LEGAL NOTICE:

This 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 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: **"© 2008 to 2023 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.

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

NVM Express® Technical Proposal (TP)

Technical Proposal ID	**8027 Enhanced HFI DHCP Client Options Support**
Revision Date	**2023.10.07**
Builds on Specification(s)	**NVM Express Boot Specification 1.0**
References	**TP4126 NVMe-oF Boot HostNQN and HostID**

Technical Proposal Author(s)

Name	**Company**
Doug Farley	Dell Technologies
Phil Cayton	Intel
Lenny Szubowicz	Red Hat
Wenhua Liu	VMWare
Hannes Reinecke	SUSE
Martin Wilck	SUSE
Rob Davis	NVIDIA

Technical Proposal Overview

Currently, the pre-OS and OS environments may receive different addresses from a DHCP Server if the drivers use different default options for the request. This could result in an OS environment that is unable to reconnect to the same namespaces that were available to the pre-OS environment; this would lead to a system being unable to successfully boot or access configured namespaces as subsystems may not recognize through other filtering means.

This proposal enables support for various meta-data attributes related to DHCP for IPv4 and IPv6 handling in the Boot Specification per HFI Transport Info Descriptor for NVMe/TCP to improve IP address consistency. With the new feature, the Pre-OS environment will be able to pass information in the NBFT to the OS-environment and specify the DHCP options it provided to the DHCP Server. The OS environment will be able to use this information to provide the same DHCP options to the DHCP Server to reestablish connectivity to namespaces with the same addresses.

This proposal will also include guidance for coexistence issues related to IPv4 and IPv6 automatic address assignment (e.g., DHCP, neighbor discovery, zero configuration networking), and provide recommendations for pre-OS, OS driver, and application writers for common defaults. Support for an OS environment's ability to differentiate between schemes used by the pre-OS environment will be established to provide more correlation consistency between DHCP clients between environments.

Revision History

Revision Date	Change Description
2023.04.25	Initial version from Boot Task Group
2023.05.02	Update field text
2023.05.16	Fix field highlights, case and links
2023.06.08	Edits from TWG review
2023.06.20	Edits from BTG Phase 3 content wording
2023.06.27	More phase 3 grammatical edit.
2023.07.27	Edits during TWG.
2023.08.15	Address 30-day Member Review comments from Mike Allison and Judy Brock
2023.10.07	Integrated

Description for Changes Document for NVM Express Boot Specification 1.0

New Features/Feature Enhancements/Required Changes:

- Add new optional fields to the HFI Transport Info Descriptor for TCP/IP in the NBFT (e.g., DHCP 4/6 Identifiers and DHCP meta-data content per HFI descriptor, differentiation data for various auto-negotiation and auto-configuration schemes that are used in DHCP or address assignment).
    - Flag in the HFI to indicate the use of this optional feature,
    - Address Assignment Scheme Identification,
    - Client Identifier Type; and
    - Client Identifier for (e.g., 'DHCPv4', 'DHCPv6', 'IPv6-ND')
- Additionally, the NVMe Boot Specification Informative Annex will be amended to:
    - specify how to use these fields and their best practices,
    - describe how to handle the various DHCP4 and DHCP6 Client Identifier schemes and which one is preferred,
    - provide recommendations on usage of the new fields in both the Pre-OS and OS environments; and
    - advise on how the OS drivers shall parse upwardly compatible changes to versioned descriptors.

Markup Conventions:

Black:	Unchanged (however, hot links are removed)
~~Red Strikethrough~~:	Deleted
Blue:	New
Blue Highlighted:	TBD values, anchors, and links to be inserted in new text
~~Purple Strikethrough~~:	Text that was moved
Purple:	Destination of moved text
Orange:	Tech is pulled in from a referenced Technical Proposal
<Green Bracketed>:	Notes to editor

