UNH–IOL
NVMe Testing Service

Test Plan for NVMe-oF Conformance
Version ~~1.1~~1.2.0
Target Specification: NVMe-oF 1.0a
Technical Document

Last Updated ~~November 27, 2018~~April 24, 2019

UNH–IOL NVMe Testing Service
21 Madbury Rd Suite 100
Durham, NH 03824

Tel: +1 603–862–0090
Fax: +1 603–862–4181
Email: nvmelab@iol.unh.edu

TABLE OF CONTENTS

Formatted: Font: Times

MODIFICATION RECORD

2018 November 27 (Version 11.0) Initial Release
David Woolf:

2019 April 24 (Version 12.0) Final Release
David Woolf:

1. Updates to discussion in Test 2.1 per TP 2008.
2. Fixed typo in Test 2.1 Case 1 procedure, per TP 2008.
3. Added text to each test procedure to clarify the necessity that Testing Station and DUT are on the same fabric network, and that disconnect should be performed at the conclusion of each test.
4. Added new FYI Test 3.2 Case 2. Clarified procedure that a Set Feature command is needed to specify the KATO value.

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.5" + Indent at: 0.75"

ACKNOWLEDGMENTS

The UNH–IOL would like to acknowledge the efforts of the following individuals in the development of this test plan:

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 a neutral environment where a product can be tested against other implementations of a common standard, both in terms of interoperability and conformance. This particular suite of tests has been developed to help implementers evaluate the NVMe-oF functionality of their products. This test suite is aimed at validating products in support of the work being directed by the NVMe Promoters Group.

These tests are designed to determine if a product conforms to specifications defined in the NVMe-oF version 1.0a specification, hereafter referred to as the "NVMe-oF Specification"). Successful completion of these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many NVMe-oF environments.

Products implementing the NVMe-oF specification may use different transports, such as RoCE, TCP, Fibre Channel, Infiniband, or iWARP. Each of these different transports may have distinct requirements described in a binding specification that is either within the NVMe-oF specification, or a separate document. The tests described in this Test Suite document are intended to check NVMe-oF requirements that will apply to all NVMe-oF requirements regardless of the transport used. Therefore, requirements that are described within the binding specification do not have corresponding test items in this document.

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 two–number, dot–notated naming system is used to catalog the tests. 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 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. These procedures should be the ideal test methodology, independent of specific tool limitations or restrictions.

Observable Results
This section lists the specific observable items 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.

REFERENCES

The following documents are referenced in this text:

1. NVMe-oF version 1.0a Specification (July 17, 2018)

ABBREVIATIONS

The following abbreviations are applied to the test titles of each of the tests described in this document for indicating the status of test requirements.

M - Mandatory
FYI - FYI
IP - In Progress

Group 1: NVMe-oF Commands

Overview:

This section describes a method for performing conformance verification for NVMe-oF products implementing NVMe-oF Commands.

Notes:

The preliminary draft descriptions for the tests defined in this group are considered complete, and the tests are pending implementation (during which time additional revisions/modifications are likely to occur).

Test 1.1 – Connect to Discovery Service (FYI)

Purpose: To verify that an NVMe-oF Controller can properly execute a Connect command.

References:

[1] NVMe-oF Specification 3.3, 5

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport.

Last Modification: November 27, 2018

Discussion: The host uses the well-known Discovery Service NQN (nqn.2014-08.org.nvmexpress.discovery) in the Connect command to a Discovery Service. The method that a host uses to obtain the NVMe Transport information necessary to connect to the well-known Discovery Service is implementation specific.

Test Setup: See Appendix A.

Case 1: Connect to Discovery Controller (FYI)

Test Procedure:

1. This test is only applicable to DUTs which implement a Discovery Controller.
2. ~~The Testing Station has established a transport connection. Establishment of the transport connect will vary depending on the transport type.~~ The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
3. ~~2.~~ All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
4. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT using well known address nqn.2014-08.org.nvmexpress.discovery.
5. ~~3.~~ Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Formatted: Font color: Auto

Observable Results:

1. Verify that the Connect Response indicates status Success and provides.

Possible Problems: None.

