NVMe® over FC: Deep Dive in Protocol, Architecture and Use Cases
Today’s Speakers

David Rodgers  
Technical Business Development Manager  
EXFO

Kamal Bakshi  
Technical Director  
Cisco

Nishant Lodha  
Director, Emerging Technologies  
Marvell Semiconductor

Mark Jones  
Director Technical Marketing, Broadcom  
FCIA Board of Directors
“The Fibre Channel Industry Association (FCIA) is a mutual benefit, non-profit, international organization of manufacturers, system integrators, developers, vendors, industry professionals, and end users.”
About the Fibre Channel Industry Association (FCIA)

25+ Years
Promoting Fibre Channel Technology

Industry Leading
Member Companies

160M FC Ports
Shipped Since 2001
2023 Fibre Channel Market Milestones

- **$50B FC Revenue**: Total Revenue of Adapter and Switch products
- **35M+ FC Ports In Service**: Adapter and switch ports estimated in service
- **160M FC Total Ports Shipped**: Adapter and switch ports

*Quillin Research 2023 SAN forecast – cumulative adapter and switch ports from 1998 through 2022
Agenda

• NVMe® Overview
• NVMe/FC Architecture
• Adoption and Virtualization Use Cases
• Performance
NVMe Overview & NVMe/FC Architecture

Kamal Bakshi
Technical Director
Cisco
NVMe® Nomenclature & Trademarks

NVM Express® (Incorporated in March 2014)
NVM Express Logo®

NVMe®
NVMe®/PCIe® specification
NVM Express® Management Interface
NVMe-MI™

NVM Express® over Fabrics
NVMe-oF™

NVMe®/TCP protocol
NVMe®/RDMA protocol
   NVMe®/IB transport
   NVMe®/RoCE transport
   NVMe®/IWARP transport

NVMe®/FC protocol
(FC-NVMe-2) INCITS/T11 standard
This standard describes the frame format and protocol definitions required to transfer commands and data between a NVMe host and NVMe Express subsystem using the Fibre Channel family of standards.

The appropriate registered trademark designations (™ or ®) must be applied.
50,000 feet view of NVMe

1970
- Channel Subsystem
- CKD CCW
- IBM
- 1964-Bus & Tag-Copper 1990-ESCON-Fiber
- 1998-FICON-FC
- (DASD)Disk Access Storage Device

1980
- Channel Protocol
- SCSI
- Server
- SCSI Disk

2000
- Protocol
- ATA
- PC
- SATA
- SATA Disk

2010/20
- Host
- NVMe
- NVMe-of FC/RDMA/TCP
- NVMe-Flash
- NVMe -All Flash Arrays
50,000 feet view of NVMe

**1998**
- **IBM** Channel Subsystem
- **FICON** Fibre Channel
- **CKD**
- **CCW**
- **CU**
- **FC-SB**
- **Non-NVMe Flash DS8900**

**1994**
- **SCSI** Server
- **SCSI** Fibre Channel
- **NVMe**

**2011**
- **NVMe-Flash**
- **PCIe Bus**

**2016**
- **NVMe-Flash**
- **Fibre Channel**
- **NVMe - All Flash Arrays**

Kamal Bakshi
NVMe 2.0 Journey

IBM/PC
- 1984: IBM-PC AT
- 1986: IBM-PC XT

PC/Bus
- 1984: ISA
- 1986: EISA
- 1998: PATA
- 2003: AHCI
- 2008: SATA
- 2011: SATA-3.5
- 2021: SATA-3.5

Server
- 1984: IBM
- 1986: PC AT
- 1994: IBM
- 1998: MMIO
- 2003: PCI
- 2008: PCIe
- 2011: PCle-6
- 2021: PCIe-6

