LEGAL NOTICE:

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this erratum, 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: **“© 2008 to 2024 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.

The NVM Express® design mark is a registered trademark of NVM Express, Inc.
PCI-SIG®, PCI Express®, and PCIe® are registered trademarks of PCI-SIG.
InfiniBand™ is a trademark and servicemark of the InfiniBand Trade Association.

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

NVM Express® Technical Errata

Errata ID	**123**
Revision Date	**11/21/2024**
Affected Spec Ver.	**NVM Express® Base Specification Revision 2.1** **NVM Express Boot Specification 1.1**
Corrected Spec Ver.	

Errata Author(s)

Name	**Company**
Mike Allison, Judy Brock, Bill Martin	Samsung
Dan Hubbard	Micron
David Black, Doug Farley	Dell
Paul Suhler, Fred Knight	Kioxia
Phil Cayton	Intel
Martin Wilck	SUSE
John Meneghini	Red Hat
Luis Freeman	IBM

Errata Overview

This ECN updates and clarifies various text within the NVM Express Base Specification Revision 2.1 and the NVM Express Boot Specification 1.1.

Revision History

Revision Date	Change Description
7/31/2024	Initial creation with the solutions for: • Bug 400 Elasticity Buffer Size field name confusion
8/14/2024	Bug solution for: • Bug 416 Revision TBD in the Notes of Figure 310 Elasticity Buffer Size field name confusion
8/21/2024	Bug solution for: • Bug 391 Memory Buffer or Data Buffer
8/23/2024	Moved bug 391 solution to ECN124.
9/4/2024	Bug solutions for: • Bug 420 Memory Buffer or Data Buffer • Bug 423 Need clarify on aborting Directive commands specifying Data Placement Directive • Bug 428 Enabling Flexible Data Placement Flow Correction • Bug 155 Security Send/Receive and reservation conflicts • Bug 331 -Missing steps in Boot Partition read
9/11/2024	Bug solutions for: • Bug 417 TP 8017a Integration - wrong location of new section • Bug 430 Prohibit enabling Data Placement Directive if FDP feature is not Enabled • Bug 429 I/O Management Receive Clarification
9/19/2024	Cleaned up if ready for a review.
11/20/2024	Integrated
11/21/2024	Corrected an authors name.

Description of Changes

NVM Express Base Specification Revision 2.1:

Editorial Changes:

- Clarified computation of data transfer times for the CMB elasticity buffer and PMR elasticity buffer.
- Resolved a Revision TBD in Figure 310 from TP4155.
- Clarified that if the Directive Type (DTYPE) field in Command Dword 11 is set to the Data Placement Directive in a Directive Send command or a Directive Receive command, then the command is aborted.
- Corrected the order of issuing a GET FEATURES/SET FEATURES command to the FDP Events Feature before a Namespace is created, but the FDP Events Features requires the NSID is provided to specify a valid Placement Identifier.
- Corrected how Reservations affect Security Send and Security Receive commands as Reservations only affect Security Send and Security Receive commands that use the namespace identifier.
- Revised text that incorrectly stated submitting commands to namespaces as commands are always submitted to controllers.
- Add missing interlock step for reading a Boot Partition as a host needs to observe Boot Partition read in progress before looking for completion.
- Corrected the integration mistake for TP 8017a - In-band Authentication of Discovery subsystems by moving intended content of section titled “Special considerations for In-band Authentication of Discovery subsystems” to intended location at end of section titled “NVMe In-band Authentication”, restoring general applicability of specification of reauthentication and authentication timeout to all NVMe over Fabrics implementations.
- Clarified existing text “If the specified namespace is not contained in an Endurance Group with Flexible Data Placement enabled, then the controller shall abort the Directive Send command with a status code of FDP Disabled” – this text was intended to convey that the FDP Directive Type cannot be enabled for a NS if the FDP Feature itself is not enabled for that NS.

NVM Express Boot Specification 1.1:

Editorial Changes:

- Clarified that the SMBIOS UUIDs expressed into the NBFT need to comply with the System Management BIOS (SMBIOS) Reference Specification (DSP0134), which in turn makes it clear to use the case and RFC 4122 for string formatting. This change makes it clear in the Boot Specification that those strings need to be lowercase in format.

Note:

BLACK text indicates unchanged text. **BLUE** text indicates newly inserted text. **~~RED stricken~~** text indicates deleted text; **ORANGE** text indicates changes from another ECN. **Purple** text indicates destination of moved text without changes. **~~Purple stricken~~** text indicates source of moved text without changes. **GREEN** text indicates editor notes.

