

UNH-IOL NVMe TCP Transport Testing Service

**Test Plan for NVMe TCP Transport Conformance
Version 17.0**

***Target Specification: NVM Express TCP Transport
Specification version 1.0a
Technical Document***



Last Updated: January 21, 2022

***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

TABLE OF CONTENTS	2
MODIFICATION RECORD	4
ACKNOWLEDGMENTS	6
INTRODUCTION	7
REFERENCES	9
ABBREVIATIONS	10
Group 1: NVMe-oF Commands	11
Test 1.1 – Connect to Discovery Service (M)	12
Case 1: Connect to Discovery Controller (M)	12
Case 2: Connect with wrong Controller ID (FYI)	12
Case 3: Disconnect Command sent to Discovery Controller (FYI)	13
Case 4: Discovery Controller Supports Explicit Persistent Connections (FYI)	13
Case 5: Discovery Controller Does not Support Explicit Persistent Connections (FYI)	13
Test 1.2 – Connect to NVM Subsystem (M)	15
Case 1: Connect to NVM Subsystem (M)	15
Case 2: Connect with Incompatible Format (FYI)	15
Test 1.3 – Property Get Command (FYI)	17
Case 1: Get Supported Properties (FYI)	17
Case 2: Get Property Invalid Offset (M)	17
Case 3: Get Reserved Property (FYI)	18
Case 4: Fabrics Commands on Wrong Queue (FYI)	18
Case 5: SQHD in Fabrics Commands when SQ Flow Control Disabled (FYI)	18
Test 1.4 – Disconnect Command (FYI)	20
Case 1: Disconnect Submitted on I/O Queue (FYI)	20
Case 2: Disconnect Submitted on Individual I/O Queue (FYI)	20
Case 3: Disconnect Submitted on Admin Queue (FYI)	21
Case 4: Unsupported RECFMT (FYI)	21
Test 1.5 – Connection Loss (FYI)	22
Case 1: Connection Loss (FYI)	22
Group 2: Controller Architecture	23
Test 2.1 – Identify Controller Data Structure Enhancements (M)	24
Case 1: CNS=01h Identify Controller Data Structure (M)	24
Test 2.2 – Controller Association after Reset (FYI)	25
Case 1: CC.EN transitions to ‘0’, then to ‘1’ within 2 minutes due to reset (FYI)	25
Test 2.3 – Admin Command Support (FYI)	26
Case 1: All Supported Commands (FYI)	26
Test 2.4 – Authentication (FYI)	27
Case 1: Secure Channel Required but Not Established (FYI)	27
Test 2.5 – Shutdown (FYI)	28
Case 1: Normal Shutdown (FYI)	28
Case 2: Abrupt Shutdown (FYI)	28
Test 2.6 – SGLs within Command Capsule (FYI)	30
Case 1: ICDOFF is Non-Zero (FYI)	30
Case 2: SGL Data Block Exceeds MSDBD (FYI)	30

Group 3: Discovery Service	31
Test 3.1 – Initialize Discovery Controller (M)	32
Case 1: Initialize Discovery Controller (M)	32
Test 3.2 – Keep Alive Timeout (M)	34
Case 1: Keep Alive Timeout (M)	34
Case 2: Traffic Based Keep Alive (M)	34
Group 4: NVMe/TCP Binding	36
Test 4.1 – ICRReq Errors (FYI)	37
Case 1: Illegal Header in ICRReq (FYI)	37
Case 2: PFV=1 (FYI)	37
Case 3: Incorrect Initial PDU (FYI)	37
Case 4: Duplicate ICRReq (FYI)	37
Test 4.2 – NVMe Connection Errors (FYI)	39
Case 1: Incorrect Subsystem NQN (FYI)	39
Case 2: Incorrect Host NQN (FYI)	39
Case 3: Incorrect CTRL ID (FYI)	39
Case 4: Incorrect RecFMT (FYI)	39
Case 5: Out of Range Queue ID (FYI)	40
Test 4.3 – Queue Depth Exceeded (FYI)	41
Case 1: IO Queue Depth Exceeded (FYI)	41
Case 2: Queue Depth Exceeded (FYI)	41
Test 4.4 – TLS Support (FYI)	42
Case 1: TLS Supported (FYI)	42
Test 4.5 – Digest Support (FYI)	43
Case 1: PDU Header Digests Not Enabled (FYI)	43
Case 2: PDU Data Digests Not Enabled (FYI)	43
Appendix A: TEST SETUP	44
Appendix B: NOTES ON TEST PROCEDURES	45
Appendix C: TEST TOOLS	46
Appendix D: NVMe-OF INTEGRATORS LIST REQUIREMENTS	47

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.

2020 March 9 (Version 13.0) Final Release

David Woolf:

1. Tests 1.1.1, 1.2.1, 2.1.1, 3.1.1, and 3.2.1 status updated from (FYI) to (M).
2. Added Tests 4.1, 4.2, 4.3 and associated sub tests.