Test 1.2 – Connect to NVM Subsystem (FYI)

Purpose: To verify that an NVMe-oF Controller can properly execute a Connect command.

References:
[1] NVMe-oF Specification 3.3

Resource Requirements:
Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport.

Last Modification: November 27, 2018

Discussion: The Connect command is used to create a Submission and Completion Queue pair. If the Admin Queue is specified, then the Connect command establishes an association between a host and a controller.

Test Setup: See Appendix A.

Case 1: Connect to NVM Subsystem (FYI)

Test Procedure:
1. This test is not applicable to a Discovery Controller.
2. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
3. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
4. The Testing Station has discovered the NVM Subsystem.
2. ~~The Testing Station has discovered the NVM Subsystem under test and a transport connection has been established. Discovery and establishment of the transport connect will vary depending on the transport type.~~
5. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT. The Connect command will have a Queue ID of 0 to establish Admin Submission and Completion Queues.
~~3.~~6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Formatted: Font color: Auto

Observable Results:
1. Verify that the Connect Response indicates status Success and provides the Controller ID allocated to the Host in DWord 0.

Case 2: Connect with Incompatible Format (FYI)

Test Procedure:
1. This test is not applicable to a Discovery Controller.
2. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
~~1.~~3. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
~~2.~~4. The Testing Station has discovered the NVM Subsystem ~~under test and a transport connection has been established. Discovery and establishment of the transport connect will vary depending on the transport type~~.
5. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT. The Connect command will have a Queue ID of 0 to establish Admin Submission and Completion Queues. The Record Format (RECFMT) will be a value of FEh, which is expected to be an unsupported value.
~~3.~~6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Formatted: Font color: Auto

Formatted: Font color: Auto

Observable Results:
1. Verify that the DUT responded to the received Connect Command with a status of 'Connect Incompatible Format'.

Possible Problems: None.

Test 1.3 – Property Get Command (FYI)

Purpose: To verify that an NVMe-oF Controller can properly execute a Property Get command.

References:
[1] NVMe-oF Specification 3.4, 3.5.1

Resource Requirements:
Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport.

Last Modification: November 27, 2018

Discussion: The Property Get command is used to specify the property value to return to the host. If an invalid property or invalid offset is specified, then a status value of Invalid Field in Command shall be returned.

Test Setup: See Appendix A.

Case 1: Get Supported Properties (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
2. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
1. The Testing Station has discovered the NVM Subsystem under test and a transport connection has been established. Discovery and establishment of the transport connect will vary depending on the transport type.
2.3. An Establish an NVM connection has been established.and Admin and IO Submission and Completion Queues have been established.
3.4. Configure the Testing Station acting as a Host to send a Property Get command (FCTYPE=04h) with the proper offset value and size for each of the following properties:
    a. 00h-07h CAP
    b. 08h-0bh VS
    c. 14h-17h CC
    d. 1Ch-1Fh CSTS
d.5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that each of the Property Get commands completed successfully and that the DUT returned values for each requested property.
2. Verify that CAP.DSTRD is set to 0h.
3. Verify that CAP.CQR is set to 1h.

Case 2: Get Property Invalid Offset (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
2. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
1. Establish an NVM connection and Admin and IO Submission and Completion.The Testing Station has discovered the NVM Subsystem under test and a transport connection has been established. Discovery and establishment of the transport connect will vary depending on the transport type.
3.
2. An NVM connection has been established. Admin and IO Submission and Completion Queues have been established.

Formatted

Formatted: Font color: Auto

~~3.~~4. Configure the Testing Station acting as a Host to send a Property Get command (FCTYPE=04h) with the following offset value:
   a. 01h-07h CAP (an invalid offset that is not at the beginning of the property)
~~a.~~5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Formatted

Formatted: Font color: Auto

Observable Results:
1. Verify that the Property Get commands completed with status "Invalid Field in Command".

