



LEGAL NOTICE:

© **Copyright 2008 to 2023 NVM Express®, Inc. ALL RIGHTS RESERVED.**

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

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008-2023 NVM Express, Inc.

NVM Express® Technical Errata

Errata ID	117
Change Date	2023/08/23
Affected Spec Ver.	NVM Express Base Specification, Rev 2.0c NVM Express TCP Transport Specification, Rev 1.0c NVM Express RDMA Transport Specification, Rev 1.0c
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Fred Knight	NetApp
John Meneghini	RedHat
David Black	Dell/EMC
Costa Sapuntzakis, Randy Jennings	Pure

Errata Overview

This Errata reorganizes section 3.9 (Keep Alive) and references to section 3.9 to increase clarity and decrease redundant and conflicting text.

Additionally, references to the Keep Alive Timer feature are corrected (many currently refer to Keep Alive feature).

Revision History

Revision Date	Change Description
3/6/23	Initial creation
3/20/23	<ul style="list-style-type: none">* Added section 3.9.TBDconfig for Keep Alive Timer configuration* Added section 3.9.TBDactive for Keep Alive Timer activation* Added Controller/Host sections to 3.9.1/3.9.2* Removed Keep Alive Timeout Interval from the general section; it gets used in 3.9.2.1.* Added examples of communication failure (e.g.)* Separated out clearing a Keep Alive value to '0' and setting it to higher than the maximum value (both end in the same error)* The controller "is expected" to round to KAS, not the controller "should" round to KAS, with added figure reference.* Reinstated and revised examples of activating Keep Alive Timer* Removed reminder text that the controller can drop a connection whenever it well pleases.* Made Keep Alive Timer/Timer Interval text in sections 3.9.1.1 & 3.9.2.1 specific to Keep Alive Command Based Keep Alive and Traffic Based Keep Alive, respectively* Revised Host text in section 3.9.1.2 and 3.9.2.2
3/27/23	<ul style="list-style-type: none">* Keep Alive Command Based Keep Alive -> Command Based Keep Alive* Clean up Command Based Keep Alive on Host text, including:

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008-2023 NVM Express, Inc.

	<ul style="list-style-type: none"> * host submits commands; it does not post them (also TBKA text), * why the host submits Keep Alives periodically, and * don't talk about the Keep Alive Timer stopping, but do talk about if a Keep Alive was received since the Keep Alive Timer started/restarted (when considering if the Keep Alive Timer expired). * In TBKA text, commands -> Admin commands and I/O commands. * Clarify 5.18 Keep Alive Command text based on other revisions. * Removed most of the text as redundant or partial with sections of 3.9. * Moved "controller should process Keep Alive command immediately" to this section. * Revise references from Keep Alive feature to Keep Alive Timer feature. * Updated references external to 3.9 to the new more specific section numbers * In the TCP Transport specification, replaced a Base spec figure number with its title instead.
4/17/23	<ul style="list-style-type: none"> * Fixed contradiction: NVMe base specification (KAS field description) says support for Keep Alive is required for all NVMe over Fabrics implementations support Keep Alive; NVMe over FC transport specification does not require Keep Alive be supported. Removed base specification requirement. * In TBKAS field description, replaced partial description of algorithms with names of algorithms. * Moved added reference from figure 341 to section 5.27.1.12 (the section where figure 341 is). * Got rid of (TBKAS) when it was not used to refer to the field. * For TBKA on the controller, made one bullet for a Keep Alive Timer Interval stopped, and two sub bullets from that on whether a new interval does or does not start. * Added "on the host" to the CBKA and TBKA host sections in "The Keep Alive Timer expires when:" bullets. * minor awk fix: "Admin Queue with the completion" -> "Admin Queue. The completion"; Context: "For Command Based Keep Alive, the Keep Alive command is sent periodically from the host to the controller over the Admin Queue. The completion of the Keep Alive command indicating that the host and controller are able to communicate."
4/28/23	<ul style="list-style-type: none"> * Limit comment about associations in 3.9.1 (Command Based Keep Alive) to message-based transports. * Replace reference to maintaining an association in 3.9.2 (Traffic Based Keep Alive) with avoiding a Keep Alive Timeout. * When -> If * When describing the KAS field and the Keep Alive command, say that the Keep Alive Timer feature is used to implement the Keep Alive capability. * In the description of KATO field in Figure 380 (Connect command), talk about setting the KATO field in the Keep Alive Timer feature, and then pointing to 3.9.TBDactivate to learn about activation (instead of indicating when it is activated). * Made KAS in the Identify Controller structure optional for Discovery Controllers. * Revised document to better fit ECN template
5/2/23	<ul style="list-style-type: none"> * Listed consistency fixes as "Fixed Contradictions" * Removed section numbers from Editorial Changes * Added phrase about Keep Alive capability to Keep Alive section (because we now reference Keep Alive capability when pointing to the Keep Alive section).
5/11/23	<ul style="list-style-type: none"> * For Command Based Keep Alive on the host "The Keep Alive Timer in the host expires if ... a Keep Alive Timeout did not occur"; the qualification came in to a version of this doc on 3/22 and does not seem to make sense, looks like cut/paste error; removing. * Removed "is expected to" when describing how controller rounds KATO value.
5/16/23	<ul style="list-style-type: none"> * I/O and Admin Controller types have a note that Keep Alive command might be required by the Transport. Keep Alive Feature should instead, and support for the Keep Alive command should depend on support of the Keep Alive Feature. * Discovery Controllers implement the Keep Alive Timer feature iff they support explicit persistent connections because supporting the Keep Alive command is tied to that, and you cannot support the Keep Alive Timer feature without supporting the Keep Alive command.