2020 July 21 (Version 14.0) Final Release

David Woolf:

1. Test case 2.3.1 added to address NVMe-oF v1.0 ECN 001 requirements that after the controller is enabled, it shall accept all supported Admin commands in addition to Fabrics commands
2. Test case 2.2.1 added to address NVMe-oF v1.0 ECN 002 requirements that host and controller association is preserved for at least 2 minutes after CC.EN transitions to '0'.
3. Test 2.4.1 added to address NVMe-oF v1.0 ECN 005 requirements around accepting commands on channels that are not yet authenticated/secure.
4. Test 1.3.4 added to address NVMe-oF v1.0 ECN 005 requirements around a Fabrics command that is not supported on an I/O Queue is sent on an I/O Queue, that command shall be aborted with a status code of Invalid Field in Command.
5. Test case 1.1.2 added to address NVMe-oF v1.0 ECN 005 requirements around Controller ID of FFFFh.
6. Test case 1.1.1 updated to clarify that the Controller ID in the Connect Command in this test should be set to FFFFh. Observable results are unchanged.
7. Test cases 2.5.1 and 2.5.2 added to address NVMe-oF v1.0 ECN 005 requirements around shutdown and the CC.SHN field.
8. Test Case 1.5.1 added to address TP8001 requirements that If the controller detects an NVMe Transport connection loss, then the controller shall stop processing all commands received on I/O Queue associated with that NVMe Transport connection.
9. Test Case 1.4.4 added to address TP8001 requirements around the RECFMT field.
10. Test Case 1.4.2 added to address TP8001 requirements around the CATTR field and deletion of Individual I/O Queues.
11. Test case 1.4.3 added to to address TP8001 requirements that if a Disconnect command is submitted on an Admin Queue, then the controller shall abort the command with a status of Invalid Queue Type.
12. Test cases 1.4.1 added to address nominal operation of the Disconnect Command as well as addressing TP8001 requirements that the Completion Queue entry for the Disconnect command shall be the last entry submitted to the I/O Queue Completion queue by the controller (i.e., no completion queue entries shall be submitted to the I/O Queue Completion Queue after the Completion Queue entry for the Disconnect command). The controller shall not perform command processing for any command on an I/O queue after sending the Completion Queue entry for the Disconnect command.
13. Test case 1.1.3 added to address TP8001 requirements that a Discovery controller shall not support the Disconnect command.

14. Test Cases 1.1.4 added to address checking for AER and Keep Alive support when explicit persistent connections are supported to support TP 8002 requirements.
15. Test Cases 1.1.5 added to address the error condition when persistent connections are requests and the DUT does not support persistent connections.
16. Test case 1.3.5 added to address TP8005 for SQ Flow Control, including SQ Flow Control Negotiation, and proper behavior when SQ Flow Control is disabled.
17. Support for iWARP and transport should be added to the NVMe-oF Conformance Test Suite and these products should be made eligible for the NVMe-oF Integrators List.

2021 May 3 (Version 15.0) Final Release

David Woolf:

1. Test cases 1.3.2 and 3.2.1 status updated from (FYI) to (M).

2021 September 23 (Version 16.0) Final Release

David Woolf:

1. Test case 3.2.2 updated from FYI to M.
2. Added Test 2.6.1 per requirements in NVMe-oF V1.1 ECN 002 that SGL descriptors in a command capsule that if the SGL descriptors following the Submission Queue Entry have a total size greater than (ICDOFF * 16), then the controller shall abort the command with the status code set to Invalid Number of SGL Descriptors.
3. Added Test 2.6.2 per requirements in NVMe-oF V1.1 ECN 002 that that if the host places more SGL Data Block of Keyed SGL Data Block descriptors in a capsule than the maximum indicated in the Maximum SGL Data Block Descriptors field in the Identify Controller data structure, then the controller shall abort the command with the status code set to Invalid Number of SGL Descriptors.
4. Added Test 4.4.1 per NVMe-oF v1.1 TP 8011 to check that NVMe/TCP implementations that are compliant with v1.1 version of the NVMe/TCP specification and that support TLS shall support TLS 1.3.
5. Added Test 4.5.1 per requirements in NVMe-oF V1.1 ECN 002 that if PDU Header Digest is not enabled, then all subsequent PDUs shall not contain a HDGST field and shall have the HDGSTF flag cleared to '0' in the PDU Header FLAGS field.
6. Added Test 4.5.2 per requirements in NVMe-oF V1.1 ECN 002 that if PDU Data Digest is not enabled, then the PDUs shall not contain a DDGST field and shall have the DDGSTF flag cleared to '0' in the PDU Header FLAGS field.

2022 January 21 (Version 17.0) Initial Release

Tim Sheehan:

1. This is the first release of the TCP Transport Test Suite and is based on the UNH-IOL NVMe-oF Conformance Test Plan version 16.0.