Case 3: Get Reserved Property (FYI)

Test Procedure:
1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
2. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
~~1.~~ Establish an NVM connection and Admin and IO Submission and Completion.~~The Testing Station has discovered the NVM Subsystem under test and a transport connection has been established. Discovery and establishment of the transport connect will vary depending on the transport type.~~
3.
~~2. An NVM connection has been established. Admin and IO Submission and Completion Queues have been established.~~
~~3.~~4. Configure the Testing Station acting as a Host to send a Property Get command (FCTYPE=04h) with the following offset value:
   a. 0Ch-0Fh INTMS
   b. 10h-13h INTMC
   c. 18h-1Bh Reserved
   d. 24h-27h AQA
   e. 28h-2Fh ASQ
   f. 30h-37h ACQ
   g. 38h-3Bh CMBLOC
   h. 3Ch-3Fh CMBSZ
   i. 40h-EFFh Reserved
   j. F00h-FFFh Reserved
   k. 1000h-12FFh Reserved
~~k.~~5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Formatted: Font color: Auto

Formatted

Observable Results:
1. Verify that the Property Get commands completed with status "Success" and all values returned were 0h.

Possible Problems: None.

Group 2: Controller Architecture

Overview:

This section describes a method for performing conformance verification for the Controller Architecture of NVMe-oF products..

Notes:

The preliminary draft descriptions for the tests defined in this group are considered complete, and the tests are pending implementation (during which time additional revisions/modifications are likely to occur).

Test 2.1 – Identify Controller Data Structure Enhancements (FYI)

Purpose: To verify that an NVMe Controller can properly populate Identify controller fields specific to NVMe over Fabrics.

References:

[1] NVMe Specification 4.1

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe transport interface.

Last Modification: ~~November 29, 2018~~April 2, 2019

Discussion: ~~Identify Controller Data Structure has enhancements specific for NVMe-oF implementations~~ Identify Controller Data Structure has enhancements specific for NVMe-oF implementations.

Test Setup: See Appendix A.

Case 1: CNS=~~00h~~ 01h Identify ~~Namespace~~ Controller Data Structure (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
2. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
3. Establish an NVM connection and Admin and IO Submission and Completion.
4. ~~1.~~ Configure the Testing Station acting as a Host to perform an Identify Controller Data Structure (CNS=01h).
5. ~~2.~~ In the Data Structure return, read the following values:
    a. Byte 1795:1792 I/O Queue Command Capsule Supported Size (IOCCSZ)
    b. Byte 1799:1796 I/O Queue Response Capsule Supported Size (IORCSZ)
    c. Byte 1801:1800 In Capsule Data Offset (ICDOFF)
    d. Byte 1802 Fabrics Controller Attributes (FCATT)
    e. Byte 1803 Maximum SGL Data Block Descriptors (MSDBD)
    f. Byte 2047:1804
6. ~~f.~~ Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the IOCCSZ has a value of 4 or greater, corresponding to 64 bytes or greater.
2. Verify that the IORCSZ has a value of 1 or greater, corresponding to 16 bytes or greater.
3. Verify that the FCATT field has bits 7:1 set to 0, as they are reserved.
4. Verify that Bytes 2047:1804 are set to 0, as they are Reserved.

Possible Problems: None.

Formatted

Formatted: Font color: Auto

Group 3: Discovery Service

Overview:

This section describes a method for performing conformance verification for NVMe-oF products implementing the Discovery Service.

Notes:

The preliminary draft descriptions for the tests defined in this group are considered complete, and the tests are pending implementation (during which time additional revisions/modifications are likely to occur).

Test 3.1 – Initialize Discovery Controller (FYI)

Purpose: To verify that an NVMe Controller can properly initialize a Discovery Controller.

References:
[1] NVMe-oF Specification 5.1

Resource Requirements:
Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport.

