LEGAL NOTICE:

This NVM Express over Fabrics revision 1.0a technical proposal is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this NVM Express over Fabrics revision 1.0a technical proposal subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **"© 2007 - 2019 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

NVM Express Workgroup
c/o VTM, Inc.
3855 SW 153rd Drive
Beaverton, OR 97003 USA
info@nvmexpress.org

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**8001 – Graceful Disconnect**
Change Date	**10/22/19**
Builds on Specification	**NVM over Fabric 1.0a**
References Other TPs	**TP8002, TP8005**

Technical Proposal Author(s)

Name	Company
Victor Gissin	Huawei
Robert Qiuxin	Huawei
David Black	Dell EMC
Philip Kufeldt	Huawei
Fred Knight, John Meneghini	NetApp
James Smart	Broadcom

This proposal defines adding a Disconnect command to delete an NVMe I/O Queue. This command is ONLY for NVMe over Fabrics. This therefore also provides a basic mechanism for an association to survive the deletion or termination of an individual I/O Queue and the loss of the associated NVM Transport connection.

Revision History

Revision Date	Change Description
03/12/2017	Initial draft.
12/15/2017	Complete functional design
01/11/2018	Optional Fabric Command Support (OFCS) approach to indicating command support chosen – delete alternative of adding a bit to Controller Attributes.
12/10/2018	Reset baseline against NVMe-oF 1.0a
03/19/19	Add additional text for where Disconnect has impact based on David Black's input.
4/8/19	Add Disconnect Request status (for target to tell the host it should disconnect).
4/11/19	Removed disconnect request status. Removed editors notes. Ready for Phase 3.
4/23/19	Add CATTR bit to indicate host support of disconnected connections. Update disconnect model (1.5.9) to include "should" for ordering of completions prior to transport connection, and if that doesn't happen, explain what the host assumes happened. Also allow controller to disconnect individual connections if the host also supports it.
4/25/19	Add a recommendation about a delay between sending the Disconnect command response and the transport connection deletion. Disconnect may terminate outstanding commands so if the host wants commands to complete first, they must wait for those completions before sending the Disconnect command – so add such a statement.
6/10/19	Clarify some of the queue handling descriptions – particularly around queue and connection deletion corner cases.
6/13/19	Clarify that multiple I/O Queue Pairs can be associated with a single NVMe Transport connection.
6/16/19	Clarify the clarification from 6/13/19.
6/18/19	Incorporate comments from David Black.
6/27/19	Clean up the theory of operation (removing some duplicate details) and make sure there are references to the sections that already have the details.
10/16/19	Correct Integration of Figure 10 & Figure 19 (error code value and bit assignment).
10/22/2019	Ratified

Description of Specification Changes

Modify section 1.5 in NVMe over Fabric 1.0a as shown below:

1.5 Theory of Operation

…

NVMe over Fabrics has the following differences from the NVMe Base specification:

…

- NVMe over Fabrics does not use the Create I/O Completion Queue, Create I/O Submission Queue, Delete I/O Completion Queue, and Delete I/O Submission Queue commands. NVMe over Fabrics does not use the Admin Submission Queue Base Address (ASQ), Admin Completion Queue Base Address (ACQ), and Admin Queue Attributes (AQA) properties (i.e., registers in PCI Express). Queues are created using the Connect ~~Fabrics~~ command (refer to section 3.3);
- NVMe over Fabrics uses the Disconnect command (refer to section 3.TBD) to delete an I/O Submission Queue and corresponding I/O Completion Queue;
- If metadata is supported, it shall be transferred as a contiguous part of the logical block. NVMe over Fabrics does not support transferring metadata from a separate buffer;

…

1.5.2 NVM Subsystem

…

While an association exists between a host and a controller, only that host may establish connections with I/O Queues of that controller by presenting the same Host NQN, Host Identifier, NVM Subsystem NQN and Controller ID in subsequent Connect command(s) using the same NVM subsystem port, NVMe Transport type, and NVMe Transport address.