ACKNOWLEDGMENTS

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

David Woolf
Tim Sheehan

UNH InterOperability Laboratory
UNH InterOperability Lavoratory

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\TCP 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 NVM Express TCP Transport Specification version 1.0a, July 22, 2021 specification, hereafter referred to as the “NVMe TCP Specification”. Successful completion of these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many NVMe\TCP environments. Transport testing should also include those tests in the respective command set test plans with an over fabrics reference.

Products implementing the NVMe TCP specification use the TCP transport protocol. This transport may have distinct requirements described in a binding specification that is either within the NVMe TCP specification, or a separate document. The tests described in this Test Suite document are intended to check NVMe\TCP requirements. 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 NVMe 2.0 refactoring effort has made it important to reference pre-2.0 test cases that have moved to a 2.0 refactored test plan. This numbering will be realized at the end of each test case with the legacy test case number noted.

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:

Old Ref : NVMe-oF version 1.1 Specification (October 22, 2019)

1. NVM Express TCP Transport Specification 1.0a July 22, 2021
2. NVM Express Base Specification 2.0a

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

The following legacy numbering is applied to the test titles of each of the tests to map the origin of the test if it is part of the 2.0 refactoring of legacy. If there is no change to the test case number then this field will be left off of the 2.0 refactored test case title.

FC – UNH-IOL NVMe-oF Conformance Test Plan version 16.0, September 23, 2021

Legacy numbering : (oFC:x.x)

Group 1: NVMe-oF Commands

Overview:

This section describes a method for performing conformance verification for NVMe/TCP products implementing NVMe-oF Commands defined in Chapter 6 of the NVMe Express Base Specification 2.0.

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 (M)

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

References:

Old Ref : NVMe-oF Specification 3.3, 4.2, 5
NVM Express Base Specification 2.0a : 6.3, 3.1.1, 2.2.3

Resource Requirements:

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

Last Modification: April 21, 2020

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.

A Discovery Controller shall support the dynamic controller model.

The Controller ID of FFFFh shall be specified as the Controller ID in a Connect command for the Admin Queue. If the controller ID is not set to FFFFh, then a status value of Connect Invalid Parameters is returned.

A host requests an explicit persistent connection to a Discovery controller and Asynchronous Event Notifications from the Discovery controller on that persistent connection by specifying a non-zero Keep Alive Timer value in the Connect command. If the Connect command specifies a non-zero Keep Alive Timer value and the Discovery controller does not support Asynchronous Events, then the Discovery controller shall return a status value of Connect Invalid Parameters

Test Setup: See Appendix A.

Case 1: Connect to Discovery Controller (M)

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 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. 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 Controller ID of FFFFh.
5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Connect Response indicates status Success.

Case 2: Connect with wrong Controller ID (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 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. 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 Controller ID of FFFEh.
5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Connect Response indicates status Connect Invalid Parameters.

Case 3: Disconnect Command sent to 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 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. 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 Controller ID of FFFFh.
5. The Testing Station sends a Disconnect command to the Discovery Controller.

Observable Results:

1. Verify that the Disconnect command is aborted.

Case 4: Discovery Controller Supports Explicit Persistent Connections (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 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. 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 non-zero Keep Alive Timer Value to indicate that it is requesting a persistent connection.
5. The Testing Station performs a Keep Alive Command.
6. The Testing Station performs an Asynchronous Event Request Command.
7. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Connect Response indicates status Success if the DUT supports persistent connections.
2. Verify that the Keep Alive commands complete successfully.
3. Verify that the Asynchronous Event Commands complete successfully.

Case 5: Discovery Controller Does not Support Explicit Persistent Connections (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 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. 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 non-zero Keep Alive Timer Value to indicate that it is requesting a persistent connection.
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 zero Keep Alive Timer Value to indicate that it is not requesting a persistent connection.
6. The Testing Station performs a Keep Alive Command.
7. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Connect Response indicates ‘Connect Invalid Parameters’ only if the DUT does not support persistent connections.
2. Verify that the Keep Alive Command does not complete successfully.

Possible Problems: None.

Test 1.2 – Connect to NVM Subsystem (M)

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

References:

Old Ref : NVMe-oF Specification 3.3
NVM Express Base Specification 2.0a : 6.3

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe/TCP 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 (M)

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 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.
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.
6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

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 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.
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.
6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

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:

Old Ref : NVMe-oF Specification 3, 3.4, 3.5.1
NVM Express Base Specification 2.0a : 6.3, 6.4, 6.5

Resource Requirements:

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

Last Modification: April 21, 2020

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.

If a Fabric command that is not supported on an I/O Queue is sent on an I/O Queue, that command shall be aborted with a status code of Invalid Field in Command.

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 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 queues.
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
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 (M)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 queues.
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)
5. Disconnect NVMe-oF controllers from the Testing Station – acting as a Host.