Description of NVM Express Base Specification 2.1 changes

Modify section 7 as shown below:

7 I/O Commands

…

7.3 I/O Management Receive command

…

If the Number of Dwords (NUMD) field corresponds to a length that is less than the size of the data structure to be returned, then only that specified portion of the data structure is transferred. If the NUMD field corresponds to a length that is greater than the size of the associated data structure, then the entire contents of the data structure are transferred and no additional data is transferred, unless otherwise specified.

…

Figure 562: I/O Management Receive – Command Dword 11

Bits	Description
31:00	**Number of Dwords (NUMD):** This field specifies the number of dwords to transfer. This is a 0's based value.

7.3.1 I/O Management Receive Operations

7.3.1.1. Reclaim Unit Handle Status (Management Operation 01h)

…

~~If the host reads beyond the size of the Reclaim Unit Handle Status data structure (refer to Figure 563), zeroes are returned.~~ If the Number of Dwords (NUMD) field corresponds to a length that is greater than the size of the Reclaim Unit Handle Status data structure (refer to Figure 563), then the entire contents of the data structure are transferred and zeroes are transferred beyond the end of that data structure.

Figure 563: Reclaim Unit Handle Status

Bytes	Description
Header	
13:00	Reserved
15:14	**Number of Reclaim Unit Handle Status Descriptors (NRUHSD):** This field indicates the number of Reclaim Unit Handle Status Descriptors in the Reclaim Unit Handle Status Descriptor list.
Reclaim Unit Handle Status Descriptor List	
47:16	**Reclaim Unit Handle Status Descriptor 1:** The first Reclaim Unit Handle Status Descriptor (refer to the applicable I/O Command Set specification).
79:48	**Reclaim Unit Handle Status Descriptor 2:** The second Reclaim Unit Handle Status Descriptor (refer to the applicable I/O Command Set specification), if any.
…	…
(NRUHSD *32)+15: (NRUHSD *32)–16	**Reclaim Unit Handle Status Descriptor NRUHSD:** The last Reclaim Unit Handle Status Descriptor (refer to the applicable I/O Command Set specification), if any.

…

Modify section 8 as shown below:

8 Extended Capabilities

…

8.1 Common Extended Capabilities

…

8.1.3 Boot Partitions

…

8.1.3.1 Reading from a Boot Partition

…

To read data from a Boot Partition, the host follows these steps:

1. Initialize the transport (e.g., PCIe link), if necessary;
2. Determine if Boot Partitions are supported by the controller (CAP.BPS);
3. Determine which Boot Partition is active (BPINFO.ABPID) and the size of the Boot Partition (BPINFO.BPSZ);
4. Allocate a physically contiguous memory buffer in the host to store the contents of a Boot Partition;
5. Initialize the address (BPMBL.BMBBA) into the memory buffer where the contents should be copied;
6. If no Boot Partition read is in progress (i.e., the Boot Read Status (BPINFO.BRS) field is not set to 01b), then initiate ~~Initiate~~ the transfer of data from a Boot Partition by writing to the Boot Partition Read Select (BPRSEL) property. This includes setting the Boot Partition ~~I~~identifier (BPRSEL.BPID) field, ~~size of~~ the Boot Partition Read Size (BPRSEL.BPRSZ) field, and the Boot Partition Read Offset (BPRSEL.BPROF) field;~~.~~
7. Wait for the ~~The~~ controller to set~~s~~ the ~~Boot Read Status (~~BPINFO.BRS~~)~~ field to 01b while the controller is transferring the Boot Partition contents to indicate that a Boot Partition read operation is in progress; and

~~7~~8. Wait for the controller to completely transfer the requested portion of the Boot Partition, indicated in the status field (BPINFO.BRS). If BPINFO.BRS is set to 10b, the requested Boot Partition data has been transferred to the Boot Partition Memory Buffer. If BPINFO.BRS is set to 11b, there was an error transferring the requested Boot Partition data and the host may request the Boot Partition data again.

…

8.1.8 Directives

…

8.1.8.4 Data Placement (Directive Type 02h, Optional)

The Data Placement Directive enables the host to specify to the controller the Reclaim Unit (refer section 8.1.10) to place the user data in I/O commands specified by the appropriate I/O Command Set specification.

The Data Placement Directive has no Directive Operations defined. Any Directive Receive command or Directive Send command that specifies ~~a~~ the Data Placement Directive ~~Type~~ in the Directive Type (DTYPE) field in Command Dword 11 shall be aborted by the controller with a status code of Invalid Field in Command.

If a Directive Send command to enable the Data Placement Directive is processed (refer to section 8.1.8.2.2.1) and the specified namespace is not contained in an Endurance Group with Flexible Data Placement enabled, then the controller shall abort the Directive Send command with a status code of FDP Disabled.