Description of Specification Changes for NVM Express Boot Specification 1.0

Modify section 3.TBD <Note to Editor: 3.TBD is taken from TP4126.>

3.TBD NVMe-oF Boot default values for HostNQN and HostID

…

The pre-OS boot environment and OS environment should use a fixed platform UUID to create a HostNQN and HostID. The implementation should use the System UUID found in the SMBIOS table. The System Management BIOS (SMBIOS) Reference Specification is described in DSP0134. The SMBIOS table is typically available to pre-OS firmware and Expansion ROM Firmware in the pre-OS boot environment as well as to the OS environment. The system UUID is a value that is likely to be fixed at manufacturing or virtual machine creation, perhaps displayed on system tags or via configuration menus, and visible to the system administrator ahead of any OS installation. These attributes allow the administrator to pre-generate the expected HostNQN and HostID values that will be used by the pre-OS boot environment and the OS environment. ~~Apriori~~ Appropriate prior understanding of the values allow the administrator to configure the NVM subsystem in the least number of steps and before there is any attempt to boot or install the OS.

…

Modify section 3.2.1 Figure 5 "NBFT Elements" to add new rows:

Figure 5: NBFT Elements

Element	Descriptor Type Designator	Description	Reference
Header	N/A	An ACPI structure header with some additional NBFT specific info.	3.2.2.1
Control Descriptor	1h	Indicates the location of host, HFI, SSNS, security, and discovery descriptors.	3.2.2.2
Host Descriptor	2h	Host information.	3.2.2.3
HFI Descriptor	3h	An indexable table of HFI Descriptors, one for each fabric interface on the host.	3.2.2.4
Subsystem Namespace Descriptor	4h	An indexable table of SSNS Descriptors.	3.2.2.5
Security Descriptor	5h	An indexable table of Security descriptors.	3.2.2.6
Discovery Descriptor	6h	An indexable table of Discovery Ddescriptors.	3.2.2.7
Heap Resident Structures			
HFI Transport Descriptor	7h	Indicated by an HFI Descriptor, corresponds to a specific transport for a single HFI.~~.~~	3.2.2.4.1.1
Reserved	8h		
SSNS Extended Info Descriptor	9h	Indicated by an SSNS Descriptor if required.	3.2.2.5.5
HFI Extended Info Descriptor	Ah	Indicated by an HFI Transport info descriptor.	3.2.2.4.1.TBD
Reserved	Bh to FFh	All other type designator values are reserved.	

Modify section 3.2.2.1 Figure 8 "NBFT Table":

Figure 8: NBFT Table

Bytes	Description
Header	
03:00	**Signature:** An ASCII string representation of the table identifier. This field shall be set to the value 4E424654h (i.e., "NBFT") for the NVMe-oF Boot Firmware Table.