	* When explaining the host checking at half the KATO timeout value, make it an “e.g.” list.
5/19/23	* Add changes from 5/16/23 to the Description of Changes * Made it so there was only the other case (optional/prohibited) in those footnotes added 5/16/23. * “If using” -> “For” for CBKA/TBKA controller/host sections
5/23/23	* For TBKAS, do not talk about CBKA; just say TBKA is not supported if cleared to ‘0’
5/26/23	* For TBKA on the host, “command is completed” -> “received a completion for a ... command”. * In description of changes, “iff” -> “if and only if”.
7/11/23	* TM->® * Fixed verb tense in Description of changes * Added highlighting to new or modified references * Added “i.e.” and ‘)’ to KAS sentence in section 3.9 intro paragraph. * Grammar changes to Transport influence on Keep Alive Timeout feature support. * 5.18 Keep Alive command, controller resets Keep Alive Timer when the controller processes the command, not when the controller receives the command. * In TBKA Host section, “at half of the Keep Alive Timeout” -> “at KATT/2”, which makes the phrasing match the CBKA Host section. * Removed “(i.e., Feature Identifier 0Fh)” each time it is mentioned in section 3.9; “refer to section 5.27.1.12” should pin that down. Added “refer to section 5.27.1.12” for Set Feature command mentions in CBKA/TBKA controller algorithms. * In the CBKA/TBKA host sections, qualify “during the use of x on the host” when describing how the host prevents the controller from detecting a KATO. * When talking about restarting the Keep Alive Timeout Interval, added e.g. for why a Keep Alive Timeout would not have happened. * In 3.9.TBDCleanup Keep Alive Timeout Cleanup, add who is detecting the Keep Alive Timeout.
7/13/23	* “in the Keep Alive Timeout Interval” -> “during the Keep Alive Timeout Interval”
7/14/23	* KATT is not introduced until TP4129, so back it out and fix “at half of the Keep Alive Timeout” to “at half of the time equal to the Keep Alive Timeout value” for both CBKA and TBKA. * Fixed some double ‘ ‘. * KAS -> the KAS field
8/23/23	* Integrated
8/30/23	Fixed the Feature Support section numbers. Reformatted to fix navigation pane.

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.
<Green Bracketed>:	Notes to editor
PURPLE:	Moved (to this location)
PURPLE:	Moved (from this location)

Description of Changes

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008-2023 NVM Express, Inc.

NVM Express Base Specification, Rev 2.0c:

Editorial Changes:

- Refined reference of Discovery controller dropping a connection due to inactivity to the new Keep Alive Timeout Cleanup section.
- Keep Alive feature -> Keep Alive Timer feature.
- Structured the Keep Alive section:
 - Applicability: Transport dependency (not a section)
 - Configuration (new section)
 - Activation (new section)
 - Command Based Keep Alive (renamed section)
 - Controller (new section)
 - Host (new section)
 - Traffic Based Keep Alive (pre-existing section)
 - Controller (new section)
 - Host (new section)
 - Keep Alive Timeout Cleanup (new section)
- Keep Alive Timer Configuration:
 - Referred to Fabrics Command & Set Features command/Keep Alive Timer feature sections.
 - Added reminder that the controller is expected to round a Keep Alive Timeout value and reference the Set Features section that described it.
- Keep Alive Timer Activation:
 - Fixed text about how the Keep Alive Timer feature is enabled and reference the new configuration section.
 - Fixed incorrect reference for Connect command KATO field.
 - Added reminder that the host needs to keep space in the Admin Submission Queue while the Keep Alive Timer is active.
- Referred to the new cleanup section in all Controller/Host sections.
- Command Based Keep Alive
 - Removed "Keep Alive" from section name.
 - Referred to association instead of connection and scope that to message-based transports.
- Command Based Keep Alive, Host:
 - Clarified that Host may use Command Based Keep Alive regardless of what the target uses.
- Keep Alive Command:
 - Moved instruction for the controller to process a Keep Alive command as soon as fetched to here.
 - Added instruction that receiving the Keep Alive command restarts the Keep Alive Timer (if active).
- KAS:
 - KAS is optional for Discovery Controllers (some support explicit persistent connections)

Fixed Contradictions:

- I/O and Admin Controller types had a note that Keep Alive command might be required by the Transport. Keep Alive Timer feature now has that note (instead of being optional) in Controller – Feature Support tables, and the Keep Alive command in the Controller – Admin Command Support tables now notes that implementation depends on support of the Keep Alive Timer feature.
- Discovery Controllers implement the Keep Alive Timer feature if and only if they support explicit persistent connections. Updated the Controller – Feature Support table to reflect this.
- Keep Alive Transport support:
 - Allowed for a transport to not specify min/max values (both RDMA & TCP explicitly do not).
- Keep Alive Timer Configuration:
 - Corrected conditions for errors; one only gets an error for disabling Keep Alive Timeout feature if the NVMe Transport binding specification requires it to be enabled.
 - Corrected conditions for the controller forcing the minimum value (0 disables the Keep Alive timer feature, which might be permitted).
- Command Based Keep Alive, Controller:
 - Consolidated text about how the Keep Alive Timer is started/restarted/expired. This involved referencing the new activation section and removing text that:
 - only mentioned Set Features
 - only mentioned Keep Alive command,

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008-2023 NVM Express, Inc.