…

8.1.10 Flexible Data Placement

…

8.1.10.2 Enabling Flexible Data Placement (Informative)

The host prepares an Endurance Group for operation in Flexible Data Placement using the following process:

1) Validate that the Flexible Data Placement capability is supported by issuing an Identify command to access the Identify Controller data structure and checking that the Flexible Data Placement Support (FDPS) bit is set to ‘1’.
2) Delete any existing namespaces that exist in the Endurance Group where Flexible Data Placement is to be enabled.
3) Issue a Get Log Page command specifying the FDP Configurations log page (refer to section 5.1.12.1.28). Parse the FDP Configurations List in the returned log page to determine the desired FDP configuration.
4) Enable Flexible Data Placement utilizing that desired FDP configuration by issuing a Set Features command specifying:

   a. the Flexible Data Placement feature;
   b. the Endurance Group Identifier field set to the ENDGID of the Endurance Group in which Flexible Data Placement is to be enabled;
   c. the FDPE bit set to ‘1’ (i.e., enabling Flexible Data Placement); and
   d. the Flexible Data Placement Configuration Index field set to the index of the desired FDP configuration from the FDP Configurations List in the FDP Configurations log page.

~~5) Issue Get Features commands for the FDP Events feature (refer to section 5.1.25.1.21) to acquire the list of supported FDP events and the enabled state of each supported FDP event.~~
~~6) Issue Set Features commands for the FDP Events feature (refer to section 5.1.25.1.21) to specify if FDP events are required for Reclaim Unit Handles.~~

5) Issue an Identify command specifying the Identify Namespace data structure (i.e., CNS 00h as defined in Figure 273) to identify which LBA formats are supported.
6) For each namespace to be created in the Endurance Group, issue a Namespace Management command specifying:

   a. the Select field set to the Create operation (refer to the NVM Express Base Specification);
   b. the User Data Format; and
   c. the Placement Handle List used to define the Reclaim Unit Handle associated with each Placement Handle of the namespace (refer to the Namespace Management section of the applicable I/O Command Set specification).

7) Issue Get Features commands for the FDP Events feature (refer to section 5.1.25.1.21) to acquire the list of supported FDP events and the enabled state of each supported FDP event.
8) Issue Set Features commands for the FDP Events feature (refer to section 5.1.25.1.21) to specify if FDP events are required for Reclaim Unit Handles.
9) For each namespace created in an Endurance Group in which the Data Placement Directive is to be utilized by the host by write commands specifying that namespace:

   a. Issue a Directive Send command specifying:

      i. the Directive Operation field set to Enable Directive (i.e., 01h);
      ii. the NSID of that namespace;
      iii. the DTYPE field in Command Dword 11 set to Identify (i.e., 00h);
      iv. the DTYPE field in Command Dword 12 set to Data Placement (i.e., 02h); and
      v. the Enable Directive bit set to ‘1’ (i.e., to enable the Data Placement Directive).

   b. Issue an Identify command with CNS value 08h (i.e., the I/O Command Set Independent Identify Namespace data structure) to determine if a volatile write cache is present for the namespace.

…

8.1.22 Reservations

…

A reservation on a namespace restricts hosts access to that namespace. If a host submits a command that uses the Namespace Identifier (NSID) field and that NSID field contains an active NSID, then ~~to a namespace~~ in the presence of a reservation on the namespace identified by that NSID ~~and~~ where that host lacks sufficient rights, ~~then the~~ that command is aborted by the controller with a status code of Reservation Conflict. If a host submits a command that uses the NSID field ~~with the~~ and that NSID field is set to FFFFFFFFh, then in the presence of a reservation on any of the namespaces impacted by that command ~~and~~ where that host lacks sufficient rights on any of ~~all~~ the impacted namespaces, ~~then the~~ that command is aborted by the controller with a status code of Reservation Conflict. Capabilities are provided that allow recovery from a reservation on a namespace held by a failing or uncooperative host.

…

If an NVM subsystem~~:~~

~~1.~~ supports reservations with a Host Identifier value of 0h, and:~~;~~
~~2.~~1. registrations or reservations are established by a host with a Host Identifier value of 0h; and
~~3.~~ 2. the Host Identifier of that host is changed to a non-zero value,

then those registrations or reservations remain associated with the Host with a Host Identifier value of 0h and are not associated with the host with the non-zero Host Identifier.