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 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 queues.
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
5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Property Get commands completed with status “Success” and all values returned were 0h.

Case 4: Fabrics Commands on Wrong Queue (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 I/O Submission and Completion queues.
4. Configure the Testing Station acting as a Host to send each of the following Fabrics Commands on the I/O Queue:
 - a. Property Set
 - b. Property Get
5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that each of the Fabrics Commands were aborted with status ‘Invalid Field in Command’.

Case 5: SQHD in Fabrics Commands when SQ Flow Control Disabled (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 I/O Submission and Completion queues. In the Connect command SQ Flow Control should be disabled.
4. Configure the Testing Station acting as a Host to send each of the following Fabrics Commands on the Admin Queue:
 - a. Property Set

- b. Property Get
- 5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

- 1. Verify that each of the Fabrics Commands responses had the SQHD field set to 0.

Possible Problems: None.

Test 1.4 – Disconnect Command (FYI)

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

References:

Old Ref : NVMe-oF Specification 3.3, 4.2, 5
NVM Express Base Specification 2.0 : 6.3, 3.1.1, 2.2.3

Resource Requirements:

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

Last Modification: April 21, 2020

Discussion: If a Disconnect command is submitted on an Admin Queue, then the controller shall abort the command with a status of Invalid Queue Type.

The host indicates support for the deletion of an individual I/O Queue by setting bit 3 to ‘1’ in the CATTR field in the Connect command used to create the Admin Queue. The controller indicates support for the deletion of an individual I/O Queue by setting bit 0 to ‘1’ in the OFCS field in the Identify Controller Attributes region of the Identify Controller data structure.

If both the host and the controller support deletion of an individual I/O Queue, then the termination of an individual I/O Queue impacts only that I/O Queue (i.e., the association and all other I/O Queues and their associated NVMe Transport connections are not impacted). If either the host or the controller does not support deletion of an individual I/O Queue, then the deletion of an individual I/O Queue or the termination of an NVMe Transport connection causes the association to be terminated.

Test Setup: See Appendix A.

Case 1: Disconnect Submitted on I/O Queue (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. The Testing Station has discovered the NVM Subsystem.
4. 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 and I/O Submission and Completion Queues.
5. The Testing Station will send a Disconnect command on the I/O Queue, with RECFMT set to 0h.
6. The Testing Station will send a READ command on the I/O Queue.

Observable Results:

1. Verify that the Disconnect Command submitted on the I/O Queue completes successfully and the I/O Queue is deleted.
2. Verify that the READ command is not processed by the Controller.

Case 2: Disconnect Submitted on Individual I/O Queue (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. The Testing Station has discovered the NVM Subsystem.
4. 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

and at least 2 I/O Submission and Completion Queues. The Connect Command will also have CATTR Bit 3 set to 1 to indicate support for deletion of individual I/O Queues.

5. Check the Connect Response. If bit 0 in the OFCS field is set to 1, to indicate that the DUT supports deletion of individual queues, proceed to the next step. If Bit 0 in the OFCS field is set to 0, this test case is not applicable.
6. The Testing Station will send I/O operations on all I/O Queues.
7. The Testing Station will send a Disconnect command on a single I/O Queue.
8. The Testing Station will send I/O operations on the remaining I/O Queues.

Observable Results:

1. Verify that the Disconnect Command submitted on the I/O Queue completes successfully and the I/O Queue is deleted.
2. Verify that the I/O commands on the remaining I/O queues complete successfully.

Case 3: Disconnect Submitted on Admin Queue (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. The Testing Station has discovered the NVM Subsystem.
4. 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.
5. The Testing Station will send a Disconnect command on the Admin Queue.
6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Disconnect Command submitted on the Admin Queue is aborted with status 85h ‘Invalid Queue Type’.

Case 4: Unsupported RECFMT (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. The Testing Station has discovered the NVM Subsystem.
4. 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 and I/O Submission and Completion Queues.
5. The Testing Station will send a Disconnect command on the I/O Queue with RECFMT set to FFh.
6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Disconnect Command submitted on the Admin Queue is aborted with status 80h ‘Incompatible Format’.

Possible Problems: None.

Test 1.5 – Connection Loss (FYI)

Purpose: To verify that an NVMe-oF Controller properly handles connection loss events.

References:

Old Ref : NVMe-oF Specification 1.5.9
NVM Express Base Specification 2.0a : 3.1.1

Resource Requirements:

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

Last Modification: April 21, 2020

Discussion: If the controller detects an NVMe Transport connection loss, then the controller shall stop processing all commands received on I/O Queue associated with that NVMe Transport connection. Until the controller detects an NVMe Transport connection loss or sends a successful completion for a Disconnect command, outstanding commands may continue being processed by the controller.

Test Setup: See Appendix A.