Figure 8: NBFT Table

Bytes	Description
07:04	**Length:** The length of the table, in bytes, including the header, starting from offset 0h. This field is used to record the size of the entire table.
08	**Major Revision:** The major revision of the structure corresponding to the Signature field. Larger major revision numbers should not be assumed backward compatible to lower major revision numbers with the same signature. This field shall be set to '1'.
09	**Checksum:** The entire table, including the Checksum field, shall sum to 0h to be considered valid. This checksum shall follow ACPI table checksum requirements. Refer to the Advanced Configuration and Power Interface (ACPI) Specification.
15:10	**OEMID:** OEMID shall be populated by the NBFT driver writer by an OEM-supplied string that identifies the OEM. All trailing bytes shall be null.
23:16	**OEM Table ID:** This field shall be populated by the NBFT driver writer with an OEM-supplied string that the OEM uses to identify the particular data table. This field is particularly useful when defining a definition block to distinguish definition block functions. The OEM assigns each dissimilar table a new OEM Table ID.
27:24	**OEM Revision:** An OEM-supplied revision number. Larger numbers are assumed to be newer revisions.
31:28	**Creator ID:** Vendor ID of utility that created the table. For instance, this may be the ID for the ASL Compiler.
35:32	**Creator Revision:** Revision of utility that created the table. For instance, this may be the ID for the ASL Compiler.
39:36	**Heap Offset (HO):** This field indicates the offset in bytes of the heap, if any, from byte offset 0h of the NBFT Table Header.
43:40	**Heap Length (HL):** The length of the heap, if any.
49:44	**Driver Signature Heap Object Reference:** This field indicates the offset in bytes of a heap object containing the Driver Signature, if any, from byte offset 0h of the NBFT Table Header. For more detail on Driver Signature, refer to section A.3.1.3.4. **Bytes** / **Description** 03:00 — **Offset:** Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header. 05:04 — **Length:** Length in bytes of the heap object, if any.
50	**Minor Revision:** The minor revision of the structure corresponding to the Signature field. If the major revision numbers are the same, any minor revision number differences shall be backwards compatable with the same signature. This field shall be ~~cleared~~ set to ~~'0'~~ '1'.
63:51	Reserved
Control Descriptor	
64	**Structure ID:** This field specifies the element (refer to Figure 5). ~~-~~This field shall be set to 1h (i.e., Control).
65	**Major Revision:** The major revision of the structure corresponding to the Signature field. Larger major revision numbers should not be assumed backward compatible to lower major revision numbers with the same signature. This field shall be set to '1'.
66	**Minor Revision:** The minor revision of the structure corresponding to the signature field. If the major revision numbers are the same, any minor revision number differences shall be backwards compatable with the same signature. This field shall be cleared to '0'.
67	Reserved
69:68	**Control Structure Length (CSL):** This field indicates the length in bytes of the Control Descriptor.
70	**Flags:** **Bits** / **Feature** 07:01 — Reserved 00 — **Block Valid:** If set to '1', then this structure is valid. If cleared to '0', then this structure is not valid.
71	Reserved

Figure 8: NBFT Table

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>77:72</td><td>Host Descriptor (HDESC): This field indicates the location and length of the Host Descriptor (refer to Figure 10).
<table><tr><th>Bytes</th><th>Description</th></tr><tr><td>03:00</td><td>Offset: Offset in bytes of the Host Descriptor from byte offset 0h of the NBFT Table Header. The Offset field shall be set to a non-zero value.</td></tr><tr><td>05:04</td><td>Length: Length in bytes of the Host Descriptor. The Length field shall be set to a non-zero value.</td></tr></table></td></tr>
<tr><td>78</td><td>Host Descriptor Version (HSV): This field indicates the version of the Host Descriptor. This field shall be set to ’1’.</td></tr>
<tr><td>79</td><td>Reserved</td></tr>
<tr><th colspan="2">Host Fabric Interface Descriptor Information</th></tr>
<tr><td>83:80</td><td>HFI Descriptor List Offset (HFIO): If this field is set to a non-zero value, then this field indicates the offset in bytes of the HFI Descriptor List (refer to Figure 11), if any, from byte offset 0h of the NBFT Table Header. If the NumHFI field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>85:84</td><td>HFI Descriptor Length (HFIL): This field indicates the length in bytes of each HFI Descriptor, if any. If the NumHFI field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>86</td><td>HFI Descriptor Version (HFIV): This field indicates the version of each HFI Descriptor. This field shall be set to ’1’.</td></tr>
<tr><td>87</td><td>Number of Host Fabric Interface Descriptors (NumHFI): This field indicates the number of HFI Descriptors (refer to Figure 12) in the HFI Descriptor List (refer to Figure 11), if any. If no interfaces have been configured, then this field shall be cleared to 0h.</td></tr>
<tr><th colspan="2">Subsystem Namespace Descriptor Information</th></tr>
<tr><td>91:88</td><td>SSNS Descriptor List Offset (SSNSO):: This field indicates the offset in bytes of the SSNS Descriptor List (refer to Figure 15), if any, from byte offset 0h of the NBFT Table Header. If the NumSSNS field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>93:92</td><td>SSNS Descriptor Length (SSNSL): This field indicates the length in bytes of each SSNS Descriptor, if any. If the NumSSNS field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>94</td><td>SSNS Descriptor Version (SSNSV): This field indicates the version of the SSNS Descriptor. This field shall be set to ’1’.</td></tr>
<tr><td>95</td><td>Number of Subsystem and Namespace Descriptors (NumSSNS): This field indicates the number of Subsystem Namespace (SSNS) Descriptors (refer to Figure 16) in the SSNS Descriptor List (refer to Figure 15), if any.</td></tr>
<tr><th colspan="2">Security Profile Information</th></tr>
<tr><td>99:96</td><td>Security Profile Descriptor List Offset (SECO): This field indicates the offset in bytes of the Security Profile Descriptor List (refer to Figure 21), if any, from byte offset 0h of the NBFT Table Header. If the NumSec field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>101:100</td><td>Security Profile Descriptor Length (SECL): This field indicates the length in bytes of each Security Profile Descriptor, if any. If the NumSec field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>102</td><td>Security Profile Descriptor Version (SECV): This field indicates the version of the Security Profile Descriptor. This field shall be set to ’1’.</td></tr>
<tr><td>103</td><td>Number of Security Profile Descriptors (NumSec): This field indicates the number of Security Profile Descriptors (refer to Figure 22), if any, in the Security Profile Descriptor List (refer to Figure 21).</td></tr>
<tr><th colspan="2">Discovery Descriptor Information</th></tr>
<tr><td>107:104</td><td>Discovery Descriptor Offset (DISCO): This field indicates the offset in bytes of the Discovery Descriptor List (refer to Figure 24), if any, from byte offset 0h of the NBFT Table Header. If the NumDisc field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>109:108</td><td>Discovery Descriptor Length (DISCL): This field indicates the length in bytes of each Discovery Descriptor, if any. If the NumDisc field is cleared to 0h, then this field is reserved.</td></tr>
<tr><td>110</td><td>Discovery Descriptor Version (DISCV): This field indicates the version of the Discovery Descriptor. This field shall be set to ’1’.</td></tr>
<tr><td>111</td><td>Number of Discovery Descriptors (NumDisc): This field indicates the number of Discovery Descriptors Figure 25), if any, in the Discovery Descriptor List (refer to Figure 24), if any.</td></tr>
<tr><td>127:112</td><td>Reserved</td></tr>
</table>