…

Support for reservations by a namespace or controller is optional. A namespace indicates support for reservations by reporting a non-zero value in the Reservation Capabilities (RESCAP) field in the Identify Namespace data structure. A controller indicates support for reservations through the Optional NVM Command Support (ONCS) field in the Identify Controller data structure (refer to Figure 312). If a host submits a command associated with reservations (i.e., Reservation Report, Reservation Register, Reservation Acquire, and Reservation Release):

- to a controller ~~or a namespace~~ that does not ~~both~~ support reservations;~~,~~ or
- that impacts a namespace that does not support reservations,

then the command is aborted by the controller with a status code of Invalid Command Opcode.

…

8.1.22.1 Reservation Types

…

Figure 646: Command Behavior in the Presence of a Reservation

NVMe Command	Write Exclusive Reservation		Exclusive Access Reservation		Write Exclusive Registrants Only or Write Exclusive All Registrants Reservation		Exclusive Access Registrants Only or Exclusive Access All Registrants Reservation	
	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant
Read Command Group								
Security Receive (Admin)[4] I/O Command Set specific Copy Commands (source)[2,3] I/O Command Set specific Read Commands[2]	A	A	C	C	A	A	C	A
Write Command Group								
Capacity Management (Admin) Flush Format NVM (Admin) Namespace Attachment (Admin) Namespace Management (Admin) Sanitize (Admin) Security Send (Admin)[4] I/O Command Set specific Copy Commands (destination)[2,3] I/O Command Set specific Write Commands[2]	C	C	C	C	C	A	C	A
Reservation Command Groups								
Reservation Acquire - Acquire	C	C	C	C	C	C	C	C
Reservation Acquire - Preempt Reservation Acquire - Preempt and Abort Reservation Release	C	A	C	A	C	A	C	A
All Other Commands Group								
All other commands[1]	A	A	A	A	A	A	A	A

Key:

A definition: A=Allowed, command processed normally by the controller
C definition: C=Conflict, command aborted by the controller with a status code of Reservation Conflict

Notes:

1. The behavior of a vendor specific command is vendor specific.
2. Refer to the applicable I/O Command Set specification
3. For an I/O Command Set specific Copy command, each source namespace is checked for reservation conflict as if accessed by a read command and the destination namespace is checked for reservation conflict as if accessed by a write command, as described in the applicable I/O command Set specification.
4. Applies only to commands that use the Namespace Identifier (NSID) field (refer to the applicable row in Figure 141). Any command that does not use the NSID field is part of the All Other Commands Group.

…

8.2 Memory-Based Transport Extended Capabilities (PCIe)

This section describes extended capabilities that are specific to the Memory-based transport model.

8.2.1 Controller Memory Buffer

…

The time required to transfer data from the write elasticity buffer to the CMB is the amount of data written to the elasticity buffer divided by the ~~Controller Memory Buffer Sustained Write Throughput~~ maximum CMB sustained write throughput (refer to section 3.1.4.19). The time to transfer the entire contents of the write elasticity buffer is the ~~Controller Memory Buffer Elasticity Buffer Size~~ size of the CMB elasticity buffer (refer to section 3.1.4.18) divided by the ~~Controller Memory Buffer Sustained Write Throughput~~ maximum CMB sustained write throughput. The host is required to account for any units differences in the CMB Elasticity Buffer Size Units field and the CMB Sustained Write Throughput Units field.

…

8.2.4 Persistent Memory Region

…

The time required to transfer data from the write elasticity buffer to non-volatile media is the amount of data written to the elasticity buffer divided by the ~~Persistent Memory Region Sustained Write Throughput~~ maximum PMR sustained write throughput (refer to section 3.1.4.26). The time to transfer the entire contents of the write elasticity buffer is the ~~Persistent Memory Region Elasticity Buffer Size~~ size of the PMR elasticity buffer (refer to section 3.1.4.25) divided by the ~~Persistent Memory Region Sustained Write Throughput~~ maximum PMR sustained write throughput. The host is required to account for any units differences in the PMR Elasticity Buffer Size Units field and the PMR Sustained Write Throughput Units field.

…

8.3 Message-Based Transport Extended Capabilities (Fabrics)

…

8.3.4 NVMe over Fabrics Secure Channel and In-band Authentication

…

8.3.4.2 NVMe In-band Authentication

The Authentication and Security Requirements (AUTHREQ) field in the Connect response capsule (refer to Figure 547) indicates whether NVMe in-band authentication is required.

If one or more of the bits in the AUTHREQ field are set to ‘1’, then the controller requires that the host authenticate on that queue in order to proceed with Fabrics, Admin, and I/O commands. Authentication success is defined by the specific security protocol that is used for authentication. If any command other than Connect, Authentication Send, or Authentication Receive is received prior to authentication success, then the controller shall abort the command with Authentication Required status.

