in the system when a stressing pattern is sent between an initiator and target.

Possible Problems: Local management information may not be available, depending on the device. Daisy chained expanders need to be configured to use interoperable routing methods. The same upstream routing method must be used throughout the SAS System (i.e Table to Subractive or Subtractive to Table).

APPENDICES

Overview:

Test suite appendices are intended to provide additional low-level technical detail pertinent to specific tests contained in this test suite. These appendices often cover topics that are outside of the scope of the standard, and are specific to the methodologies used for performing the measurements in this test suite. Appendix topics may also include discussion regarding a specific interpretation of the standard (for the purposes of this test suite), for cases where a particular specification may appear unclear or otherwise open to multiple interpretations.

Scope:

Test suite appendices are considered informative supplements, and pertain solely to the test definitions and procedures contained in this test suite.

Appendix A – Packet Error Rate Measurement

Purpose: To develop a procedure for bit error rate measurement through the application of statistical methods.

References:

[1] Miller, Irwin and John E. Freund, Probability and Statistics for Engineers (Second Edition), Prentice-Hall, 1977, pp. 194-210, 240-245.

Last Modification: November 4, 2004 (Version 1.0)

Discussion:

A.1 – Introduction

One key performance parameter for all digital communication systems is the bit error rate (BER). The bit error rate is the probability that a given bit will be received in error. The BER may also be interpreted as the average number of errors that would occur in a sequence of n bits.

While the bit error rate concept is quite simple, the measurement of this parameter poses some significant challenges. The first challenge is deciding the number of bits, n, that must be sent in order to make a reliable measurement. For example, if 10 bits were sent and no errors were observed, it would be foolish to conclude that the bit error rate is zero. However, common sense tells us that the more bits that are sent without error, the more reasonable this conclusion becomes. In the interest of keeping the test duration as short as possible, we want to send the smallest number of bits that provides us with an acceptable margin of error.

This brings us to the second challenge of BER measurement. Given that we send n bits, what reasonable statements can be made about the bit error rate based on the number of errors observed? Returning to the previous example, if 10 bits are sent and no errors are observed, it is unreasonable to say that the BER is zero. However, it may be more reasonable to say that the BER is 10^{-1} or better. Furthermore, you are absolutely certain that the bit error rate is not 1.

In this appendix, two statistical methods, hypothesis testing and confidence intervals, are applied to help us answer the questions of how many bits we should be sent and what conclusions can be made from the test results.

A.2 – Statistical Model

A statistical model for the number of errors that will be observed in a sequence of n bits must be developed before we apply the aforementioned statistical methods. For this model, we will assume that every bit received is an independent Bernoulli trial. A Bernoulli trial is a test for which there are only two possible outcomes (i.e. a coin toss). Let us say that p is the probability that a bit error will occur. This implies that the probability that a bit error will not occur is (1-p).

The property of independence implies that the outcome of one Bernoulli trial has no effect on the outcomes of the other Bernoulli trials. While this assumption is not necessarily true for all digital communications systems, it is still used to simplify the analysis.

The number of successful outcomes, k, in n independent Bernoulli trials is taken from a binomial distribution. The binomial distribution is defined in equation A-1.

$$b(k; n, p) = C_{n,k}\, p^{k} (1-p)^{n-k} \qquad \text{(Equation A-1)}$$

Note that in this case, a successful outcome is a bit error. The coefficient $C_{n,k}$ is referred to as the binomial coefficient or “n-choose-k”. It is the number of combinations of k successes in n trials. Returning to coin toss analogy, there are 3 ways to get 2 heads from 3 coin tosses: (tails, heads, heads), (heads, tails, heads), and (heads, heads, tails). Therefore, $C_{3,2}$ would be 3. A more precise mathematical definition is given in equation A-2.

$$C_{n,k} = \frac{n!}{k!(n-k)!} \qquad \text{(Equation A-2)}$$

This model reflects the fact that for a given probability, p, a test in which n bits are sent could yield many possible outcomes. However, some outcomes are more likely than others and this likelihood principle allows us to make conclusions about the BER for a given test result.

A.3 – Hypothesis Test

The statistical method of hypothesis testing will allow us to establish a value of n, the number of bits to be sent, for the BER measurement. Naturally, the test begins with a hypothesis. In this case, we will hypothesize that the probability of a bit error, p, for the system is less than some target BER, P_0. This hypothesis is stated formally in equation A-3.

$$H_0 : p \le P_0 \qquad \text{(Equation A-3)}$$

We now construct a test for this hypothesis. In this case, we will take the obvious approach of sending n bits and counting the number errors, k. We will interpret the test results as shown in table A-1.