Update Section 3.2.2.4.1 Figure 12 "HFI Transport Info Descriptor Header Template":

Figure 12: HFI Transport Info Descriptor Header Template

Bytes	Description
00	**Structure ID:** This field shall be set to 7h (i.e., HFI Transport Info; refer to Figure 6).
01	**Version:** This field shall be set to 1h.
02	**HFI Transport Type:** Refer to Figure 8.
03	**Transport Info Version:** ~~Implementations compliant to this specification shall set this field to 1h.~~This field shall be set to 1h.
05:04	**HFI Descriptor Index:** The value of the HFI Descriptor Index field of the HFI Descriptor (refer to Figure 12) whose HFI Transport Info Descriptor Heap Object Reference field indicates this HFI Transport Info Descriptor.

Update Section 3.2.2.4.1.1 Figure 13: "HFI Transport Info Descriptor – NVMe/TCP":

Figure 13: HFI Transport Info Descriptor – NVMe/TCP

Bytes	O/M[1]	Description
00	M	**Structure ID:** This field shall be set to 7h (i.e., HFI Transport Info; refer to Figure 6).
01	M	**Version:** The value of the HFI Transport Version field shall always monotomically increase. This field shall be set to 1h.
02	M	**HFI Transport Type:** This field shall be set to 03h (i.e., NVMe/TCP; refer to Figure 8).
03	M	**Transport Info Version:** This field shall be set to 2h. ~~Implementations compliant to this specification shall set this field to 1h.~~
05:04	M	**HFI Descriptor Index:** The value of the HFI Descriptor Index field of the HFI Descriptor (refer to Figure 12) whose HFI Transport Info Descriptor Heap Object Reference field indicates this HFI Transport Info Descriptor.