If all bits in the AUTHREQ field are cleared to ‘0’, then the controller does not require the host to authenticate, and the NVM subsystem shall not abort any command with a status code value of Authentication Required.

Refer to section 8.3.4.2.1 for considerations on Discovery subsystems.

~~8.3.4.2.1 Special considerations for In-band Authentication of Discovery subsystems~~

~~Hosts that have been configured to authenticate Discovery subsystems with an in-band authentication protocol that supports both unidirectional authentication and bidirectional authentication (e.g., DH-HMAC-CHAP, refer to section 8.3.4.5) should behave as follows:~~

- ~~If the host connected to a Discovery subsystem using the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery) and the Discovery subsystem did not request authentication, then the host should not perform an authentication transaction;~~

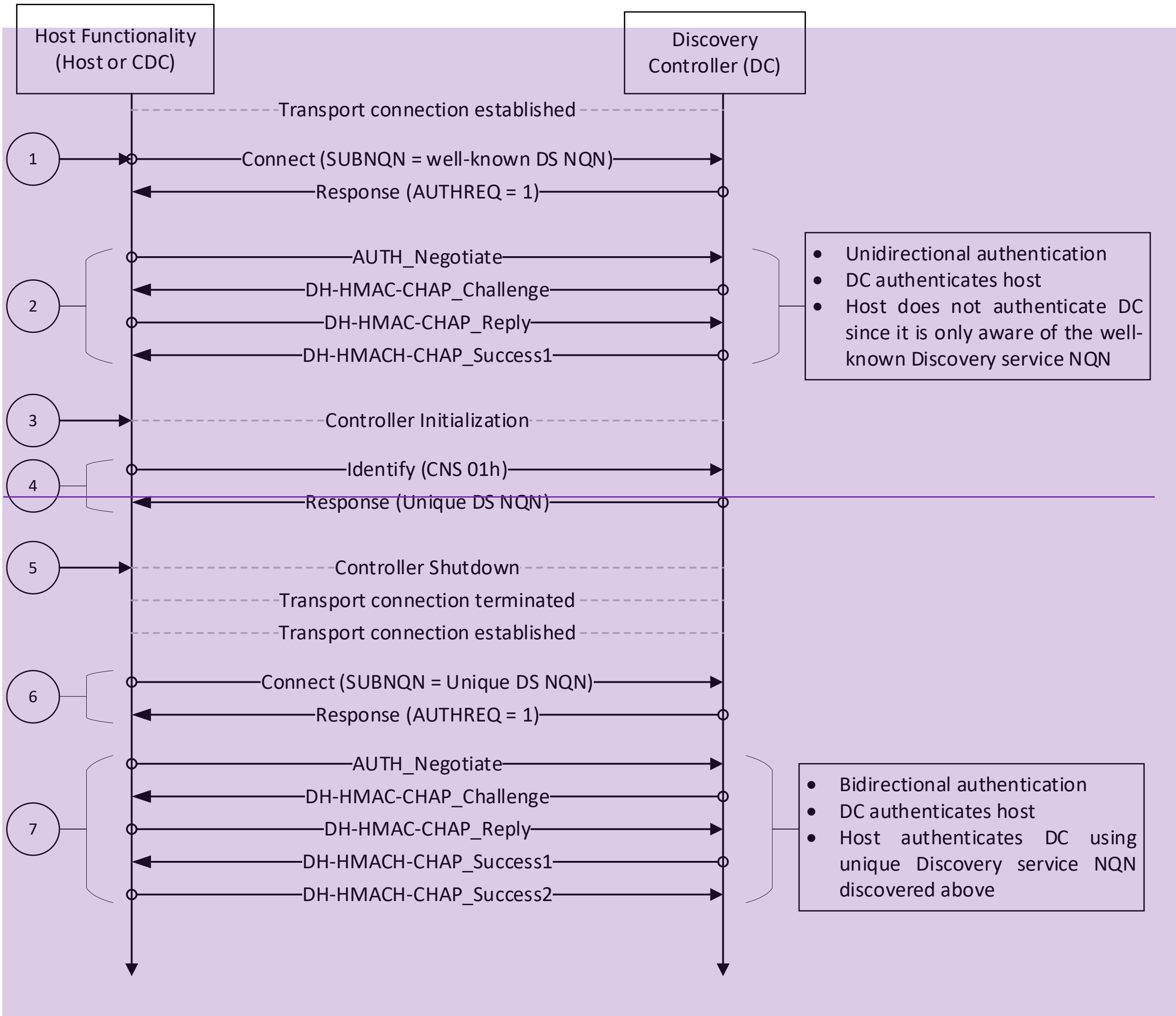
Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008-2024 NVM Express, Inc.