Table A-1: Acceptance and rejections regions for H_0

Test Result	**Conclusion**
k = 0	H_0 is true
k > 0	H_0 is false

We now acknowledge the possibility that our conclusion is in error. Statisticians define two different categories of error. A type I error is made when the hypothesis is rejected even though it is true. A type II error is made when the hypothesis is accepted even though it is false. The probability of a type I and a type II error are denoted as α and β respectively. Table A-2 defines type I and type II errors in the context of this test.

Table A-2: Definitions of type I and type II errors

Type I Error	k > 0 even though p ≤ BER
Type II Error	k = 0 even though p > BER

A type II error is arguably more serious and we will define n so that the probability of a type II error, β, is acceptable. The probability of a type II error is given in equation A-4.

$$\beta = (1-p)^n < (1-P_0)^n \qquad \text{(Equation A-4)}$$

Equation A-4 illustrates that the upper bound on the probability of a type II error is a function of the target bit error rate and n. By solving this equation for n, we can determine the minimum number of bits that need to sent in order to verify that p is less than a given P_0 for a given probability of type II error.

$$n > \frac{\ln(\beta)}{\ln(1-P_0)} \qquad \text{(Equation A-5)}$$

Let us now examine the probability of a type I error. The definition of α is given in equation A-6.

$$\alpha = 1-(1-p)^n \le 1-(1-P_0)^n \qquad \text{(Equation A-6)}$$

Equation A-6 shows that while we increase n to make β small, we simultaneously raise the upper bound on α. This makes sense since the likelihood of observing a bit error increases with the number of bits that you send, no matter how small bit error rate is. Therefore, while the hypothesis test is very useful in determining a reasonable value for n, we must be very careful in interpreting the results. Specifically, if we send n bits and observe no errors, we are confident that p is less than our target bit error rate (our level of confidence depends on how small we made β). However, if we do observe bit errors, we cannot be quick to assume that the system did not meet the BER target since the probability of a type I error is so large. In the case of $k > 0$, a confidence interval can be used to help us interpret k.

A.4 – Confidence Interval

The statistical method of confidence intervals will be used to establish a lower bound on the bit error rate given that $k > 0$. A confidence interval is a range of values that is likely to contain the actual value of some parameter of interest. The interval is derived from the measured value of the parameter, referred to as the point estimate, and the confidence level, $(1-\alpha)$, the probability that the parameter's actual value lies within the interval.

A confidence interval requires a statistical model of the parameter to be bounded. In this case, we use the statistical model for k given in equation A-1. If we were to compute the area under the binomial curve for some interval, we would be computing the probability that k lies within that interval. This concept is shown in figure A-1.

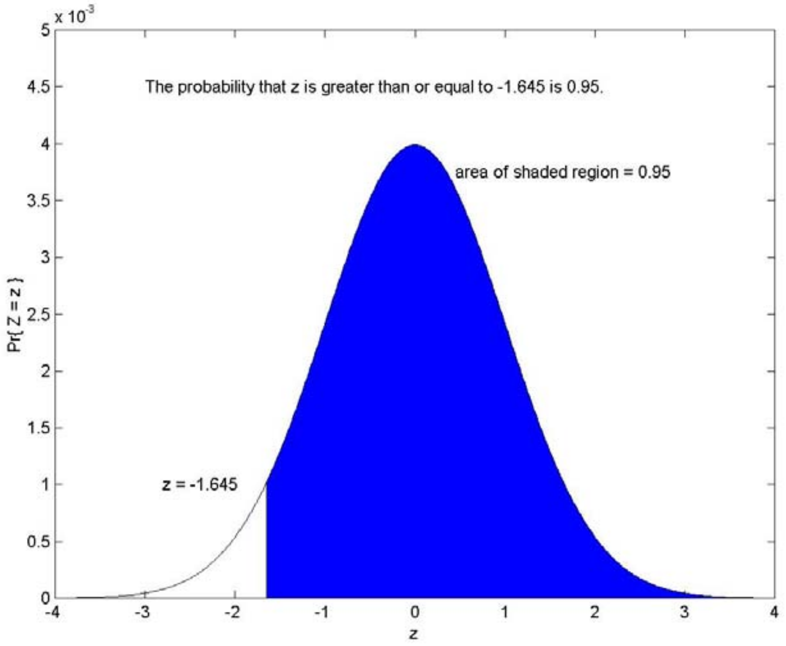

Figure A-1: Computing the probability that $z \geq -1.645$ (standard normal distribution).