Case 1: Connection Loss (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. The Testing Station has discovered the NVM Subsystem.
4. 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 and I/O Submission and Completion Queues.
5. The Testing Station will send a WRITE command on the I/O Queue.
6. The Testing Station will cause a temporary Connection Loss event before a response to the WRITE command is generated.
7. The Testing Station will reconnect to the DUT and perform a READ operation to the same location as the WRITE operation was attempted to.

Observable Results:

1. Verify that the WRITE command is not processed by the Controller.

Possible Problems: Ensuring that connection loss events are caused in a uniform and timely way may vary between transports.

Group 2: Controller Architecture

Overview:

This section describes a method for performing conformance verification for the Controller Architecture of NVMe/TCP products defined in Chapter 3 of the NVMe Express Base Specification 2.0.

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 (M)

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

References:

- Old Ref : NVMe Specification 4.1
- NVM Express Base Specification 2.0a : 5

Resource Requirements:

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

Last Modification: April 2, 2019

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

Test Setup: See Appendix A.

Case 1: CNS=01h Identify Controller Data Structure (M)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. Configure the Testing Station acting as a Host to perform an Identify Controller Data Structure (CNS=01h).
5. 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. 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.

Test 2.2 – Controller Association after Reset (FYI)

Purpose: To verify that host and controller association is preserved for at least 2 minutes after CC.EN transitions to ‘0’.

References:

Old Ref : NVMe-oF Specification 4.4, 4.5
NVM Express Base Specification 2.0 : 3.3.2.4, 5

Resource Requirements:

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

Last Modification: April 21, 2020

Discussion: Host to Controller associations require an Admin Queue, which can only be deleted when an association is removed. Host to Controller associations are maintained for at least 2 minutes after a reset operation. To verify this in the following test, a Admin Queue, and therefore an association are established. Then a reset operation is performed. Within 2 minutes, the testing station attempts to use the previously established Admin Queue, and expects operations to that Admin Queue to be successful, since the Host to Controller association is maintained for at least 2 minutes after shutdown.

Test Setup: See Appendix A.

Case 1: CC.EN transitions to ‘0’, then to ‘1’ within 2 minutes due to reset (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. The Testing Station acting as a host will establish an association with the DUT by connecting to the controller’s Admin Queue using the Fabrics Connect command. Within the Connect command, the host specifies the Host NQN, NVM Subsystem NQN, Host Identifier. The Host will also establish Admin and IO Submission and Completion queues.
4. The Testing Station will perform a READ operation using the established IO Submission and Completion queues and the same Host NQN and Host Identifier used in the connect command.
5. Configure the Testing Station acting as a Host to perform a Controller Level Reset.
6. The Testing Station will poll CC.EN until it is set to 0.
7. The Testing Station will poll CC.EN until it is set to 1. If this does not occur within 2 minutes this test is not applicable.
8. The Testing Station will poll CSTS.RDY until it is set to 1.
9. The Testing Station will perform a Connect Command to the same Controller using a different Host NQN and Host Identifier than what was used in the original Connect command.
10. The Testing Station will perform a Identify operation using the previously established Admin Submission and Completion queues and the same Host NQN and Host Identifier than what was used in the connect command.
11. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that the Connect Command after the reset operation failed with status 84h ‘Connect Invalid Host’.
2. Verify that the Identify operation after the Controller Level Reset was successful, and that it was able to be performed on the original Admin Queue, without establishing a new Admin Queue.

Possible Problems: None.

Test 2.3 – Admin Command Support (FYI)

Purpose: To verify that an NVMe-oF Controller can properly execute all supported Admin commands.

References:

Old Ref : NVMe-oF Specification 1.5.7, 4.3
NVM Express Base Specification 2.0a : 3.3.2.3, 5

Resource Requirements:

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

Last Modification: April 21, 2020

Discussion: When an Admin Queue is first created, the associated controller is disabled (i.e., CC.EN is initialized to '0'). A disabled controller shall abort all commands other than Fabrics commands on the Admin Queue with a status of Command Sequence Error. After the controller is enabled, it shall accept all supported Admin commands in addition to Fabrics commands.

NVMe over Fabrics does not support the Admin commands associated with I/O Queue creation and deletion (Create I/O Completion Queue, Create I/O Submission Queue, Delete I/O Completion Queue, Delete I/O Submission Queue) defined in the NVMe Base specification.

Test Setup: See Appendix A.

Case 1: All Supported Commands (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 queues.
4. Configure the Testing Station acting as a Host to send each of the following Admin Commands:
 - a. Get Log Page
 - b. Identify
 - c. Abort
 - d. Set Feature
 - e. Get Feature
 - f. Device Self Test (Short)
 - g. Device Self Test (Extended)
5. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that each of the Admin commands completed successfully.

Possible Problems: None.

Test 2.4 – Authentication (FYI)

Purpose: Verify that a controller associated with an NVM subsystem that requires a fabric secure channel shall not accept any commands (i.e., Fabrics commands, Admin commands, or I/O commands) on an NVMe Transport until a secure channel is established.