- directly tied Keep Alive Timer expiring with a Keep Alive Timeout occurring (retained text uses "may consider")
- Command Based Keep Alive, Host:
 - Consolidated text about how the Keep Alive Timer is started/restarted/expired. This involved removing text that:
 - only mentioned Keep Alive command
 - directly tied Keep Alive Timer expiring with a Keep Alive Timeout occurring (retained text uses "may consider")
- Traffic Based Keep Alive, Controller:
 - Consolidated text about how the Keep Alive Timer is started/restarted/expired. This involved referencing the new activation section and removing text that:
 - only mentioned Set Features
 - only mentioned Keep Alive command
 - directly tied Keep Alive Timer expiring with a Keep Alive Timeout occurring (retained text uses "may consider")
 - tied Keep Alive Timer expiry to only the Keep Alive Timeout Interval ending (not whether traffic was received)
 - stated the Admin command or I/O command just had to be submitted to the controller (not fetched by the controller)
- Traffic Based Keep Alive, Host:
 - Consolidated text about how the Keep Alive Timer is started/restarted/expired. This involved removing text that:
 - only mentioned Keep Alive command,
 - directly tied Keep Alive Timer expiring with a Keep Alive Timeout occurring (retained text uses "may consider").
- Cleanup:
 - Cleanup actions happen if a Keep Alive Timeout is detected (not just the Keep Alive Timer expiring)
 - A Keep Alive Timeout terminates all connections in an association, not just one.
- TBKAS:
 - Removed partial, conflicting description of the algorithm.
- KAS:
 - Removed requirement for all NVMe over Fabrics implementations to support the Keep Alive Timer feature (Fibre Channel does not require it).
- Keep Alive Command:
 - Removed partial, conflicting description of algorithm.
- Connect Command and Response:
 - Removed text about enabling Keep Alive Timer feature when Connect command completes and reference the new activation section.

NVM Express TCP Transport Specification, Rev 1.0c

Editorial Changes:

- Keep Alive feature -> Keep Alive Timer feature.
- Fixed two references to the base specification to use the name of the sections instead of the section numbers.

NVM Express RDMA Transport Specification, Rev 1.0c

Editorial Changes:

- Keep Alive Timer Feature -> Keep Alive Timer feature.

Description of NVM Express Base Specification, Rev 2.0c changes

3 NVM Express Architecture

3.1 NVM Controller Architecture

3.1.2 Controller Types

3.1.2.1 I/O Controller

3.1.2.1.1 Command Support

...

Figure 22: I/O Controller – Admin Command Support

Command	Command Support Requirements ¹	Reference
...		
Keep Alive	NOTE 2M ²	5.18
...		
Notes:		
1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited		
2. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory. Prohibited if the Keep Alive Timer feature is not supported (refer to section 3.9).		
...		

3.1.2.1.3 Features Support

...

Figure 25: I/O Controller – Feature Support

Feature Name	Feature Support Requirements ¹	Logged in Persistent Event Log ¹
...		
Keep Alive Timer	O M TBD	O
...		
Notes:		
...		
TBD. Optional if not required by the NVMe Transport (refer to section 3.9).		

3.1.2.2 Administrative Controller

3.1.2.2.1 Command Support

...

Figure 28: Administrative Controller – Admin Command Support

Command	Command Support Requirements ¹	Reference
...		
Keep Alive	NOTE-2M ²	5.18
...		
Notes: 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited 2. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory. Prohibited if the Keep Alive Timer feature is not supported (refer to section 3.9).		
...		

3.1.2.2.3 Features Support

...

Figure 30: Administrative Controller – Feature Support

Feature Name	Feature Support Requirements ¹	Logged in Persistent Event Log ¹
...		
Keep Alive Timer	O ^M TBD	O
...		
Notes: ... TBD. Optional if not required by the NVMe Transport (refer to section 3.9).		

3.1.2.3 Discovery Controller

...

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 3.9.TBDCleanup.

...

3.1.2.3.4 Features Support

...

Figure 34: Discovery Controller – Feature Support

Feature Name	Feature Support Requirements ¹	Logged in Persistent Event Log ¹
...		
Keep Alive Timer	OM TBD	O
...		
Notes:		
...		
TBD. For Discovery controllers that do not support explicit persistent connections, this Keep Alive Timer feature is prohibited.		

3.3 NVM Queue Models

3.3.3 Queueing Data Structures

3.3.3.2 Common Completion Queue Entry

3.3.3.2.1 Status Field Definition

3.3.3.2.1.1 Generic Command Status Definition

...

Figure 94: Status Code – Generic Command Status Values

Value	Description	I/O Command Set Specific	I/O Command Set(s) ¹
...			
1Ah	Keep Alive Timeout Invalid: The Keep Alive Timeout value specified is invalid. This may be due to an attempt to specify a value of 0h on a transport that requires the Keep Alive Timer feature to be enabled. This may be due to the value specified being too large for the associated NVMe Transport as defined in the NVMe Transport binding specification.	No	
...			

3.9 Keep Alive

The Keep Alive capability uses the Keep Alive tTimer as a watchdog timer intended to detect communication failures (e.g., transport failure, host failure, or controller failure) between a host and a controller. If the Keep Alive Timer feature is supported (i.e., the KAS field is set to a non-zero value (refer

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008-2023 NVM Express, Inc.

to Figure 275)), the controller shall support the Keep Alive command ~~a malfunctioning connection, controller, or host. The Keep Alive Timeout Interval is the period during which the Keep Alive Timer is activated.~~

<NOTE: target structure:

- Applicability: Transport dependency
- Configuring
- Activating
- Command Based Keep Alive
 - Controller
 - Host
- Traffic Based Keep Alive
 - Controller
 - Host
- Keep Alive Timeout Cleanup

>

The NVMe Transport binding specification for the associated NVMe Transport defines ~~for the associated NVMe Transport:~~

- the minimum Keep Alive Timeout value, if any;
- the maximum Keep Alive Timeout value, if any; and
- if ~~support for~~ the Keep Alive Timer feature is required to be supported and enabled.