Last Modification: November 27, 2018

Discussion: The initialization process for the Discovery controller is described below:
1. NVMe in-band authentication is performed if required;
2. The host determines the controller's capabilities by reading the Controller Capabilities property;
3. The host configures the controller's settings by writing the Controller Configuration property, including setting CC.EN to '1' to enable command processing;
4. The host waits for the controller to indicate it is ready to process commands. The controller is ready to process commands when CSTS.RDY is set to '1' in the Controller Status property; and
5. The host determines the features and capabilities of the controller by issuing the Identify command, specifying the Controller data structure.

After initializing the Discovery controller, the host reads the Discovery Log Page.

Test Setup: See Appendix A.

Case 1: Initialize Discovery Controller (FYI)

Test Procedure:
1. This test is only applicable to DUTs which implement a Discovery Controller.
2. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
3. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
~~2. The Testing Station has established a transport connection has been established. Establishment of the transport connect will vary depending on the transport type.~~
~~3.~~4. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT using well known address nqn.2014-08.org.nvmexpress.discovery.
~~4.~~5. The Testing Station acting as a Host will use the Get Property Command to read the Controller Capabilities.
~~5.~~6. The Testing Station acting as a Host will use the Set Property Command to set CC.EN to 1.
~~6.~~7. The Testing Station acting as a Host will use the Get Property Command to read CSTS.RDY. Repeat until CSTS.RDY=1.
~~7.~~8. The Testing Station will transmit an Identify Controller Data Structure (CNS=01h).
9. The Testing Station will transmit a Get Log Page Command for the Discovery Log Page (LID=70h)
~~8.~~10. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Formatted: Font color: Auto

Observable Results:
1. Verify that the Connect Response indicates status Success.
2. Verify that the Get Properties commands indicates status Success.
3. Verify that the Set Properties command indicates status Success.
4. Verify that the Identify command indicates status Success and the following fields are populated correctly:
    a. Firmware Revision (FR)
    b. Maximum Data Transfer Size (MDTS)
    c. Controller ID (CNTLID)

   d. Version (VER)
   e. Log Page Attributes (LPA)
   f. Error Log Page Attributes (ELPE)
   g. Maximum Outstanding Commands (MAXCMD)
   h. SGL Support (SGLS)
   i. NVM Subsystem NVMe Qualified Name (SUBNQN)
5. Verify that the DUT returns the Discovery Log Page and the following fields are populated correctly:
   a. Transport Type (TRTYPE) should reflect the transport being used.
   b. Address Family (ADRFAM) should reflect the address and transport being used.
   c. Admin Max SQ Size (ASQSZ) indicates a minimum of 32.
   d. Transport Service Identifier (TRSVCID) reflects the transport being used.
   e. Record Format (RECFMT) is 0h.

Possible Problems: None.

Test 3.2 – Keep Alive Timeout (FYI)

Purpose: To verify that an NVMe Controller can properly process a Keep Alive Timeout.

References:

[1] NVMe-oF Specification 7.1.2

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport.

Last Modification: November 29, 2018

Discussion: The controller shall treat a Keep Alive Timeout in the same manner as connection loss. If the Keep Alive feature is in use and the timer expires, then the controller shall:

- stop processing commands and set the Controller Fatal Status (CSTS.CFS) bit to '1';
- terminate the NVMe Transport connection; and
- break the host to controller association.

After completing these steps, a controller may accept a Connect command for the Admin Queue from the same or another host in order to form a new association.

Test Setup: See Appendix A.