References:

Old Ref : NVMe-oF Specification 1.5.8, 4.4
NVMe Express Base Specification 2.0a : 2.2.5, 3.5.2

Resource Requirements:

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

Last Modification: April 21, 2020

Discussion: A controller associated with an NVM subsystem that requires a fabric secure channel shall not accept any commands (i.e., Fabrics commands, Admin commands, or I/O commands) on an NVMe Transport until a secure channel is established.

Test Setup: See Appendix A.

Case 1: Secure Channel Required but Not Established (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. 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.
4. Check the AUTHREQ field in the Connect Command response. If AUTHREQ=0 this test is not applicable.
5. If AUTHREQ=1, configure the Testing Station acting as a Host to send each of the following Admin Commands before Authentication can be completed:
 - a. Get Log Page
 - b. Identify
 - c. Abort
 - d. Set Feature
 - e. Get Feature
 - f. Device Self Test (Short)
 - g. Device Self Test (Extended)
6. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

Observable Results:

1. Verify that none of the Admin commands completed successfully.

Possible Problems: None.

Test 2.5 – Shutdown (FYI)

Purpose: Verify that a controller uses CC.SHN properly.

References:

Old Ref : NVMe-oF Specification 4.6
NVM Express Base Specification 2.0a : 3.6.2

Resource Requirements:

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

Last Modification: April 21, 2020

Discussion: To initiate a shutdown of a controller, the host should use the Property Set command (refer to section 3.6) to set the Shutdown Notification (CC.SHN) field.

From the time a shutdown is initiated until:

- a Controller Level Reset occurs; or
 - the controller, if dynamic, is removed from the NVM subsystem,
- the controller shall:
- process only Fabrics commands; and
 - disable the Keep Alive timer, if supported.

Test Setup: See Appendix A.

Case 1: Normal Shutdown (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 queues.
4. Configure the Testing Station acting as a Host to send a Property Set command to set CC.SHN to 01b, Normal Shutdown.
5. Send an Identify Command
6. Send a Property Get Command for any supported property.
7. 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.
2. Verify that the Identify command is not processed.
3. Verify that the Shutdown completes.

Case 2: Abrupt Shutdown (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 queues.
4. Configure the Testing Station acting as a Host to send a Property Set command to set CC.SHN to 10b, Abrupt Shutdown.
5. Send an Identify Command
6. Send a Property Get Command for any supported property.

7. 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.
2. Verify that the Identify command is not processed.
3. Verify that the Shutdown completes.

Possible Problems: None.

Test 2.6 – SGLs within Command Capsule (FYI)

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

References:

Old Ref : NVMe Specification 4.1
NVM Express Base Specification 2.0a : 5.17

Resource Requirements:

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

Last Modification: June 22, 2021

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

Test Setup: See Appendix A.

Case 1: ICDOFF is Non-Zero (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. Configure the Testing Station acting as a Host to perform an Identify Controller Data Structure (CNS=01h).
5. In the Data Structure return, read the following values:
 - a. Byte 1801:1800 In Capsule Data Offset (ICDOFF)
6. If ICDOFF is set to 0 this test is not applicable.
7. The Testing Station should issue a command capsule with SGL descriptors such that the Submission Queue Entry has a size greater than ICDOFF * 16.

Observable Results:

1. Verify that the DUT aborted the command with status code set to Invalid Number of SGL Descriptors.

Case 2: SGL Data Block Exceeds MSDBD (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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. Configure the Testing Station acting as a Host to perform an Identify Controller Data Structure (CNS=01h).
5. In the Data Structure return, read the following values:
 - a. Byte 1803 Maximum SGL Data Block Descriptors (MSDBD)
6. The Testing Station should issue a command capsule with more SGL Data Block of Keyed SGL Data Block descriptors in a capsule than the maximum indicated in the MSDBD field in the Identify Controller data structure.

Observable Results:

1. Verify that the DUT aborted the command with status code set to Invalid Number of SGL Descriptors.

Possible Problems: None.

Group 3: Discovery Service

Overview:

This section describes a method for performing conformance verification for NVMe\TCP products implementing the Discovery Service defined in Chapter 3 of the NVM Express Base Specification 2.0.

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 (M)

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

References:

Old Ref : NVMe Specification 4.1
NVM Express Base Specification 2.0a : 3.1

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe\TCP 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.
6. After initializing the Discovery controller, the host reads the Discovery Log Page.

Test Setup: See Appendix A.

Case 1: Initialize Discovery Controller (M)

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 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. 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. The Testing Station acting as a Host will use the Get Property Command to read the Controller Capabilities.
6. The Testing Station acting as a Host will use the Set Property Command to set CC.EN to 1.
7. The Testing Station acting as a Host will use the Get Property Command to read CSTS.RDY. Repeat until CSTS.RDY=1.
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)
10. Disconnect NVMe-oF controllers from the Testing Station acting as a Host.