- ~~If the host connected to a Discovery subsystem using the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery) and the Discovery subsystem requested authentication, then the host should perform only unidirectional authentication (i.e., the Discovery subsystem may authenticate the host, but the host should not authenticate the well-known Discovery Service NQN); or~~
- ~~If the host connected to a Discovery subsystem using the unique Discovery Service NQN for that Discovery subsystem (refer to section 3.1.3.3), regardless of whether the Discovery subsystem requested authentication, then the host may perform unidirectional authentication or bidirectional authentication (i.e., the host may authenticate the unique Discovery Service NQN for that Discovery subsystem).~~

~~Figure 726 illustrates a process that a host is able to use to retrieve the unique Discovery Service NQN and perform in-band authentication for a Discovery subsystem if that host has been configured to authenticate Discovery subsystems and the Discovery subsystem that the host connects to also requires authentication (i.e., the AUTHREQ field is not cleared to zero) using DH-HMAC-CHAP for authentication. This process includes:~~

1. ~~connect to the Discovery subsystem using the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery);~~
2. ~~perform unidirectional authentication with the Discovery subsystem;~~
3. ~~perform controller initialization (refer to section 3.5);~~
4. ~~retrieve the unique Discovery Service NQN (refer to section 3.1.3.3);~~
5. ~~perform controller shutdown (refer to section 3.6);~~
6. ~~reconnect to the Discovery subsystem using the unique Discovery Service NQN for that Discovery subsystem; and~~
7. ~~perform bidirectional authentication with the Discovery subsystem using the unique Discovery Service NQN for that Discovery subsystem.~~

~~Figure 726: Unique Discovery Service NQN retrieval for bidirectional authentication~~


Host Functionality (Host or CDC)
Discovery Controller (DC)
Transport connection established
1
Connect (SUBNQN = well-known DS NQN)
Response (AUTHREQ = 1)
2
AUTH_Negotiate
DH-HMAC-CHAP_Challenge
DH-HMAC-CHAP_Reply
DH-HMACH-CHAP_Success1
Unidirectional authentication
DC authenticates host
Host does not authenticate DC since it is only aware of the well-known Discovery service NQN
3
Controller Initialization
4
Identify (CNS 01h)
Response (Unique DS NQN)
5
Controller Shutdown
Transport connection terminated
Transport connection established
6
Connect (SUBNQN = Unique DS NQN)
Response (AUTHREQ = 1)
7
AUTH_Negotiate
DH-HMAC-CHAP_Challenge
DH-HMAC-CHAP_Reply
DH-HMACH-CHAP_Success1
DH-HMACH-CHAP_Success2
Bidirectional authentication
DC authenticates host
Host authenticates DC using unique Discovery service NQN discovered above

The host may initiate a subsequent authentication transaction at any time for reauthentication purposes. Initiating reauthentication shall not invalidate a prior authentication. If the reauthentication transaction concludes with the controller sending an AUTH_Failure1 message (refer to section 8.3.4.4.2), then the controller shall terminate all commands with a status code of Operation Denied and disconnect the NVMe over Fabrics connection. If the reauthentication transaction concludes with the host sending an AUTH_Failure2 message, then the host shall disconnect the NVMe over Fabrics connection.

The state of an in-progress authentication transaction is soft-state. If the subsequent command in an authentication transaction is not received by the controller within a timeout equal to:

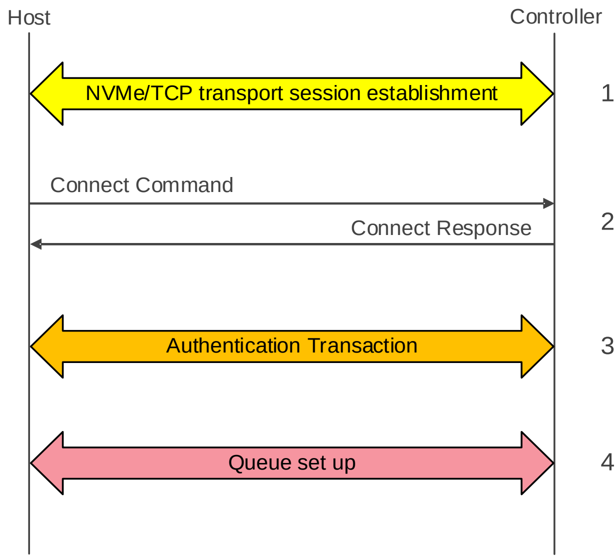
- the Keep Alive Timeout value (refer to Figure 545), if the Keep Alive Timer is enabled; or
- the default Keep Alive Timeout value (i.e., two minutes), if the Keep Alive Timer is disabled;

then the authentication transaction has timed out and the controller should discard the authentication transaction state (including the T_ID value, refer to section 8.3.4.4.1).