NVMe Transports that do not detect a connection loss in a timely manner shall require that the Keep Alive Timer feature be supported and enabled.

3.9.TBDconfig Keep Alive Timer Configuration

A host configures the Keep Alive Timer feature by specifying a Keep Alive Timeout value in:

- the KATO field of a Fabric Connect command (refer to section 6.3); or
- the KATO field of a Set Features command specifying the Keep Alive Timer feature (refer to section 5.27.1.12).

If an NVMe Transport binding specification requires the use of the Keep Alive Timer feature, and a command attempts to disable the Keep Alive Timer by ~~setting~~ clearing the Keep Alive Timeout value to 0h, then the controller shall abort that command with a status code of Keep Alive Timeout Invalid and the Keep Alive Timeout value at the controller shall not be changed. ~~or~~ If a command attempts to set the Keep Alive Timeout value to a value that exceeds the maximum allowed by the associated NVMe Transport binding specification, ~~a status code of Keep Alive Timeout Invalid shall be returned~~ then the controller shall abort that command with a status code of Keep Alive Timeout Invalid and the Keep Alive Timeout value at the controller shall not be changed. If a command sets the Keep Alive Timeout value to a non-zero value that is less than the minimum supported by the NVMe Transport or less than the minimum supported by the specific implementation, then the controller sets the Keep Alive Timeout value to that minimum value.

As described in section 5.27.1.12, the controller rounds any Keep Alive Timeout value set by the host up to the nearest granularity as reported in the Keep Alive Support (KAS) field (refer to Figure 275). To retrieve the Keep Alive Timeout value being used by the controller, a host may issue a Get Features command for the Keep Alive Timer feature.

3.9.TBDactive Keep Alive Timer Activation

The Keep Alive Timer is active if:

- CC.EN is set to '1';
- CSTS.RDY is set to '1';

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008-2023 NVM Express, Inc.

- CC.SHN is cleared to '00b';
- CSTS.SHST is cleared to '00b'; and
- the Keep Alive Timer feature ~~has been~~ is enabled ~~with~~ as a result of the KATO field (refer to ~~section 5.27.1.12 and section 6.3~~) being set to a non-zero value (refer to section ~~3.9.TBD~~config).

Otherwise, the Keep Alive Timer is inactive, and a Keep Alive Timeout as described in sections ~~3.9.1 and 3.9.2~~ shall not occur. Activating an inactive Keep Alive ~~t~~Timer (e.g., a Set Features command successfully sets the Keep Alive Timeout value to a non-zero value from a value of 0h, or ~~, enabling a controller with the Keep Alive feature in use,~~ enabling a controller that supports NVMe over Fabrics where the Connect command specified a non-zero Keep Alive Timeout value (refer to ~~section 3.1.2.3 Figure 380~~)) shall initialize the Keep Alive ~~t~~Timer to the Keep Alive Timeout value.

While the Keep Alive Timer is active, the host should ensure that the Admin Submission Queue has space for a Keep Alive command.

<NOTE: Timer starting/expiring is moved to 3.9.1 and 3.9.2, where it is defined more precisely.>

~~A Keep Alive Timeout Interval on the controller starts when:~~

- ~~• a successful completion queue entry is posted for a Set Features command with Feature Identifier 0Fh and a non-zero KATO field.~~

~~A Keep Alive Timeout Interval on the host starts when:~~

- ~~• a Set Features command with Feature Identifier 0Fh and a non-zero Keep Alive Timeout (KATO) field is posted to the Admin submission queue; or~~
- ~~• a Keep Alive command was posted to the Admin submission queue.~~

~~Both on the host and the controller the Keep Alive Timeout Interval ends the time specified by the KATO field after the interval started (refer to Figure 341). A Keep Alive Timeout occurs when the Keep Alive Timer expires. The Keep Alive Timer expires:~~

- ~~• if the TBKAS bit is cleared to '0', at the end of the Keep Alive Timeout Interval and no Keep Alive Command was processed during the Keep Alive Timeout Interval (refer to section 3.9.2); and~~
- ~~• if the TBKAS bit is set to '1', at the end of the Keep Alive Timeout Interval and no Admin command or I/O command was processed during the Keep Alive Timeout Interval (refer to section 3.9.2).~~

<NOTE: the following paragraph is wrong for TBKA. For Command Based Keep Alive, it is redundant with text in that section>

~~The Keep Alive Timeout is the maximum time a connection remains established without processing a Keep Alive command. The Keep Alive timer in the controller expires when a Keep Alive command is not received within the Keep Alive Timeout interval.~~

<NOTE: Activating timer moved to 3.9.TBAactivate>

~~The Keep Alive timer is active if:~~

- ~~• CC.EN is set to '1';~~
- ~~• CSTS.RDY is set to '1';~~
- ~~• CC.SHN is cleared to '00b';~~
- ~~• CSTS.SHST is cleared to '00b'; and~~
- ~~• the Keep Alive Timer feature has been enabled with a KATO field (refer to section 5.27.1.12 and section 6.3) set to a non-zero value,~~

~~otherwise, the Keep Alive timer is inactive and a Keep Alive Timeout shall not occur. Activating an inactive Keep Alive timer (e.g., enabling a controller with the Keep Alive feature in use, enabling a controller that supports NVMe over Fabrics where the Connect command specified a non-zero Keep Alive Timeout (refer to section 3.1.2.3)) shall initialize the Keep Alive timer to the Keep Alive Timeout value.~~