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 (M)

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

References:

Old Ref : NVMe-oF Specification 7.1.2
NVM Express Base Specification 2.0a : 3.9

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe\TCP 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 (M)

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 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. 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.
5. The Testing Station acting as a Host will use the Get Property Command to read the Controller Capabilities.
6. The Testing Station acting as a Host will use the Set Property Command to set CC.EN to 1.
7. The Testing Station acting as a Host will use the Get Property Command to read CSTS.RDY. Repeat until CSTS.RDY=1.
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).
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.
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.
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”.

Case 2: Traffic Based Keep Alive (M)

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 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.

Group 4: NVMe/TCP Binding

Overview:

This section describes a method for performing conformance verification for NVMe/TCP products implementing the NVMe/TCP, and is only applicable to those products, defined in Chapter 3 of the NVMe Express TCP Transport Specification 1.0a.

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 4.1 – ICReq Errors (FYI)

Purpose: To verify that an NVMe Controller can properly handle errored ICReq PDUs.

References:

Old Ref : NVMe-oF Specification
NVMe Express TCP Transport Specification 1.0a : 3

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: An NVMe/TCP target should properly handle ICReq errors. This test is only applicable to products implementing the NVMe/TCP transport.

Test Setup: See Appendix A.

Case 1: Illegal Header in ICReq (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem
3. Send ICReq PDU to the NVMe-oF NVM Subsystem with illegal Header
4. After response is received, close all open TCP connections.

Observable Results:

1. Verify that the DUT sends C2HTerm in response with error status of invalid PDU Header.

Case 2: PFV=1 (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem
3. Send ICReq PDU with PFV=1 to the NVMe-oF NVM Subsystem.
4. After response is received, close all open TCP connections.

Observable Results:

1. Verify that the DUT sends C2HTerm in response with error status of invalid PDU Header.

Case 3: Incorrect Initial PDU (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem
3. Send a TCP PDU which is not an ICReq PDU to the NVMe-oF NVM Subsystem.
4. After response is received, close all open TCP connections.

Observable Results:

1. Verify that the DUT sends C2HTerm in response with error status of Sequence Error.

Case 4: Duplicate ICReq (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem
3. Send valid ICReq to the NVMe-oF NVM Subsystem with legal Header.
4. Send valid ICReq to the NVMe-oF NVM Subsystem with legal Header, identical to the ICReq sent in the previous step.
5. After response is received, close all open TCP connections.

Observable Results:

1. Verify that the first ICReq PDU completes successfully.
2. Verify that for the second ICReq, DUT sends C2HTerm in response with error status of Sequence Error.

Possible Problems: None.

Test 4.2 – NVMe Connection Errors (FYI)

Purpose: To verify that an NVMe Controller can properly handle errored NVMe Connection Requests.

References:

Old Ref : NVMe-oF Specification
NVMe Express TCP Transport Specification 1.0a : 3

Resource Requirements:

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

Last Modification: December 1, 2019

Discussion: An NVMe/TCP target should properly handle connection request errors. This test is only applicable to products implementing the NVMe/TCP transport.

Test Setup: See Appendix A.

Case 1: Incorrect Subsystem NQN (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem, with an incorrect Subsystem NQN.

Observable Results:

1. Verify that the NVMe Connect does not complete successfully.

Case 2: Incorrect Host NQN (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem, with an incorrect Host NQN.

Observable Results:

1. Verify that the NVMe Connect does not complete successfully.

Case 3: Incorrect CTRL ID (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem, with an incorrect CTRL ID.

Observable Results:

1. Verify that the NVMe Connect does not complete successfully.

Case 4: Incorrect RecFMT (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.

2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem, with an incorrect RecFMT.

Observable Results:

1. Verify that the NVMe Connect does not complete successfully.

Case 5: Out of Range Queue ID (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem, with an out of range Queue ID.

Observable Results:

1. Verify that the NVMe Connect does not complete successfully.

Possible Problems: None.

Test 4.3 – Queue Depth Exceeded (FYI)

Purpose: To verify that an NVMe Controller can properly handle Queue Depth errors.

References:

Old Ref : NVMe-oF Specification
NVMe Express TCP Transport Specification 1.0a : 3

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport. This test is only applicable to products implementing the NVMe/TCP transport.

Last Modification: December 1, 2019

Discussion: An NVMe/TCP target should properly handle Queue Depth errors.

Test Setup: See Appendix A.

Case 1: IO Queue Depth Exceeded (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem
3. Send valid ICReq to the NVMe-oF NVM Subsystem with legal Header.
4. Establish Admin Completion and Submission queues.
5. Establish IO Completion and Submission queues.
6. Send n valid READ commands to a single queue in the NVMe-oF NVM Subsystem, where n is greater than the maximum queue depth supported by the DUT. (i.e. if Max Queue Depth is 128, send 256 READ commands).
7. After response is received, close all open TCP connections.