Figure 13: HFI Transport Info Descriptor – NVMe/TCP

<table>
<tr><th>Bytes</th><th>O/M[1]</th><th>Description</th></tr>
<tr><td>06</td><td>M</td><td>HFI Transport Flags:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:~~03~~04</td><td>Reserved</td></tr>
<tr><td>03</td><td>Interface Address Autoconfigured: If set to ‘1’, then the IP Address, Subnet Mask Prefix, and IP Gateway fields are populated by an advanced stateless mechanism (e.g., IPv6-SLAAC or IPv6-ND); in addition, the DHCP Override bit shall be cleared to ‘0’, and the IP Origin field shall be cleared to 0h.

The values of these HFI fields should not be the source for the values used by the OS. The process for an OS driver to acquire these values is outside the scope of this specification.

If cleared to ‘0’, then the HFI information was not set by such a mechanism and this bit is reserved.</td></tr>
<tr><td>02</td><td>DHCP Override: If set to ’1’, then HFI information was populated by consuming the DHCP on this interface. In addition, for implementations compliant with Major Revision ‘1’ and Minor Revision ‘1’ or later of this specification, the IPv4 and IPv6 DHCP Client Identifier Heap identifier Identity Association Identifier (IAID) and DHCP Unique Identifier (DUID) objects in the HFI Transport Info Extended Information Descriptor (refer to FIGURE TBD.1) shall also be populated.

If cleared to ’0’, then the HFI information was set administratively by a configuration interface to the driver and pre-OS environment.</td></tr>
<tr><td>01</td><td>Global Route vs. Link Local Override Flag: If set to ’1’, then the BIOS utilized this interface described by HFI to be the default route with highest priority. If cleared to 0, then routes are local to their own scope.</td></tr>
<tr><td>00</td><td>Descriptor Valid: If set to ’1’, then this descriptor is valid. If cleared to 0, then this descriptor is reserved.</td></tr>
</table></td></tr>
<tr><td>10:07</td><td>M</td><td>PCI Express Routing ID for the HFI Transport Function: This field indicates the PCI Express Routing ID as specified in the PCI Express Base Specification.

Note: If the platform supports PCIe ARI, then the Device bits and the Function bits may be concatenated for that purpose.
<table>
<tr><th>Bits</th><th>Description</th><th>Refer to</th></tr>
<tr><td>31:16</td><td>Segment Group Number</td><td>PCI Firmware Specification</td></tr>
<tr><td>15:08</td><td>Bus</td><td rowspan="3">PCI Express Base Specification</td></tr>
<tr><td>07:03</td><td>Device</td></tr>
<tr><td>02:00</td><td>Function</td></tr>
</table></td></tr>
<tr><td>16:11</td><td>M</td><td>MAC Address: The MAC address of this HFI, in EUI-48™ format, as defined in the IEEE Guidelines for Use of Extended Unique Identifiers. This field shall be set to a non-zero value.</td></tr>
<tr><td>18:17</td><td>O</td><td>VLAN: If this field is set to a non-zero value, then this field contains the VLAN identifier if the VLAN associated with this HFI, as defined in IEEE 802.1q-2018.

If no VLAN is associated with this HFI, then this field shall be cleared to 0h.</td></tr>
<tr><td>19</td><td>O</td><td>IP Origin: If this field is set to a non-zero value, then this field indicates the source of Ethernet L3 configuration information used by the driver for this interface. ~~The values defined in this field correspond to those defined in the Win 32 API: NL_PREFIX_ORIGIN enumeration specification.~~

The IP Origin values are defined as:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0h</td><td>IpPrefixOriginOther</td></tr>
<tr><td>1h</td><td>IpPrefixOriginManual</td></tr>
<tr><td>2h</td><td>Reserved</td></tr>
</table></td></tr>
</table>

Figure 13: HFI Transport Info Descriptor – NVMe/TCP

<table>
<tr><th>Bytes</th><th>O/M[1]</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>3h</td><td>IpPrefixOriginDhcp</td></tr>
<tr><td>4h</td><td>IpPrefixOriginRouterAdvertisement</td></tr>
<tr><td>5h to FFh</td><td>Reserved</td></tr>
</table>

If this field is set to 1h, then the IP Address was not automatically configured on this interface, and the IP Address specified in this HFI descriptor may also be used by the OS for connections via this HFI.