<NOTE: end Activating timer moved to section 3.9.TBAactivate>

<NOTE: Host behavior is moved and revised to 3.9.1.2 and 3.9.2.2.>

~~The host may consider a Keep Alive Timeout to have occurred when the completion of the Keep Alive command is not received within the Keep Alive Timeout interval. The host is intended to send Keep Alive commands at a faster rate than the Keep Alive Timeout accounting for transport roundtrip times, transport delays, command execution times, and the Keep Alive Timer granularity.~~

<NOTE: Keep Alive Timer expiring text moved to end, 3.9.TBDcleanup>

~~If a Keep Alive Timer expires:~~

~~a) the controller shall:~~

- ~~• record an Error Information Log Entry with the status code Keep Alive Timeout Expired;~~
- ~~• stop processing commands;~~
- ~~• set the Controller Fatal Status (CSTS.CFS) bit to '1'; and~~
- ~~• for message-based NVMe Transports:~~
 - ~~○ terminate the NVMe Transport connection; and~~
 - ~~○ break the host to controller association;~~

~~and~~

~~b) the host assumes all outstanding commands are not completed and re-issues commands as appropriate.~~

~~For message-based NVMe Transports, after completing these steps, a controller may accept a Connect command (refer to section 6.3) for the Admin Queue from the same or another host in order to form a new association.~~

<NOTE: end Keep Alive Timer expiring text moved to end, 3.9.TBDcleanup>

~~The Keep Alive command restarts the timeout period; other commands have no effect on the timeout.~~

<NOTE: moved to the end of section 5.18> ~~The controller should process the Keep Alive command as soon as the command is fetched.~~

<NOTE: Transport info moved to beginning of section 3.9>

~~The NVMe Transport binding specification defines for the associated NVMe Transport:~~

- ~~• the minimum Keep Alive Timeout value;~~
- ~~• the maximum Keep Alive Timeout value; and~~
- ~~• if support for the Keep Alive feature is required.~~

~~NVMe Transports that do not detect a connection loss in a timely manner shall require that the Keep Alive feature be enabled. If a command attempts to disable the Keep Alive timer by setting the Keep Alive Timeout value to 0h or to a value that exceeds the maximum allowed by the associated NVMe Transport binding specification, a status code of Keep Alive Timeout Invalid shall be returned. If a command sets the Keep Alive Timeout value to a value that is less than the minimum supported by the NVMe Transport or less than the minimum supported by the specific implementation, then the controller sets the Keep Alive Timeout value to that minimum value.~~

<NOTE: end Transport info moved to beginning of section 3.9>

3.9.1 **Keep Alive** Command Based Keep Alive

For Command Based Keep Alive, the Keep Alive command is sent periodically from the host to the controller over the Admin Queue. The completion of the Keep Alive command indicates that the host and controller are able to communicate. For message-based transports, ~~t~~The Keep Alive Timeout is the maximum time ~~a connection~~ an association remains established without processing a Keep Alive command.

3.9.1.1 Command Based Keep Alive on the Controller

~~Keep Alive Command Based Keep Alive restricts the Keep Alive Timer on both the host and the controller to be restarted only upon the processing of a Keep Alive command. This~~ The controller is using Command Based Keep Alive ~~mode is in use~~ if the TBKAS bit is cleared to '0' in the CTRATT field in the Identify Controller data structure (refer to [Figure 275](#)).

<NOTE, from section 3.9>

~~A Keep Alive Timeout Interval on the controller starts when:~~

- ~~• a successful completion queue entry is posted for a Set Features command with Feature Identifier 0Fh and a non-zero KATO field.~~

~~Both on the host and the controller the Keep Alive Timeout Interval ends the time specified by the KATO field after the interval started (refer to Figure 341). A Keep Alive Timeout occurs when the Keep Alive Timer expires. The Keep Alive Timer expires:~~

- ~~• if the TBKAS bit is cleared to '0', at the end of the Keep Alive Timeout Interval and no Keep Alive Command was processed during the Keep Alive Timeout Interval (refer to section 3.9.2); and~~

~~The Keep Alive command restarts the timeout period~~

<NOTE, end from section 3.9>

For Command Based Keep Alive:

- The Keep Alive Timer in the controller starts when the Keep Alive Timer becomes active (refer to section 3.9.TBAActive).
- The Keep Alive Timer in the controller restarts if:
 - a Keep Alive command completes successfully; or
 - a Set Features command specifying the Keep Alive Timer feature and a non-zero KATO field (refer to section 5.27.1.12) completes successfully.
- The Keep Alive Timer in the controller expires if:
 - the Keep Alive Timer is active in the controller; and
 - time equal to the Keep Alive Timeout value has elapsed since the Keep Alive Timer was most recently started or restarted.