Observable Results:

1. Verify that the first ICReq PDU completes successfully.
2. Verify that the connection remains open.

Case 2: Queue Depth Exceeded (FYI)

Test Procedure:

1. Establish a TCP Connection between the Host and NVMe-oF NVM Subsystem.
2. Perform an NVMe Connect command to connect the Host and NVMe-oF NVM Subsystem
3. Send 256 identical ICReq commands to the NVMe-oF NVM Subsystem.
4. After response is received, close all open TCP connections.

Observable Results:

1. Verify that the connection remains open.

Possible Problems: None.

Test 4.4 – TLS Support (FYI)

Purpose: To verify that an NVMe Controller supports the proper version of TLS if supported.

References:

Old Ref : NVMe-oF Specification
NVM Express TCP Transport Specification 1.0a : 3

Resource Requirements:

Tools capable of monitoring and decoding traffic on the chosen NVMe-oF transport. This test is only applicable to products implementing the NVMe/TCP transport.

Last Modification: June 22, 2021

Discussion: An NVMe/TCP target should properly handle Queue Depth errors.

Test Setup: See Appendix A.

Case 1: TLS Supported (FYI)

Test Procedure:

1. Determine the NVMe-oF or NVMe/TCP version supported by the DUT. If the NVMe-oF version is not v1.1 or the NVMe/TCP version is not v1.0, then this test is not applicable. Check the VS field of the Discovery Controller properties.
2. Configure the Testing Station to read the Discovery Log Page of the DUT.

Observable Results:

1. Verify that the Discovery Log Page returned indicates a TSAS SECTYPE of either 00 (TLS not supported) or 03 (TLS v1.3 supported).

Possible Problems: None.

Test 4.5 – Digest Support (FYI)

Purpose: To verify that an NVMe Controller can properly enable and disable Digests.

References:

Old Ref : NVMe-oF Specification 7.4.6
NVM Express TCP Transport Specification 1.0a : 3.3.1.1

Resource Requirements:

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

Last Modification: June 22, 2021

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

Test Setup: See Appendix A.

Case 1: PDU Header Digests Not Enabled (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 queues. Configure the Testing Station to issue an ICRReq PDU with HDGST_ENABLE set to 0 so that Header Digests are not used on the connection.
4. Configure the Testing Station acting as a Host to perform Write and Read operations to the DUT.

Observable Results:

1. Verify that all subsequent PDUs from the DUT did not contain a HDGST field and had the HDGSTF flag cleared to '0' in the PDU Header FLAGS field.

Case 2: PDU Data Digests Not Enabled (FYI)

Test Procedure:

1. The Testing Station and the DUT are on the same fabric network. The Testing Station knows the network address 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 queues. Configure the Testing Station to issue an ICRReq PDU with DDGST_ENABLE set to 0 so that Data Digests are not used on the connection.
4. Configure the Testing Station acting as a Host to perform Write and Read operations to the DUT.

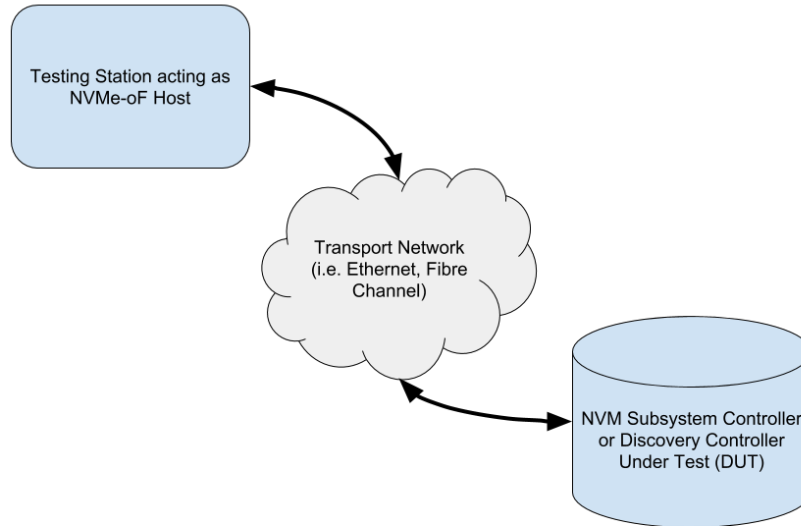
Observable Results:

1. Verify that all subsequent PDUs from the DUT did not contain a DDGST field and had the DDGSTF flag cleared to '0' in the PDU Header FLAGS field.

Possible Problems: None.

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, NVMe/TCP, and NVMe/iWARP transports.

If using the PC Edition of the IOL INTERACT NVMe-oF Test Software, UNH-IOL recommends using v16.0 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: April 21, 2020

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.