Case 1: Keep Alive Timeout (FYI)

Test Procedure:

1. This test is only applicable to DUTs which implement a Discovery Controller and a transport that requires the Keep Alive feature.
2. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
3. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
2. ~~The Testing Station has established a transport connection has been established. Establishment of the transport connect will vary depending on the transport type.~~
3. 4. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT using well known address nqn.2014-08.org.nvmexpress.discovery and a Set Feature command to set the desired KATO value.
4. 5. The Testing Station acting as a Host will use the Get Property Command to read the Controller Capabilities.
5. 6. The Testing Station acting as a Host will use the Set Property Command to set CC.EN to 1.
6. 7. The Testing Station acting as a Host will use the Get Property Command to read CSTS.RDY. Repeat until CSTS.RDY=1.
7. 8. The Testing Station will transmit an Identify Controller Data Structure (CNS=01h).
8. 9. The Testing Station will transmit a Get Log Page Command for the Discovery Log Page (LID=70h).
9. 10. The Testing Station should send at least 3 Keep Alive Commands using the prescribed Keep Alive Timeout. After sending 3 Keep Alive commands, the testing station should cease sending Keep Alive commands. Allow the Keep Alive timer to timeout.
10. 11. The Testing Station will use the Get Property Command to read the CSTS.CFS bit.
12. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT using well known address nqn.2014-08.org.nvmexpress.discovery.
11. 13. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that when the Keep Alive Timer times out the DUT sets CSTS.CFS to 1.
2. Verify that the final Connect command completes with status "Success".

Formatted: Font color: Auto

Case 2: Traffic Based Keep Alive (FYI)

Test Procedure:

1. This test is only applicable to DUTs which implement a Discovery Controller and a transport that requires the Keep Alive feature.
2. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address or WWN of the DUT and is able to send packets to the DUT.
3. All NVMe-oF NVM Subsystems are disconnected from the Testing Station acting as a Host.
4. Check CTRATT Bit 6. If CTRATT Bit 6 is not set to 1, then the DUT does not support TBKAT and this test is not applicable.
5. Configure the Testing Station acting as a Host to issue a Connect command (FCTYPE=01h) to the DUT using well known address nqn.2014-08.org.nvmexpress.discovery and a Set Feature command to set the desired KATO value.
6. The Testing Station acting as a Host will use the Get Property Command to read the Controller Capabilities.
7. The Testing Station acting as a Host will use the Set Property Command to set CC.EN to 1.
8. The Testing Station acting as a Host will use the Get Property Command to read CSTS.RDY. Repeat until CSTS.RDY=1.
9. The Testing Station will transmit an Identify Controller Data Structure (CNS=01h).
10. The Testing Station will transmit a Get Log Page Command for the Discovery Log Page (LID=70h).
11. The Testing Station should send at least 3 Keep Alive Commands using the prescribed Keep Alive Timeout.
12. After sending 3 Keep Alive commands, the testing station should cease sending Keep Alive command but continue performing READ commands separated by no more than the current KATO value. At least one KATO should expire with no Keep Alive Command being sent, but instead READ commands should be sent.
13. After sending 3 at least 3 READ commands, the testing station should cease sending READ commands but continue performing Identify commands separated by no more than the current KATO value. At least one KATO should expire with no Keep Alive or READ Commands being sent, but instead Identify commands should be sent.
14. The Testing Station will use the Get Property Command to read the CSTS.CFS bit.
15. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Keep Alive Timer does not expire, and CSTS.CFS (Controller Fatal Status) is not set to 0 during the test execution.
2. Verify that the Keep Alive, READ, and Identify commands all complete with status “Success”.

Possible Problems: None.

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

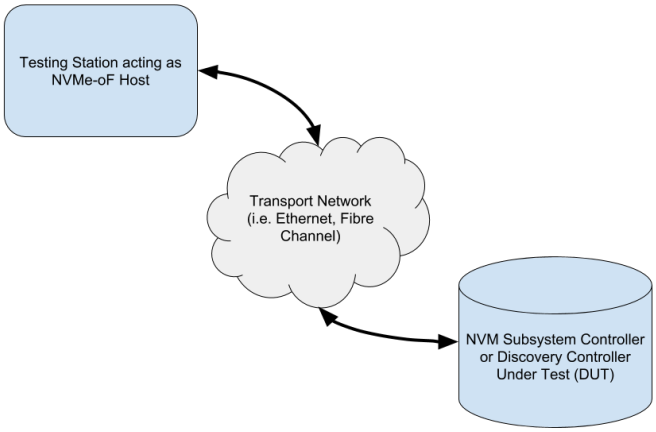
Formatted: Font color: Auto

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"

Appendix A: TEST SETUP

The diagram below outlines a simple test setup for testing NVMe-oF products for conformance. While a network is shown, it is also possible to perform the tests using a direct connection.