If this field is set to 3h, then the IP Address was acquired through DHCP, and the IP Address specified in this HFI should not be reused by the OS and the HFI Transport Info Extended Information Descriptor shall contain the DHCP Client Identifier used by the Pre-OS driver.

If this field is set to 4h, then the IP Address was acquired through an external process like IPv6 RA and the IP Address specified in this HFI should not be reused by the OS. This process is outside the scope of this specification.

This field should be cleared to ‘0’ if the IP Origin field is unused by driver.

The values defined in this field correspond to those defined in the Win 32 API: NL_PREFIX_ORIGIN enumeration specification.</td></tr>
<tr><td>35:20</td><td>M</td><td>IP Address: This field indicates the IPv4 or IPv6 address of this HFI. This field shall be set to a non-zero value.

The format of this field is defined by section 1.5.5.1.</td></tr>
<tr><td>36</td><td>M</td><td>Subnet Mask Prefix: This field indicates the IPv4 or IPv6 subnet mask in CIDR routing prefix notation.</td></tr>
<tr><td>52:37</td><td>O</td><td>IP Gateway If this field is set to a non-zero value, this field indicates the IPv4 or IPv6 address of the IP gateway for this HFI. If this field is cleared to 0h, then no IP gateway is specified.</td></tr>
<tr><td>53</td><td></td><td>Reserved</td></tr>
<tr><td>55:54</td><td>O</td><td>Route Metric: If this field is set to a non-zero value, this field indicates the cost value for the route indicated by this HF. This field contains the value utilized by the pre-OS driver when chosing among all availible routes. Lower values relate to higher priority. Refer to IETF RFC 4249.

If the pre-OS driver supports routing and did not configure a specific route metric for this interface, then the pre-OS driver should set this value to 500.

If the pre-OS driver does not support routing, then this field should be cleared to 0h.</td></tr>
<tr><td>71:56</td><td>O</td><td>Primary DNS: If this field is set to a non-zero value, this field indicates the IPv4 or IPv6 address of the Primary DNS server for this HFI, if any, from byte offset 0h of the NBFT Table Header. If this field is cleared to 0h, then no Primary DNS is specified.</td></tr>
<tr><td>87:72</td><td>O</td><td>Secondary DNS: If this field is set to a non-zero value, this field indicates the IPv4 or IPv6 address of the Secondary DNS server for this HFI, if any, from byte offset 0h of the NBFT Table Header. If this field is cleared to ’0’, then no Secondary DNS is specified.</td></tr>
<tr><td>103:88</td><td>O</td><td>DHCP Server: If the DHCP Override bit is set to ’1’, then this field indicates the IPv4 or IPv6 address of the DHCP server used to assign this HFI address. If that bit is cleared to ’0’, then this field is reserved.</td></tr>
<tr><td>109:104</td><td>O</td><td>Host Name Heap Object Reference: If this field is set to a non-zero value, then:

a) this field indicates the location and size of a heap object containing a Host Name string (refer to section 1.5.6); and
b) the Offset field and the Length field shall be set to non-zero values.

If this field is cleared to 0h, then this field is reserved.
<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>03:00</td><td>Offset: Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header.</td></tr>
</table>
</td></tr>
</table>

Figure 13: HFI Transport Info Descriptor – NVMe/TCP

Bytes	O/M[1]	Description
		05:04 **Length:** Length in bytes of the heap object, if any.
115:110	O	**HFI Transport Info Extended Information Descriptor Object Reference:** If this field is set to a non-zero value, then: a) this field indicates the location and size of a heap object containing a HFI Transport Info Extended Information Descriptor (refer to Figure <TBD.1>); and b) the Offset field and the Length field shall be set to non-zero values. If this field is cleared to 0h, then this field is reserved. In version 1.0 of this specification, this field was reserved. **Bytes** / **Description** 03:00 **Offset:** Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header. 05:04 **Length:** Length in bytes of the heap object, if any.
127:~~110~~116		Reserved

Notes:
1. O/M definition: O = Optional, M = Mandatory.