<NOTE: moved to the beginning of 3.9.1 as an introductory paragraph>~~The Keep Alive Timeout is the maximum time a connection remains established without processing a Keep Alive command. If the Keep Alive Timer in the controller expires and a Keep Alive command has not been processed within the Keep Alive Timeout Interval, then the controller may consider a Keep Alive Timeout to have occurred. Upon the occurrence of a Keep Alive Timeout, the controller shall perform the cleanup actions described in section 3.9.TBDcleanup.~~ <NOTE: this is not a technical change; this addition references cleanup actions consistently>

3.9.1.2 Command Based Keep Alive on the Host

<NOTE, from section 3.9>

~~A Keep Alive Timeout Interval on the host starts when:~~

- ~~• a Set Features command with Feature Identifier 0Fh and a non-zero Keep Alive Timeout (KATO) field is posted to the Admin-submission queue; or~~
- ~~• a Keep Alive command was posted to the Admin-submission queue.~~

~~Both on the host and the controller the Keep Alive Timeout Interval ends the time specified by the KATO field after the interval started (refer to Figure 341). A Keep Alive Timeout occurs when the Keep Alive Timer expires. The Keep Alive Timer expires:~~

- ~~• if the TBKAS bit is cleared to '0', at the end of the Keep Alive Timeout Interval and no Keep Alive Command was processed during the Keep Alive Timeout Interval (refer to section 3.9.2); and~~

~~...~~

~~The host may consider a Keep Alive Timeout to have occurred when the completion of the Keep Alive command is not received within the Keep Alive Timeout interval. The host is intended to send Keep Alive commands at a faster rate than the Keep Alive Timeout accounting for transport roundtrip times, transport delays, command execution times, and the Keep Alive Timer granularity.~~

<NOTE, end from section 3.9>

The host may use Command Based Keep Alive regardless of the Keep Alive mode used by the controller. If the Keep Alive Timer in the host expires and a completion of a Keep Alive command has not been received within the Keep Alive Timeout Interval, then the host may consider a Keep Alive Timeout to have occurred. To prevent the controller from detecting a Keep Alive Timeout during the use of Command Based Keep Alive on the host, the host should send Keep Alive commands at half of the time equal to the Keep Alive Timeout value to accounting for delays (e.g., transport roundtrip times, transport delays, command processing times, and the Keep Alive Timer granularity).

For Command Based Keep Alive:

- The Keep Alive Timer in the host starts or restarts if:
 - a Set Features command with Feature Identifier 0Fh and a non-zero Keep Alive Timeout (KATO) field was postsubmitted to the Admin submission queue; or
 - a Keep Alive command was postsubmitted to the Admin submission queue.
- The Keep Alive Timer in the host expires if:
 - time equal to the Keep Alive Timeout value has elapsed since the Keep Alive Timer was most recently started or restarted; and
 - a completion of a Keep Alive command has not been received with the Keep Alive Timeout Interval since the Keep Alive Timer was most recently started or restarted.

If the Keep Alive Timer in the host expires, then the host may consider a Keep Alive Timeout to have occurred. If a host detects a Keep Alive Timeout, the host should perform the actions described in section 3.9.TBDcleanup.

3.9.2 Traffic Based Keep Alive

Traffic Based Keep Alive (TBKAS) allows the host and controller to avoid a Keep Alive Timeout-restart the Traffic Based Keep Alive Timer in the presence of Admin or I/O command processing without sending Keep Alive commands.

3.9.2.1 Traffic Based Keep Alive on the Controller

The Controller support for Traffic Based Keep Alive is indicated by the TBKAS bit is indicated in the Controller Attributes in the Identify Controller data structure (refer to Figure 275). If the Controller does not support Traffic Based Keep Alive (i.e., the TBKAS bit is cleared to '0'), then the operation of the Keep Alive Timer feature is described in section 3.9.1.

The Traffic Based Keep Alive Timeout occurs if a connection remains established without processing an Admin command or an I/O command during the Keep Alive Timeout Interval. If an Admin command or an I/O command is processed within the Keep Alive Timeout Interval, then upon the expiration of the Keep Alive Timer, the Keep Alive Timer shall be restarted.

<NOTE, from section 3.9>

A Keep Alive Timeout Interval on the controller starts when:

- a successful completion queue entry is posted for a Set Features command with Feature Identifier 0Fh and a non-zero KATO field.

Both on the host and the controller the Keep Alive Timeout Interval ends the time specified by the KATO field after the interval started (refer to Figure 341). A Keep Alive Timeout occurs when the Keep Alive Timer expires. The Keep Alive Timer expires:

- if the TBKAS bit is set to '1', at the end of the Keep Alive Timeout Interval and no Admin command or I/O command was processed during the Keep Alive Timeout Interval (refer to section 3.9.2).

The Keep Alive command restarts the timeout period

<NOTE, end from section 3.9>

For Traffic Based Keep Alive:

- A Keep Alive Timeout Interval on the controller starts when the Keep Alive Timer becomes active (refer to section 3.9.TBDactive).

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008-2023 NVM Express, Inc.

- The controller may consider a Keep Alive Timeout to have occurred if: ~~<NOTE added colon>-no Admin command or no I/O command is submitted to the controller (as defined in section 3.4.4) within the Keep Alive Timeout Interval. If an Admin command or an I/O command is transferred to the Controller within the Keep Alive Timeout Interval, then upon the expiration of the Keep Alive Timer the controller shall restart the Keep Alive Timer.~~
 - the Keep Alive Timer is active;
 - time equal to the Keep Alive Timeout value has elapsed since the start of the most recent Keep Alive Timeout Interval; and
 - no Admin command or I/O command was fetched by the controller during the Keep Alive Timeout Interval.
- A Keep Alive Timeout Interval on the controller ends and:
 - a new Keep Alive Timeout Interval does not start if:
 - a Keep Alive Timeout occurs; or
 - the Keep Alive Timer becomes inactive (refer to section 3.9.TBDactive).
 - a new Keep Alive Timeout Interval starts if:
 - a Keep Alive command completes successfully;
 - a Set Features command specifying the Keep Alive Timer feature and a non-zero KATO field (refer to section 5.27.1.12) completes successfully; or
 - time equal to the Keep Alive Timeout value has elapsed since the start of the Keep Alive Timeout Interval and a Keep Alive Timeout did not occur during the Keep Alive Timeout Interval (e.g., an Admin command or an I/O command was fetched by the controller during the Keep Alive Timeout Interval).