Appendix B: NOTES ON TEST PROCEDURES

There are scenarios where in test procedures it is desirable to leave certain aspects of the testing procedure as general as possible. In these cases, the steps in the described test procedure may use placeholder values, or may intentionally use non–specific terminology, and the final determination of interpretation or choice of values is left to the discretion of the test technician. The following is an attempt to capture and describe all such instances used throughout the procedures.

Ports on Testing Station and Device Under Test

In general, any NVMe-oF capable Port on the Testing Station or Device Under Test may be used as an interface with a test station or interoperability partner. There is *assumed* to be no difference in behavior, with respect to the protocols involved in this test suite, between any two NVMe-oF ports on the Testing Station or Device Under Test. Hence, actual ports used may be chosen for convenience. However, it is recommended that the port used in the test configuration is recorded by the test technician.

Use of "various"

To maintain generality, some steps will specify that "various other values" (or the like) should be used in place of a given parameter. Ideally, all possible values would be tested in this case. However, limits on available time may constrain the ability of the test technician to attempt this. Given this, a subset of the set of applicable values must generally be used.

When deciding how many values should be used, it should be noted that the more values that are tested, the greater the confidence of the results obtained (although there is a diminishing return on this).

When deciding which specific values to use, it is generally recommended to choose them at pseudo–randomly yet deterministically. However, if there exist subsets of the applicable values with special significance, values from each subset should be attempted.

Appendix C: TEST TOOLS

The Tests described in this document can be performed using available IOL INTERACT NVMe-oF Test Software available from UNH–IOL. As of the writing of this document, version of this tool were available that support NVMe/FC, NVMe/RoCE, and NVME/TCP transports.

If using the PC Edition of the IOL INTERACT NVMe-oF Test Software, UNH-IOL recommends using v11.0 or higher. This software is available via https://www.iol.unh.edu/solutions/test-tools/interact.

Appendix D: NVME-OF INTEGRATORS LIST REQUIREMENTS

Purpose: To provide guidance on what tests are required for NVMe-oF Integrators List Qualification

References:

[1] NVMe Integrators List Policy Document

Resource Requirements:

NVMe Host Platform and Device.

Last Modification: November 28, 2017

Discussion: Each Test defined in this document is defined as being Mandatory (M), FYI, or In Progress (IP). This primary designation is shown in the title of the test case and is understood to apply to any NVMe-oF product. The following examples are provided:

Test 1.1 Example Name (M)– Test is mandatory for all NVMe-oF products.
Test 2.1 Example Name (FYI)– Test is FYI for all NVMe-oF products.
Test 3.1 Example Name (IP)- Test is still under development, and is considered FYI all for NVMe-oF products.

If a Test is designated as Mandatory, a product must pass this test in order to qualify for the NVMe-oF Integrators List. For tests that deal with features defined as optional in the NVMe-oF specification, a check is performed at the beginning of the test to determine if the optional feature is supported or not. If the optional feature is not supported the test is marked as 'Not Applicable' and does not impact qualification for the Integrators List.

If a Test is designated as FYI, a device does not need to pass this test in order to qualify for the NVMe-oF Integrators List. Tests designated as FYI may become Mandatory tests in the future.

If a Test is designated as In Progress, a device does not need to pass this test in order to qualify for the NVMe-oF Integrators List. These test cases are still under development. Tests designated as In Progress may become FYI or Mandatory tests in the future.

Any Test may have a Case within it with a different designation as the Test itself (i.e. a Mandatory test may include FYI cases). In this case, only the Mandatory Cases are required for NVMe-oF Integrators List qualification.