Add section 3.2.2.4.1.TBD as shown below:

3.2.2.4.1.TBD HFI Transport Info Extended Information Descriptor

The HFI Transport Info Extended Information Descriptor is structured as a standard table located in heap space as required.

Figure <TBD.1>: HFI Transport Info Extended Information Descriptor

Bytes	O/M[1]	Description
00	M	**Structure ID:** This field shall be set to Ah (i.e., HFI Ext Info; refer to Figure 5).
01	M	**Version:** This field shall be set to 1h.
03:02	M	**HFI Transport Info Descriptor Index:** This field shall be set to the same vaue as the HFI Transport Info Descriptor Index field of the referencing HFI Transport Info Descriptor (refer to Figure 13 and the example in section A.3.2.TBD).
07:04	M	**Flags:** **Bits** / **Feature** 31:02 Reserved 01 **DHCP Client Identifier:** If set to '1', then the IPv4 and IPv6 DHCP Client Identifier Heap Object Reference field is valid and the IP Origin field shall be set to '3'. If cleared to '0', then the IPv4 and IPv6 DHCP Client Identifier Heap Object Reference is reserved. 00 **Descriptor Valid:** If set to '1', then this descriptor is valid. If cleared to '0', then this descriptor is reserved.
13:08	O	**IPv4 and IPv6 DHCP Client Identifier Heap IAID Object Reference**: If the DHCP Client Identifier bit is set to '1', then this then this heap object references the Identity Association Identifier (IAID). This identifier shall be constructed in a manner that is conformant to the RFC4361 Client Identifier. Where the IAID field is as specififed in RFC3315 and RFC6355. The IAID is stored in little endian format within the heap object. If this field is cleared to 0h, then this field is reserved. **Bytes** / **Description**

Figure <TBD.1>: HFI Transport Info Extended Information Descriptor

Bytes	O/M[1]	Description
		03:00 **Offset:** Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header. 05:04 **Length:** Length in bytes of the heap object, if any.
19:14	O	**IPv4 and IPv6 DHCP Client Identifier Heap DUID Object Reference**: If the DHCP Client Identifier bit is set to '1', then this then this heap object refereces the DHCP Unique Identifier (DUID). This identifier shall be constructed in a manner that is conformant to the RFC4361 Client Identifier. Where the DUID field is as specifiifed in RFC3315 and RFC6355. The DUID is stored in little endian format within the heap object. If this field is cleared to 0h, then this field is reserved. **Bytes** / **Description** 03:00 **Offset:** Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header. 05:04 **Length:** Length in bytes of the heap object, if any.
Notes: 1. O/M definition: O = Optional, M = Mandatory.		

Add new section A.3.2.TBD as shown below:

A.3.2.TBD HFI TCP/IP Address DHCP Configuration

The HFI Transport Information Extended Info Descriptor may be utilized for either DHCPv4 and DHCPv6 information. In cases where both IPv4 and IPv6 were configured simultaneously their details would be present in separate HFI Descriptors. Each of those HFI Descriptors may then have an Extended Info Descriptor with their relevant DHCP Client Identifier information.

What follows are examples of DHCPv4 and DHCPv6 information in such descriptors.

In an DHCPv4 example in <Figure TBD.2>, where the HFI Transport Info Descriptor has an Index of 05h, referencing a parent HFI Descriptor of that index with:

- IAID of 00061af0h, and
- DUID of 01_01041faa_01070001_12317afeh.

In an DHCPv4 example in <Figure TBD.2>, the DUID was comprised of a 1-byte type field and 12-bytes of identify (refer to RFC4361 for more details).