The association ~~exists until~~ between a host and controller is terminated if:

- the controller is shutdown as described in section 4.6;
- a Controller Level Reset occurs; ~~or~~
- the NVMe Transport connection is lost between the host and controller for the Admin ~~or any I/O~~ Queue; or
- an NVMe Transport connection is lost between the host and controller for any I/O Queue and the host or controller does not support individual I/O Queue deletion (refer to section 1.5.9).

There is no explicit NVMe command that breaks the NVMe Transport ~~connection~~ association between a host and controller. The Disconnect command (refer to section 3.TBD) provides a method to delete an NVMe I/O Queue (refer to section 1.5.9). While a controller is associated to a host the controller is busy and no other associations may be made to that controller.

…

Insert a new section 1.5.9 in NVMe over Fabric 1.0a as shown below:

1.5.9 I/O Queue Deletion

NVMe over Fabrics deletes an individual I/O Queue and may delete the associated NVMe Transport connection as a result of:

- the exchange of a Disconnect command and response (refer to section 3.TBD) between a host and controller; or
- the detection and processing of a transport error on an NVMe Transport connection.

The host indicates support for the deletion of an individual I/O Queue by setting bit 2 to '1' in the CATTR field in the Connect command (refer to Figure 19) 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 (refer to Figure 28).

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.

NVMe over Fabrics uses the Disconnect command to delete an Individual I/O Queue. This command is sent on the I/O Submission Queue to be deleted and affects only that I/O Submission Queue and its associated I/O Completion Queue (i.e., other I/O Queues are not affected). To delete an I/O Queue, the NVMe Transport connection for that I/O Queue is used. If all Queues associated with an NVMe transport connection are deleted, then the NVMe Transport connection may be deleted after completion of the Disconnect command. Actions necessary to delete the NVMe transport connection are transport specific. The association between the host and the controller is not affected.

If a Disconnect command returns a status other than success, the host may delete an I/O Queue using other methods including:

- waiting a vendor specific amount of time and retry the Disconnect command;
- deleting the NVMe Transport connection (note: this may impact other I/O Queues);
- performing a Controller Level Reset (note: this impacts other I/O Queues); or
- ending the host to controller association.

If the transport requires a separate NVMe Transport connection for each Admin and I/O Queue (refer to section 1.5.7), then the host should not delete an NVMe Transport connection until after:

- a Disconnect command has been submitted to the I/O Submission Queue; and
- the response for that Disconnect command has been received by the host on the corresponding I/O Completion Queue or a vendor specific timeout (refer to section 7.1.2) has occurred while waiting for that response.

If the transport requires a separate NVMe Transport connection for each Admin and I/O Queue, then the controller should not delete an NVMe Transport connection until after:

- a Disconnect command has been received on the I/O Submission Queue and processed by the controller;
- the responses for commands received by the controller on that I/O Submission Queue prior to receiving the Disconnect command have been sent to the host on the corresponding I/O Completion Queue; and
- the resulting response for that Disconnect command has been sent to the host on the corresponding I/O Completion queue (i.e., this response is the last response sent). It is recommended that the controller delay destroying the NVMe Transport connection to allow time for the Disconnect command response to be received by the host (e.g., a transport specific event occurs or a transport specific time period elapses).

If the transport utilizes the same NVMe Transport connection for all Admin and I/O Queues associated with a particular controller (refer to section 1.5.7), then the deletion of an individual I/O Queue has no impact on the NVMe Transport connection.

A Disconnect command is the last I/O Submission Queue entry processed by the controller for an I/O Queue. Controller processing of the Disconnect command completes or aborts all commands on the I/O Queue on which the Disconnect command was received. The controller determines whether to complete or abort each of those commands.

The response to the Disconnect command is the last I/O Completion Queue entry processed by the host for an I/O Queue. To avoid command aborts the host should wait for outstanding commands on an I/O Queue to complete before sending the Disconnect command.

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.