To compute the area under the binomial curve, we need a value for the parameter p. To compute a confidence interval for k, you assume that k/n, the point estimate for p, is the actual value of p.

Note that figure A-1 illustrates the computation of the lower tolerance bound for k, a special case where the confidence interval is $[k_l, +\infty]$. A lower tolerance bound implies that in a percentage of future tests, the value of k will be greater than k_l. In other words, actual value of k is greater than k_l with probability equal to the confidence level. Therefore, if k_l/n is greater than P_0, we can say that the system does not meet the target bit error rate with probability $(1-\alpha)$. By reducing α, we reduce the probability of making a type I error.

To determine the value of k_l, it is useful to assume that the binomial distribution can be approximated by a normal (Gaussian) distribution when n is large. The mean and variance of this equivalent distribution are the mean and variance of the corresponding binomial distribution (given in equations A-7 and A-8).

$$\mu_K = np \qquad \text{(Equation A-7)}$$

$$\sigma_K^2 = np(1-p) \qquad \text{(Equation A-8)}$$

Now, let α be the probability that $Z \le z_\alpha$ where Z is a standard normal random variable. A standard random variable is one whose mean is zero and whose variance is one. The random variable K can be standardized as shown in equation A-9.

$$Z = \frac{K - \mu_K}{\sigma_K} \qquad \text{(Equation A-9)}$$

Note that Z is greater than z_α with probability (1-α), the confidence level. We apply this inequality to equation A-9 and solve for K to get equation A-10.

$$\begin{aligned} K &> \mu_K + z_\alpha \sigma_K \\ K &> np + z_\alpha \sqrt{np(1-p)} \end{aligned} \qquad \text{(Equation A-10)}$$

As mentioned before, we assume that p is k/n. We can now generate an expression for k_l, the value that K will exceed with probability (1-α). This expression is given in equation A-11.

$$k_l = k + z_\alpha n \sqrt{\frac{(k/n)(1-k/n)}{n}} \qquad \text{(Equation A-11)}$$

Finally, we argue that if K exceeds k_l, then the actual value of p must exceed k_l/n. Therefore, we can generate an expression for p_l, the value that p will exceed with probability (1-α), and compare it to the target bit error rate. By applying this comparison (given in equation A-12) the probability of a type I error can be greatly reduced. For example, by setting z_α to –1.645, the probability of a type I error is reduced to 5%.

$$P_0 \ge p_l = \frac{k_l}{n} = \frac{k}{n} + z_\alpha \sqrt{\frac{(k/n)(1-k/n)}{n}} \qquad \text{(Equation A-12)}$$

A.5 – Sample Test Construction

We now compress the theory presented in sections A-2 through A-4 into two inequalities that may be used to construct a bit error rate test. First, we take equation A-5 and assume that $\ln(1-P_0)$ is $-P_0$ (valid for P_0 much less than one). The result is equation A-13.

$$n > \frac{-\ln(\beta)}{P_0} \qquad \text{(Equation A-13)}$$

Second, we examine equation A-12. Assuming that (1-k/n) is very close to 1 and substituting $-\ln(\beta)/P_0$ for n, we get equation A-14.

$$-\ln(\beta) \ge k + z_\alpha \sqrt{k} \qquad \text{(Equation A-14)}$$

The largest value of k that satisfies equation A-14 is k_l. The benefit of these two equations is that a bit error rate test is uniquely defined by β and α and that the test scales with P_0. Table A-3 defines n and k_l in terms of β and α.

Table A-3: n and k_l as a function of β and α.

β	**$-\ln(\beta)$**	**n**	**α**	**z_α**	**k_l**
0.10	2.30	$2.30/P_0$	0.10	–1.29	5
0.10	2.30	$2.30/P_0$	0.05	–1.65	6
0.05	3.00	$3.00/P_0$	0.05	–1.65	7
0.05	3.00	$3.00/P_0$	0.01	–2.33	10
0.01	4.60	$4.60/P_0$	0.05	–1.65	9
0.01	4.60	$4.60/P_0$	0.01	–2.33	13

As an example, let us construct a test to determine if a given system is operating at a bit error rate of 10^{-12} or better. Given that a 5% chance of a type I error is acceptable, the test would take the form of sending $3x10^{12}$ bits and counting the number of errors. If no errors are counted, we are confident that the BER was 10^{-12} or better.