For an initial authentication, an authentication transaction timeout should be treated as an authentication failure with termination of the transport connection. For reauthentication, an authentication transaction timeout should not be treated as an authentication failure. Authentication commands used to continue that transaction after an authentication transaction timeout should be aborted with a status code of Command Sequence Error.

Figure 727 shows an example of authentication transaction for NVMe/TCP.

Figure 727: Example of authentication transaction for NVMe/TCP


Host
Controller
NVMe/TCP transport session establishment
Connect Command
Connect Response
Authentication Transaction
Queue set up

1. An NVMe/TCP transport session is established
2. The Connect exchange is performed to set up an NVMe Queue and associate host to controller
3. The host performs an authentication transaction with the controller to authenticate the end-points
4. Queue is set up, ready for subsequent operations

8.3.4.2.1 Special considerations for In-band Authentication of Discovery subsystems

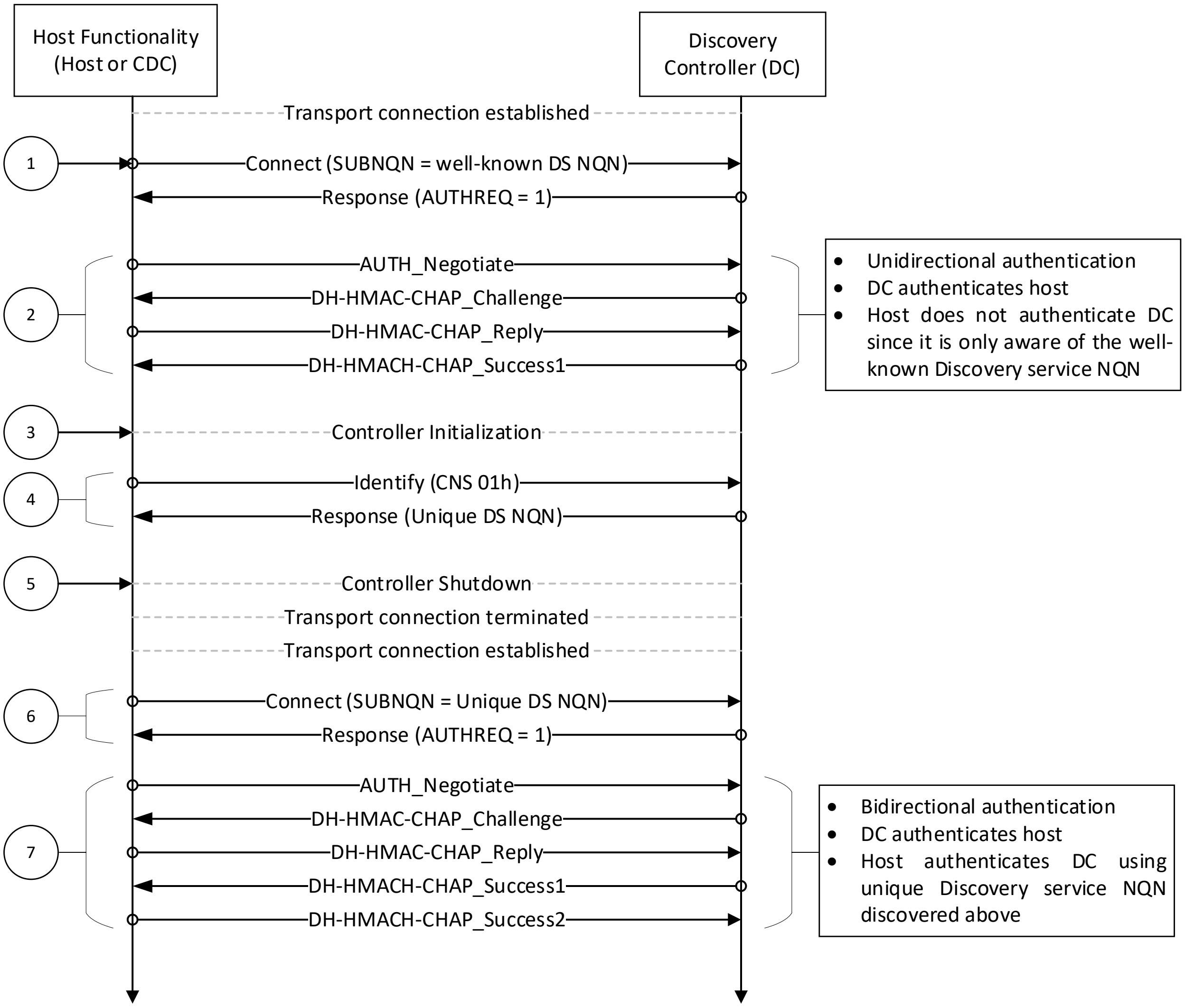
Hosts that have been configured to authenticate Discovery subsystems with an in-band authentication protocol that supports both unidirectional authentication and bidirectional authentication (e.g., DH-HMAC-CHAP, refer to section 8.3.4.5) should behave as follows:

- If the host connected to a Discovery subsystem using the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery) and the Discovery subsystem did not request authentication, then the host should not perform an authentication transaction;
- If the host connected to a Discovery subsystem using the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery) and the Discovery subsystem requested authentication, then the host should perform only unidirectional authentication (i.e., the Discovery subsystem may authenticate the host, but the host should not authenticate the well-known Discovery Service NQN); or
- If the host connected to a Discovery subsystem using the unique Discovery Service NQN for that Discovery subsystem (refer to section 3.1.3.3), regardless of whether the Discovery subsystem requested authentication, then the host may perform unidirectional authentication or bidirectional authentication (i.e., the host may authenticate the unique Discovery Service NQN for that Discovery subsystem).

Figure 726 illustrates a process that a host is able to use to retrieve the unique Discovery Service NQN and perform in-band authentication for a Discovery subsystem if that host has been configured to authenticate Discovery subsystems and the Discovery subsystem that the host connects to also requires authentication (i.e., the AUTHREQ field is not cleared to zero) using DH-HMAC-CHAP for authentication. This process includes:

1. connect to the Discovery subsystem using the well-known Discovery Service NQN (i.e., nqn.2014-08.org.nvmexpress.discovery);
2. perform unidirectional authentication with the Discovery subsystem;
3. perform controller initialization (refer to section 3.5);
4. retrieve the unique Discovery Service NQN (refer to section 3.1.3.3);
5. perform controller shutdown (refer to section 3.6);
6. reconnect to the Discovery subsystem using the unique Discovery Service NQN for that Discovery subsystem; and
7. perform bidirectional authentication with the Discovery subsystem using the unique Discovery Service NQN for that Discovery subsystem.

Figure 726: Unique Discovery Service NQN retrieval for bidirectional authentication


Host Functionality (Host or CDC)
Discovery Controller (DC)
Transport connection established
1
Connect (SUBNQN = well-known DS NQN)
Response (AUTHREQ = 1)
2
AUTH_Negotiate
DH-HMAC-CHAP_Challenge
DH-HMAC-CHAP_Reply
DH-HMACH-CHAP_Success1
Unidirectional authentication
DC authenticates host
Host does not authenticate DC since it is only aware of the well-known Discovery service NQN
3
Controller Initialization
4
Identify (CNS 01h)
Response (Unique DS NQN)
5
Controller Shutdown
Transport connection terminated
Transport connection established
6
Connect (SUBNQN = Unique DS NQN)
Response (AUTHREQ = 1)
7
AUTH_Negotiate
DH-HMAC-CHAP_Challenge
DH-HMAC-CHAP_Reply
DH-HMACH-CHAP_Success1
DH-HMACH-CHAP_Success2
Bidirectional authentication
DC authenticates host
Host authenticates DC using unique Discovery service NQN discovered above

Description of NVM Express Boot Specification 1.1 changes

Modify section 1 as shown below:

1 Introduction

…

1.4 References

…

RFC 4122, P. Leach, M. Mealling, R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", July 2005. Available from https://www.rfc-editor.org/rfc/rfc4122.

…

Modify section 3 as shown below:

3 Boot Mechanisms

…

3.1 NVMe-oF Boot default values for Host NQN and HostID

…

3.1.1 Creating HostNQN and HostID Using SMBIOS System UUID

…

Generation of the Host NQN value:

> The System UUID is converted into a NQN per the NVM Express Base Specification, NVMe Qualified Names section, using the second format which creates a unique identifier without a naming authority and using a UUID. String encoded values of SMBIOS UUIDs shall be normalized to lowercase for maximal compatibility and ease of comparison between values (refer to RFC 4122).
>
> For example, the System UUID "00112233-4455-6677-8899-aabbccddeeff" results in the NQN string:
>
> > "nqn.2014-08.org.nvmexpress:uuid:00112233-4455-6677-8899-aabbccddeeff"

…