If the host detects an NVMe Transport connection loss before the responses are received for all outstanding commands submitted to the associated I/O Queue, then there is no further information available to the host about the state of those commands (e.g., each individual outstanding command may have been completed or aborted by the controller).

If an NVMe Transport connection is lost as a result of an NVMe Transport error, then before performing recovery actions related to commands sent on I/O queues associated with that NVMe Transport connection, the host should wait for at least the longer of:

- the NVMe Keep Alive timeout; or
- the underlying fabric transport timeout, if any.

Modify section 2.2.1 in NVMe over Fabric 1.0a as shown below:

2.2.1 Status Values

…

Fabrics commands use an allocation of command specific status values from 80h-BFh (refer to Figure 32 of the NVMe Base specification). Refer to Figure 10.

Figure 10: Fabrics Command Specific Status Values

Value	Description	Commands Affected
80h	~~**Connect**~~ **Incompatible Format:** The NVM subsystem does not support the ~~Connect~~ command ~~R~~record ~~F~~format specified by the host.	Connect, Disconnect
81h	~~**Connect**~~ **Controller Busy:** The controller is already associated with a host (Connect command). This value is also returned if there is no available controller (Connect command). The controller is not able to disconnect the I/O Queue at the current time (Disconnect command).	Connect, Disconnect
82h	**Connect Invalid Parameters:** One or more of the command parameters (e.g., Host NQN, Subsystem NQN, Host Identifier, Controller ID, Queue ID) specified are not valid.	Connect
83h	**Connect Restart Discovery:** The NVM subsystem requested is not available. The host should restart the discovery process.	Connect
84h	**Connect Invalid Host:** The host is not allowed to establish an association to any controller in the NVM subsystem or the host is not allowed to establish an association to the specified controller.	Connect
85h	**Invalid Queue Type:** The command was sent on the wrong queue type (e.g., a Disconnect command was sent on the Admin queue).	Disconnect
~~85h~~86h – 8Fh	Reserved	
90h	**Discover Restart:** The snapshot of the records is now invalid or out of date. The host should re-read the Discovery Log Page.	Get Log Page
91h	**Authentication Required:** NVMe in-band authentication is required and the queue has not yet been authenticated.	NOTE 1
92h – BFh	Reserved	
NOTES: 1. All commands other than Connect, Authenticate Send, and Authenticate Receive.		

Modify Figure 14 (Fabric Command Types) in Section 3 as shown below:

3 Commands

…

Figure 14: Fabric Command Types

Command Type by Field			Combined Command Type[2]	O/M[1]	I/O Queue[3]	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[4]				
0b	000 00b	00b	00h	M	No	Property Set
0b	000 00b	01b	01h	M	Yes	Connect
0b	000 01b	00b	04h	M	No	Property Get
0b	000 01b	01b	05h	O	Yes	Authentication Send
0b	000 01b	10b	06h	O	Yes	Authentication Receive
0b	TBD <000 10b>	00b	08h	O	Yes	Disconnect
	Vendor Specific					
1b	na	na	C0h – FFh	O		Vendor specific

NOTES:
1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. All Fabrics commands, other than the Disconnect command, may be submitted on the Admin Queue. The I/O Queue supports Fabrics commands specified in this column.
4. 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = reserved

Modify Figure 19 (Connect Command - Submission Queue Entry) in Section 3.3 as shown below:

3.3 Connect Command and Response

…

Figure 19: Connect Command – Submission Queue Entry

Byte	Description
…	
41:40	**Record Format (RECFMT):** Specifies the format of the Connect command capsule. ~~If a new format is defined, this value is incremented by one.~~ The format of the record specified in this definition shall be 0h. If the NVM subsystem does not support the value specified, then a status value of ~~Connect~~ Incompatible Format shall be returned.
…	

<table>
<tr><th>Byte</th><th>Description</th></tr>
<tr><td>46</td><td>Connect Attributes (CATTR): This field indicates attributes for the connection.

Bits 7:~~24~~ are reserved.