Figure <TBD.2>: Example DHCPv4 Heap

Descriptor	Bytes	Field	Example
HFI Transport Info Extended Information Descriptor	00	Structure ID	Ah
	01	Version	01h
	03:02	HFI Transport Info Descriptor Index	05h
	07:04	Flags	03h
	13:08	IPv4 and IPv6 DHCP Client Identifier IAID Heap Object Reference	**Bytes** / **Field** / **Example** 03:00 Offset 200h 05:04 Length 04h

Figure <TBD.2>: Example DHCPv4 Heap

Descriptor	Bytes	Field	Example
	19:14	IPv4 and IPv6 DHCP Client Identifier DUID Heap Object Reference	Bytes: 03:00, Field: Offset, Example: 400h Bytes: 05:04, Field: Length, Example: 17h
IPv4 and IPv6 DHCP Client Identifier IAID Heap Object	03:00	n/a	00h 06h 1ah f0h
IPv4 and IPv6 DHCP Client Identifier DUID Heap Object	12:00	n/a	01h 01h 04h 1fh aah 01h 07h 00h 01h 12h 31h 7ah feh

In an DHCPv6 example in <Figure TBD.3>, where the HFI Transport Info Descriptor Index of beh, referencing a parent HFI Descriptor of that index with:

- IAID of ff061af0h, and
- DUID of 04_2ec5b3e1_0c744013_9db5b706_7ba572a7h.

Figure <TBD.3>: Example DHCPv6 Heap

Descriptor	Bytes	Field	Example
HFI Transport Info Extended Information Descriptor	00	Structure ID	Ah
	01	Version	01h
	03:02	HFI Transport Info Descriptor Index	beh
	07:04	Flags	03h
	13:08	IPv4 and IPv6 DHCP Client Identifier IAID Heap Object Reference	Bytes: 03:00, Field: Offset, Example: 300h Bytes: 05:04, Field: Length, Example: 04h
	19:14	IPv4 and IPv6 DHCP Client Identifier DUID Heap Object Reference	Bytes: 03:00, Field: Offset, Example: 500h Bytes: 05:04, Field: Length, Example: 17h
IPv4 and IPv6 DHCP Client Identifier IAID Heap Object	03:00	n/a	ffh 06h 1ah f0h
IPv4 and IPv6 DHCP Client Identifier DUID Heap Object	16:00	n/a	04h 2eh c5h b3h e1h 0ch 74h 40h 13h 9dh b5h b7h 06h 7bh a5h 72h a7h

Add new sub-sections below A.4.1.2 as shown below:

A.4.1.2.TBD1 NBFT Version Checking

An NBFT consumer written to expect NBFT header X.y (denoted in Major.Minor number format) should accept any NBFT with header X.z. The NBFT consumer may at its discretion handle NBFT header W.z where W < X. However, an NBFT consumer written to header version X.y should not accept an NBFT with version Z.y where Z > X.

A.4.1.2.TBD2 Unexpected Values in Reserved Fields

Producers and consumers may be implemented at different levels of compatibility as such consumers of the NBFT should not attempt to decode any values in any reserved field or value.

A.4.1.2.TBD3 Unepected Descriptor Types Fields

For example, if a new HFI Transport Descriptor Type values (See Figure 7) appears, an OS should flag an informational message for diagnostics but otherwise ignore any features or functions the OS is not able to decode or consume.

A.4.1.2.TBD4 NBFT Data including DHCP Client Information

An NBFT that implements version 1.0 (refer to Figure 8 for Major Revision and Minor Revision field) of this specification may not contain enough data to recreate the Client Identifier in the OS environment. To acquire the same IP address in an OS environment, the fields in the HFI and HFI Ext for DHCP Client Identifier provide the Client Identifier used by the pre-OS environment. Presented in Figure <TBD.4> is a reference to when a Client Identifier may be assembled deterministically.

Figure <TBD.4>: Pre-OS and OS NBFT version interactions for DHCP

		OS Driver NBFT Version	
		1.0	**1.1 (or later)**
Pre-OS NBFT Version	**1.0**	DHCP IP may differ	DHCP IP may differ
	1.1 (or later)	DHCP IP may differ	DHCP IP shall be the same