Upon the occurrence of a Keep Alive Timeout, the controller shall perform the cleanup actions described in section 3.9.TBDcleanup.

3.9.2.2 Traffic Based Keep Alive on the Host

<NOTE, from section 3.9>

A Keep Alive Timeout Interval on the host starts when:

- ~~a Set Features command with Feature Identifier 0Fh and a non-zero Keep Alive Timeout (KATO) field is posted to the Admin submission queue; or~~
- ~~a Keep Alive command was posted to the Admin submission queue.~~

~~Both on the host and the controller the Keep Alive Timeout Interval ends the time specified by the KATO field after the interval started (refer to Figure 341). A Keep Alive Timeout occurs when the Keep Alive Timer expires. The Keep Alive Timer expires:~~

- ~~if the TBKAS bit is set to '1', at the end of the Keep Alive Timeout Interval and no Admin command or I/O command was processed during the Keep Alive Timeout Interval (refer to section 3.9.2).~~

...

~~The host may consider a Keep Alive Timeout to have occurred when the completion of the Keep Alive command is not received within the Keep Alive Timeout interval. The host is intended to send Keep Alive commands at a faster rate than the Keep Alive Timeout accounting for transport roundtrip times, transport delays, command execution times, and the Keep Alive Timer granularity.~~

<NOTE, end from section 3.9>

The host may use Traffic Based Keep Alive only if the controller is using Traffic Based Keep Alive. ~~The host may consider a Traffic Based Keep Alive Timeout to have occurred if the host does not receive a completion of any Admin command or any I/O command within the Keep Alive Timeout Interval. If an Admin command or an I/O command is completed within the Keep Alive Timeout Interval, then upon expiration of the Keep Alive Timer, the host shall restart the Keep Alive Timer.~~ The host should check for a command completion queue entry for any Admin commands and I/O commands at half of the time equal to the Keep Alive Timeout value to accounting for delays (e.g., transport roundtrip times, transport delays, command processing times, and the Keep Alive Timer granularity). To prevent the controller from detecting a Keep Alive Timeout during the use of Traffic Based Keep Alive on the host, if no Admin command and no I/O command is sent to the controller during half of the Keep Alive Timeout Interval, the host should send a Keep Alive command.

For Traffic Based Keep Alive:

- The Keep Alive Timer in the host starts or restarts if:
 - a Set Features command with Feature Identifier 0Fh and a non-zero Keep Alive Timeout (KATO) field ~~was post~~submitted to the Admin submission queue; ~~or~~
 - a Keep Alive command was ~~post~~submitted to the Admin submission queue; ~~or~~
 - time equal to the Keep Alive Timeout value has elapsed since the Keep Alive Timer was most recently started or restarted, and a completion was received for an Admin command or an I/O command since the Keep Alive Timer was most recently started or restarted.
- The Keep Alive Timer in the host expires if:
 - time equal to the Keep Alive Timeout value has elapsed since the Keep Alive Timer was most recently started or restarted; and
 - the host ~~does~~ not receive a completion of any Admin command or any I/O command since the Keep Alive Timer was most recently started or restarted.

If the Keep Alive Timer in the host expires, ~~T~~the host may consider a ~~Traffic-Based~~ Keep Alive Timeout to have occurred. If a host detects a Keep Alive Timeout, the host should perform the actions described in section **3.9.TBDCleanup**.

3.9.TBDCleanup Keep Alive Timeout Cleanup

If a Keep Alive Timer ~~expires~~out is detected:

- a) by the controller, the controller shall:
 - record an Error Information Log Entry with the status code Keep Alive Timeout Expired;
 - stop processing commands;
 - set the Controller Fatal Status (CSTS.CFS) bit to '1'; and
 - for message-based NVMe Transports:
 - terminate the NVMe Transport connections for this association; and
 - break the host to controller association;and
- b) by the host, the host assumes all outstanding commands are not completed and re-issues commands as appropriate.

For message-based NVMe Transports, after completing these steps, a controller may accept a Connect command (refer to section **6.3**) for the Admin Queue from the same or another host in order to form a new association.

5 Admin Command Set

5.17 Identify command

5.17.2 Identify Data Structures

5.17.2.1 Identify Controller Data Structure (CNS 01h)

...

Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O ¹	Admin ¹	Disc ¹	Description
...				

Bytes	I/O ¹	Admin ¹	Disc ¹	Description								
99:96	M	M	R	Controller Attributes (CTRATT): This field indicates attributes of the controller.								
				<table><tr><th>Bits</th><th>Description</th></tr><tr><td>...</td><td></td></tr><tr><td>6</td><td>Traffic Based Keep Alive Support (TBKAS): If set to '1', then the controller supports restarting the Keep Alive Timer if an Admin command or an I/O command is processed during the Keep Alive Timeout Interval uses Traffic Based Keep Alive (refer to section 3.9.2.1<NOTE: section modification>). If cleared to '0', then the controller supports restarting the Keep Alive Timer only if a Keep Alive command is processed (refer to section 3.9.1) does not use Traffic Based Keep Alive.</td></tr><tr><td>...</td><td></td></tr></table>	Bits	Description	...		6	Traffic Based Keep Alive Support (TBKAS): If set to '1', then the controller supports restarting the Keep Alive Timer if an Admin command or an I/O command is processed during the Keep Alive Timeout Interval uses Traffic Based Keep Alive (refer to section 3.9.2.1<NOTE: section modification>). If cleared to '0', then the controller supports restarting the Keep Alive Timer only if a Keep Alive command is processed (refer to section 3.9.1) does not use Traffic Based Keep Alive.	...	
				Bits	Description							
				...								
6	Traffic Based Keep Alive Support (TBKAS): If set to '1', then the controller supports restarting the Keep Alive Timer if an Admin command or an I/O command is processed during the Keep Alive Timeout Interval uses Traffic Based Keep Alive (refer to section 3.9.2.1<NOTE: section modification>). If cleared to '0', then the controller supports restarting the Keep Alive Timer only if a Keep Alive command is processed (refer to section 3.9.1) does not use Traffic Based Keep Alive.											
...												
...												
...												
321:320	M	M	RO	Keep Alive Support (KAS): This field indicates the granularity of the Keep Alive Timer KATO field in 100 millisecond units (refer to section 3.9.5.27.1.12). If this field is cleared to 0h, then the Keep Alive Timer feature is not supported. The Keep Alive Timer feature is used by the Keep Alive capability shall be supported for NVMe over Fabrics implementations as described in section 3.9.								
...												