Bit 3 indicates support for deleting individual I/O Queues. If this bit is set to ‘1’, then the host supports the deletion of individual I/O Queues. If this bit is cleared to ‘0’, then the host does not support the deletion of individual I/O Queues.

Bit 2 <see TP8005>

Bits 1:0 indicate the priority class to use for commands within this Submission Queue. This field is only used when the weighted round robin with urgent priority class is the arbitration mechanism selected, the field is ignored if weighted round robin with urgent priority class is not used. Refer to section 4.11 of the NVMe Base specification. This field is only valid for I/O Queues. It shall be set to 00b for Admin Queue connections.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>Urgent</td></tr>
<tr><td>01b</td><td>High</td></tr>
<tr><td>10b</td><td>Medium</td></tr>
<tr><td>11b</td><td>Low</td></tr>
</table></td></tr>
<tr><td>…</td><td></td></tr>
</table>

Add a new section 3.TBD (alphabetically after the Connect command) as shown below:

3.TBD Disconnect Command and Response

The Disconnect command is used to delete the I/O Queue on which the command is submitted. If a Disconnect command is submitted on an Admin Queue, then the controller shall abort the command with a status of Invalid Queue Type. If the controller is not able to delete the I/O Queue, then the controller shall abort the command with a status of Controller Busy. The fields for the Submission Queue Entry are defined in Figure A.

The NVMe Transport connection is not deleted upon issuance of a Disconnect command; the host and controller may delete the NVMe Transport connection and associated resources after all NVMe Queues (I/O Queues and/or Admin Queue) associated with that NVMe Transport connection have been deleted (refer to section 1.5.9 and section 4.5).

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.

The host should not submit commands to an I/O Submission Queue after the submission of a Disconnect command to that I/O Submission Queue; submitting commands to an I/O Queue after a Disconnect command is submitted to that I/O Queue results in undefined behavior.

Figure A: Disconnect Command – Submission Queue Entry

Byte	Description
00	**Opcode (OPC):** Set to 7Fh to indicate a Fabrics command.
01	Reserved
03:02	**Command Identifier (CID):** This field specifies a unique identifier for the command. Refer to the definition in Figure 7.
04	**Fabrics Command Type (FCTYPE):** Set to 08h to indicate a Disconnect command.
23:05	Reserved
39:24	**SGL Descriptor 1 (SGL1):** This field is reserved, as there is no data transferred by this command.
41:40	**Record Format (RECFMT):** Specifies the format of the Disconnect command capsule. The format of the record specified in this definition shall be 0h. If the NVM subsystem does not support the value specified, then a status value of Incompatible Format shall be returned.
42:63	Reserved

The Disconnect response provides status for the Disconnect command. The Disconnect response is defined in Figure B.

Figure B: Disconnect Response

Byte	Description
07:00	Reserved
09:08	**SQ Head Pointer (SQHD):** Indicates the current Submission Queue Head pointer for the associated Submission Queue.
11:10	Reserved
13:12	**Command Identifier (CID):** Indicates the identifier of the command that is being completed.
15:14	**Status (STS):** Specifies status for the command. Refer to section 2.2.1 for values specific to the Disconnect command.

Modify Figure 28 (Identify Controller Attributes) in Section 4.1 as shown below:

4.1 Identify Controller Data Structure Enhancements

…

Figure 28: Identify Controller Attributes

1795:1792	M	**I/O Queue Command Capsule Supported Size (IOCCSZ):** This field defines the I/O command capsule size in 16 byte units. The minimum value that may be specified is 4 corresponding to 64 bytes.
1799:1796	M	**I/O Queue Response Capsule Supported Size (IORCSZ):** This field defines the I/O response capsule size in 16 byte units. The minimum value that may be specified is 1 corresponding to 16 bytes.
1801:1800	M	**In Capsule Data Offset (ICDOFF):** This field defines the offset where data starts within a capsule. This value is applicable to I/O Queues only (the Admin Queue shall use a value of 0h). The value is specified in 16 byte units. The offset is from the end of the Submission Queue Entry within the command capsule (starting at 64 bytes in the command capsule). The minimum value is 0 and the maximum value is FFFFh.
1802	M	**Controller Attributes (CTRATTR):** This field indicates attributes of the controller. Bits 7:1 are reserved. Bit 0 if cleared to '0' then the NVM subsystem uses a dynamic controller model. Bit 0 if set to '1' then the NVM subsystem uses a static controller model.
1803	M	**Maximum SGL Data Block Descriptors (MSDBD):** This field indicates the maximum number of (Keyed) SGL Data Block descriptors that a host is allowed to place in a capsule. A value of 0h indicates no limit.
TBD <1805:1804>	M	Optional Fabric Commands Support (OFCS): Indicate whether the controller supports optional fabric commands. Bits 15:1 are reserved. Bit 0 if cleared to '0' then the controller does not support the Disconnect command. Bit 0 if set to '1' then the controller supports the Disconnect command and deletion of individual I/O Queues.
2047:1806	M	Reserved

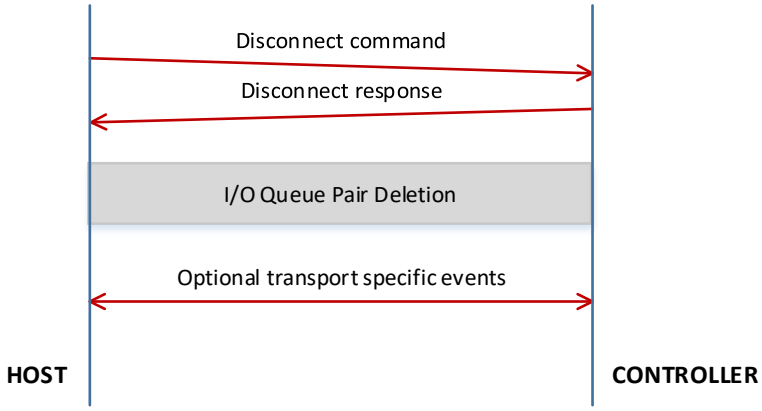
Add a new section 4.5 (renumber the current 4.5 (Shutdown) section to 4.6) as shown below:

4.5 I/O Queue Deletion

Individual I/O Queues may be deleted by the host. Admin Queue deletion is not possible. Deletion of the Admin Queue requires termination of the entire association between the host and controller.

The host selects the I/O Queue to delete. The Disconnect command is used to request deletion of both the I/O Submission Queue and the I/O Completion Queue. Outstanding commands on the specified I/O Queue are completed or aborted by the controller before the Disconnect command is completed by the controller. The successful completion of the Disconnect command indicates an implicit completion status of Command Aborted due to SQ Deletion for any outstanding commands on that I/O Queue for which a completion queue entry has not been returned by the controller. Upon successful completion of the Disconnect command, the host and controller may release their respective transport resources and their queue resources that are dedicated to that I/O Queue. The host and controller may also use timers to determine when transport resources should be released or may retrain transport resources (e.g., an NVMe Transport connection) for reuse. Figure C is a ladder diagram that describes the queue deletion process for an I/O Queue where the transport resources dedicated to that queue, including an NVMe Transport connection, are released after the I/O Queue is deleted.

Figure C: I/O Queue Deletion Flow



Sequence of events:

1) The host submits a Disconnect command to the I/O Queue that is to be deleted.
2) The controller processes the Disconnect command. As part of this processing, the controller completes or aborts all commands on this I/O Queue other than the Disconnect command.
3) The controller sends the Disconnect command response (i.e., completion) to the host.
4) Upon receipt of the Disconnect command completion, the host deletes its NVMe resources for the I/O Queue.
5) The host may send an optional transport-specific event to the controller to indicate that the host has processed the Disconnect command completion.
6) The controller deletes its NVMe resources for this I/O Queue.
7) The host removes the NVMe Transport connection and releases associated transport resources if the NVMe Transport connection is not associated with any other NVMe Queues.
8) The controller removes the NVMe Transport connection and releases associated transport resources if the NVMe Transport connection is not associated with any other NVMe Queues.