Given that a 5% chance of a type II error is acceptable, we find that k_l is 7. If more than 7 errors are counted, we are confident that the bit error rate is greater than 10^{-12}. However, what if between 1 and 7 errors are counted? These cases may be handled several different ways. One option is to make a statement about the bit error rate (whether it is less than or greater than 10^{-12}) at a lower level of confidence. Another option would be to state that the test result is success since we cannot establish with an acceptable probability of error that the BER is greater than 10^{-12}. Such a statement implies that we failed to meet the burden of proof for the conjecture that the BER exceed 10^{-12}. Of course, the burden of proof could be shifted to the device under test which would imply that any outcome other than $k = 0$ would correspond to failure (the device under test failed to prove to us that the BER was no more than 10^{-12}). If neither of these solutions are acceptable, it is always an option to perform a more vigorous bit error rate test in order to clarify the result.

A.6 – Packet Error Rate Measurement

It is often easier to measure packet errors than it is to measure bit errors. In these cases, it is helpful to have some linkage between the packet error rate and the bit error rate. To make this linkage, we assume that the bit error rate is low enough and the packet size is small enough so that each packet error contains exactly one bit error.

To complete the linkage, some care must be taken regarding how many packets to send. A bit error is only detectable in the region of the packet that is covered by the cyclic redundancy check (CRC). In the context of SAS, this region is the first bit after the SOF to the last bit of the CRC. There is no guarantee that errors in the SOF, EOF, and other regions will be detected. Therefore, we must translate n from the number of bits are sent to the number of “observable” bits that are sent. This will increase the test duration since a portion of the time will be spent sending unobservable bits.

For packets of length x bits, at least n/x packets must be sent to perform the equivalent bit error rate test. If no packet errors are observed, the conclusion is that the bit error rate is less than P_0. If more than k_l packet errors are observed, the conclusion is that the bit error rate is greater than P_0.

Note that x is the length of the packet after encoding. In other words, in an 8B10B encoding environment, a 512-byte packet is 5120 bits in length after encoding. Also note that to reinforce the assumption that there is only one bit error per packet error, a test should be run with the shortest possible packets. However, if extremely low bit error rates are to be verified, it may be favorable to use long packets to increase the percentage of observable bits and reduce the test duration.

Appendix B – Optional In-System Physical Layer Measurements

Purpose: To define an optional set of physical layer measurements that may be performed on a pair of SAS devices, with respect to a specific system implementation.

References:

[1] SAS Standard, Clause 5
[2] UNH IOL SAS Physical Layer Test Suite, Appendix 5.B – Alternate Jitter Test Setup

Last Modification: January 20, 2005 (Version 1.0)

Discussion:

B.1 – Introduction

The SAS Standard defines a set of specifications that govern the physical layer characteristics of SAS devices. The conformance tests of Clause 5 are defined with respect to a specific SAS transceiver, where various electrical signaling characteristics of a transmitter device are measured under different test load conditions. These load conditions are comprised of two cases: 1) an 'ideal' zero-length termination attached directly to the transmitter device, and 2) a controlled 'worst-case' maximum-attenuation test channel that meets the TCTF requirements defined in Clause 5. Most of the conformance tests are performed with one or both test load cases, the general intention being that if the device can demonstrate conformant signaling characteristics across both the best and worst-case channels, it is more likely to successfully interoperate in typical SAS environments (provided the Link Partner's signaling is also conformant, and the devices are connected across a compliant SAS channel).

While the method of specifying transceiver conformance with respect to ideal best and worst-case test load conditions is a perfectly valid scientific approach, there may be cases where there is additional value in observing the physical layer characteristics of a device *in system*, i.e., with respect to a specific implementation, Link Partner, and channel. In addition to performing a system-level Packet Error Rate Test, there are several physical layer signaling measurements that can be performed that may provide some insight into how much margin a device may have in a given implementation. Because the termination characteristics of actual SAS devices are not likely to equal the ideal termination provided by a DSO, the optimal amplitude and emphasis settings for a given application may not be the same as those used for performing the formal conformance tests. Therefore, performing a set of basic in-system physical layer tests also allows for further performance optimization of a specific system. In this appendix, a modified set of tests will be defined, which, when performed in conjunction with the Packet Error Rate Test, may allow for better system performance.

It should be stated that these measurements are not intended to serve as a substitute for formal conformance tests, but rather are presented as informative supplement to the system level Packet Error Rate Interoperability Test. While they are not a requirement for performing the Packet Error Rate test, they are offered as an optional set of measurements, for instances mainly where an in-depth evaluation of a specific system is desired. (As opposed to the case where a single device is being tested against multiple devices in the interoperability test bed, in which case it may not be practical to perform the physical layer measurements for every test case.) These tests may also be used for optimizing the transmitter amplitude and emphasis settings for a specific system implementation prior to running the formal Packet Error Rate Test, in order to maximize system performance.