5.18 Keep Alive command

The Keep Alive command (refer to section 5.27.1.12) and associated functionality is used by the host to ~~determine that the controller is operational and used by the controller to determine that the host is operational. The host and controller are operational when each is accessible and able to issue or process commands. The controller indicates the granularity of the Keep Alive Timer in the KAS field in the Identify Controller data structure (refer to Figure 275).~~

If a Keep Alive Timeout has been enabled on the Admin Queue, the Keep Alive Timer is restarted when:

- ~~• a Keep Alive command (refer to section 3.9.1) is processed; or~~
- ~~• at the end of the Keep Alive Timeout (refer to section 3.9.2) when TBKAS is set to '1' and an Admin command or an I/O command is processed during the Keep Alive Timeout Interval and the Keep Alive Timer feature (refer to section 5.27.1.12) are used by the Keep Alive capability (refer to section 3.9).~~ The controller should process the Keep Alive command as soon as the command is fetched. If the Keep Alive Timer is active (refer to section 3.9.TBDactive), then upon processing a Keep Alive command, the controller restarts the Keep Alive Timer.

All command specific fields are reserved.

5.18.1 Command Completion

Upon completion of the Keep Alive command, the controller shall post a completion queue entry to the Admin Completion Queue indicating the status for the command.

5.27 Set Features command

5.27.1 Feature Specific Information

5.27.1.12 Keep Alive Timer (Feature Identifier 0Fh)

This Feature controls the Keep Alive Timer. Refer to section 3.9 for Keep Alive details. The attributes are specified in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 341 are returned in Dword 0 of the completion queue entry for that command.

Figure 341: Keep Alive Timer – Command Dword 11

Bits	Description
31:00	<p>Keep Alive Timeout (KATO): This field specifies the timeout value for the Keep Alive Timer feature in milliseconds. The controller rounds up the value specified to the granularity indicated in the KAS field in the Identify Controller data structure. If cleared to 0h, then the Keep Alive Timer is disabled.</p> <p>The default value for this field is 0h for NVMe transports that do not require use of the Keep Alive Timer feature (e.g., NVMe over PCIe). For NVMe transports that require use of the Keep Alive Timer feature (e.g., RDMA and TCP), the default value for this field is 1D4C0h (i.e., 120,000 milliseconds or 2 minutes) rounded up to the granularity indicated in the KAS field.</p> <p>Refer to the applicable NVMe Transport Binding specification for details.</p>

6 Fabrics Command Set

6.3 Connect Command and Response

...

Figure 380: Connect Command – Submission Queue Entry

Bytes	Description
...	
51:48	<p>Keep Alive Timeout (KATO): In the Connect command for the Admin Queue, this field has the same definition as the Keep Alive Timeout (KATO) field of the {Keep Alive Timer feature} defined in (refer to section 5.27.1.12). Upon successful completion of the Connect command, the controller shall enable and activate set the KATO field in the Keep Alive Timer feature. as described in Refer to section 3.9. TBDactive for a description of activating the Keep Alive Timer.</p> <p>In the Connect command for an I/O Queue, this field is reserved.</p>
...	

Description of NVM Express TCP Transport Specification, Rev 1.0c changes

3 Transport Binding

3.4 Keep Alive Model

The NVMe/TCP Transport requires the use of the Keep Alive [Timer](#) feature (refer to [the Keep Alive section 7.42](#) in the NVM Express Base Specification). The NVMe/TCP Transport does not impose any limitations on the minimum and maximum Keep Alive Timeout value. The minimum should be set large enough to account for any transient fabric interconnect failures between the host and controller.

TCP level keep alive functionality is not prohibited but it is recommended that the TCP level keep alive timeout is set to a higher value than the NVMe Keep Alive Timeout to avoid conflicting policies.

3.5 Error Handling Model

3.5.1 Transport Error Handling Model

...

When a non-fatal transport error is detected, affected commands are completed with a Transient Transport Error status code (refer to [the Status Code – Generic Command Status Values Figure 426](#) in the NVM Express Base Specification).

Description of NVM Express RDMA Transport Specification, Rev 1.0c changes

3 Transport Binding

3.4 Keep Alive Model

Keep Alive functionality is not supported by all RDMA Provider Types at the RDMA Transport layer. As a result, the RDMA Transport requires the use of the Keep Alive ~~Timer~~ Feature (refer to the Keep Alive section in the NVM Express Base Specification). It is recommended that any RDMA provider level functionality be disabled to avoid redundant and conflicting policies.

The RDMA Transport does not impose any limitations on the minimum and maximum Keep Alive Timeout value. The minimum should be set large enough to account for any transient fabric interconnect failures between the host and controller.