4.6 Shutdown

…

Modify section 5 in NVMe over Fabric 1.0a as shown below:

5 Discovery Service

…

Discovery controllers that do not support explicit persistent connections shall not support Keep Alive commands and may use a fixed Discovery controller activity timeout value (e.g., 2 minutes). If no commands are received by such a Discovery controller within that time period, the controller may perform the actions for Keep Alive Timer expiration defined in section 7.1.2.

A Discovery controller shall not support the Disconnect command.

A Discovery Log Page with multiple entries for the same NVM subsystem indicates that there are multiple fabric paths to the NVM subsystem, and/or that multiple static controllers may share a fabric path. The host may use this information to form multiple associations to controllers within an NVM subsystem.

…

Modify section 7 in NVMe over Fabric 1.0a as shown below:

7 Transport Definition

7.1 Transport Requirements

This section defines requirements that all NVMe Transports that support an NVMe over Fabrics implementation shall meet.

The NVMe Transport may support NVMe Transport ~~level~~ error detection and report errors to the NVMe layer in command status values. The controller may record NVMe Transport specific errors in the Error Information Log. Transport errors that cause loss of a message or loss of data in a way that the low-level NVMe Transport cannot replay or recover should cause:

- the deletion of the individual I/O Queues (refer to section 4.5) and the associated NVMe Transport connection on which that NVMe Transport level error occurred; or
- termination of the NVMe Transport connection and ending of the association between the host and controller.

The NVMe Transport shall provide reliable delivery of capsules between a host and NVM subsystem (and allocated controller) over each connection. The NVMe Transport may deliver command capsules in any order on each queue except for I/O commands that are part of fused operations (refer to section 4.10 of the NVMe Base specification).

For command capsules that are part of fused operations for I/O commands, the NVMe Transport:

a) shall deliver the first and second command capsules for each fused operation to the queue in-order; and
b) shall not deliver any other command capsule for the same Submission Queue between delivery of the two command capsules for a fused operation.

The NVMe Transport shall provide reliable delivery of response capsules from an NVMe subsystem to a host over each connection. The NVMe Transport shall deliver response capsules that include an SQ Head Pointer (SQHD) value to the host in-order; this includes all Connect response capsules and all Disconnect response capsules.

7.1.1 Submission Queue Head Pointer Update Optimization

This optimization does not apply to queue pairs for which Submission Queue (SQ) flow control is disabled, as the SQHD field is reserved if SQ flow control is disabled, refer to section 2.4 and to section 3.3.

The NVMe Transport may omit transmission of the SQHD value for a response capsule that:

a) contains a Generic Command status (i.e., Status Code Type 0h) indicating successful completion of a command (i.e., Status Code 00h); and
b) is not a Connect response capsule; and
c) is not a Disconnect response capsule.

If a new SQHD value is not received in a response capsule, the host continues to use its previous SQHD value. Thus, at the NVMe layer there is a logical progression of SQHD values despite the fact that the NVMe Transport may not actually transfer the SQHD value in each response capsule.

The NVMe Transport may deliver response capsules that do not contain an SQHD value to the host in any order. The applicable NVMe Transport binding specification defines how presence versus absence of an SQHD value in a response capsule is indicated by the NVMe Transport.

Periodic SQHD updates at the host are required to avoid Submission Queue (SQ) starvation as SQHD value transmission in responses is the only means of releasing SQ slots for host reuse.

An NVMe Transport may transmit an SQHD value in every response capsule. If an NVMe Transport does not transmit an SQHD value in every response capsule, then an SQHD value should be transmitted periodically (e.g., in at least one of every n response capsules on a CQ, where n is 10% of the size of the associated SQ) or more often. An SQHD value should always be transmitted if 90% or more of the slots in the associated SQ are occupied at the subsystem.
…