B.2 – List of Tests

The proposed tests for this appendix are all based on tests defined in Groups 1 and 2 of the UNH IOL SAS Physical Layer Test Suite. The list is a modified subset of these tests, which are performed using the setup described in Appendix 5.B of the Physical Layer Test Suite (*Alternate Jitter Test Setup*). The measurements are all performed using a differential probe, which is placed at the transmitter and receiver pins of the two transceivers comprising the System Under Test.

Table B-1, below, summarizes the tests of Groups 1 and 2 of the IOL SAS Physical Layer Test Suite, and describes the modifications that are required to perform the tests on devices operating in a functional SAS system. In the table, ‘near end’ denotes the TX signal, as probed directly at the transmitter IC output pins. ‘Far end’ denotes the TX signal probed at the output of the test channel, at the RX pins of the opposite transceiver IC.

The tests in Table B-1 below are intended to be performed twice, once for each transceiver in the system.

Table B-1: Group 1&2 Tests, with Modification Notes

Test	Name	Comments
5.1.1	TX Maximum Transients	N/A
5.1.2	RX Maximum Transients	N/A
5.1.3	TX Device Off Voltage	Measure at near-end
5.1.4	TX OOB Offset Delta	Measure at near-end
5.1.5	TX OOB Common Mode Delta	N/A
5.1.6	TX Minimum OOB Align Burst Amplitude	Measure at near-end and far-end
5.1.7	TX Maximum Noise During OOB Idle	Measure at near-end and far-end
5.2.1	TX Bit Rate	Measure at near-end
5.2.2	TX Jitter	Measure at far-end
5.2.3	TX Output Imbalance	N/A
5.2.4	TX Rise and Fall Times	Measure at near-end
5.2.5	TX Skew	N/A

Of the tests above, several are marked ‘N/A’, which are not applicable to the in-system environment for various reasons. The TX and RX transient tests are not likely to produce any meaningful results when performed in-system (as it would be difficult to determine which device was causing the transients), and are best performed separately on each device. Tests 5.1.5, 5.2.3, and 5.2.5 cannot be performed in-system, as they require separate access to each half of the differential signal pair, which are not accessible when using a differential probe.

Of the tests than can be performed, test 5.1.3 (TX Device Off Voltage) is not likely to offer any additional information as it is a relatively straightforward test, however it is included here for completeness. Test 5.1.4 (TX OOB Offset Delta) could potentially offer insight into the symmetry of the test channel and link partner receiver IC termination (which would manifest itself as an offset delta if not properly balanced.) The TX minimum OOB Align Burst Amplitude measurement (test 5.1.6) will provide insight into the relative robustness of the OOB bursts at both ends of the test channel.

Test 5.1.7 (TX Maximum Noise During OOB Idle) is especially useful in cases where the test channel is a backplane or midplane, which may contain other active components capable of creating excessive noise that could potentially affect the successful completion of OOB, or adversely affect data signaling.

Test 5.2.1 (TX Bit Rate) could provide useful insight into the relative bit rate speed difference between the two system devices, which could theoretically result in interoperability issues in cases where device TX/RX tolerances happen to be excessively mismatched.

Test 5.2.4 (TX Rise and Fall Times) could be useful in determining the in-system rise and fall times of the data signaling, which could theoretically be linked to EMI and crosstalk-related interoperability issues, caused by excessive high-frequency signal energy associated with faster rise times.

Perhaps the most useful of these tests however, is the TX jitter test (5.2.2), as it provides insight into the amount of margin contained in the signaling eye appearing at each receiver device. This information, when combined with the results of the Packet Error Rate Test, can provide a quantifiable estimate of the robustness of a particular system, including the amount of margin found in each side of the link (for cases where the BER is shown to be at least 1E-12).

B.3 – Conclusion

In this appendix, a series of optional, informative physical layer measurements is defined, with the purpose of observing certain physical layer signaling characteristics of two devices, under operation in a specific system implementation. While these tests are based on formal conformance tests defined in the UNH IOL SAS Physical Layer Test Suite, they are performed in a modified form, and are not intended to serve as a substitute for formal conformance tests. Rather, they are meant to augment the results of the Packet Error Rate Test for two devices, by providing additional insight into the amount of margin two devices may contain when operating in a specific SAS system implementation.

In general, these tests, as well as the Packet Error Rate Test, are best performed in addition to the full set of UNH IOL SAS conformance test suites, which should ideally performed on each of the system devices separately, before attempting to verify the functionality of the entire system.