SAN
- 1994: 1.06Gb/s
- 1998: 1.5Gb/s
- 2003: x4 12Gb/s
- 2011: x4 256Gb/s

IB
- 1994: SAS
- 1998: SAS
- 2003: SAS
- 2011: SAS-5
- 2021: SAS-5

Flash
- 1994: Flash
- 1998: Flash
- 2003: Flash
- 2011: Flash
- 2021: Flash

NVMe Specifications
- NVMe/PCIe
- NVMe/IB
- NVMe/Trusted Computing
- NVMe/FC
- NVMe/TCP
- NVMe-1.0
- NVMe-2.0
- NVMe-oF
- NVMe-FC (FC-NVMe-2*)
- NVMe-IB, NVMe/RoCEv2

*INCITS/T11
NVMe 2.0 Specifications (4 Categories, 10 Docs., 1032 pages)

https://nvmexpress.org/specifications/

I-Mgmt.

(9) Management Interface Specification
(10) NVMe Network Boot Specification

II-Base Specification
(1) NVMe Base Specification + Admin Cmd, Fabric Cmd, Common I/O Cmd

III-NVMe Specific I/O command sets
(2) NVM Command Sets
(3) Zoned NS Command Sets
(4) Key Value Command Sets

IV-Transport Mapping Specifications (PCIe, IB, Enet, FC)

Message/Capsule Based Model
Shared Memory Based Model
Fibre Channel

(5) NVMe/PCIe
(6) RDMA Spec.
(7) FC-NVMe-2 (T11) + FC-FS-6 (T11)
(8) NVMe/TCP
(9) NVMe/TCP
- Ethernet
  - iSCSI
  - FCoE
  - FCIP
  - iFCP
(NVMe/IB, RoCEv2)
- RDMA
  - SRP
  - iSER
(NVMe/FC)
- Fibre Channel
  - FCP/SCSI

(NVMe/PCIe)
(NVMe/IB)
(NVMe/RoCEv2)

Fibre Channel
NVMe SSD (Local Storage)

NVMe
- New Block Storage Protocol for Flash
- Maps directly into PCIe
- Replaces SCSI commands
- Transport mapping for RDMA/FC/TCP

Admin Command
- Create/Delete I/O SQ
- Create/Delete I/O CQ
- Get Log Page
- Identify
- Abort
- Set/Get Feature
- Async. Event Request

I/O Command
- Read/Write
- Flush

Fabric Command
- Connect/Disconnect
- Set/Get Property

NVMe/PCIe SSD

M.2 form factor

Kamal Bakshi

NVMe SSD (Local Storage)

NVMe
- New Block Storage Protocol for Flash
- Maps directly into PCIe
- Replaces SCSI commands
- Transport mapping for RDMA/FC/TCP

Admin Command
- Create/Delete I/O SQ
- Create/Delete I/O CQ
- Get Log Page
- Identify
- Abort
- Set/Get Feature
- Async. Event Request

I/O Command
- Read/Write
- Flush

Fabric Command
- Connect/Disconnect
- Set/Get Property

NVMe/PCIe SSD

M.2 form factor
FC-NVMe (Remote Storage)
FC-NVMe (Remote Storage)
FC-NVMe (NVMe Subsystem)

-NVMe Subsystem consists of multiple CNTLs
-Controllers provide access to NS via SQ/CQ
-Subsystem Port (Port-ID) is a protocol interface between an NVM subsystem & host
FC-NVMe (FC Mapping Abstractions)

NVMe Host Submits a NVMe_Write command as SQE (Submission Queue Entry)
Data pointed by the Host SGL is placed in a Data Series and command is passed to NVMe-FC layer
The Host NVMe-FC layer specifies the NVMe-FC association with the NVMe controller.
The Host NVMe-FC layer maintains a mapping of Host queues (NVMe-oF) to the NVMe controller’s NVMe queues (SQ/CQ) via connection IDs.
Upon receiving the SQE command NVMe_Port allocates XID for the NVMe-FC I/O operation and associates the NVMe command in the SQE to the Exchange. All NVMe IUs for the NVMe-FC I/O operation are transmitted as part of this Exchange.
The initiator NVMe_Port transmits the NVMe_CMND IU payload to start the NVMe-FC I/O operation.
FC-NVMe (HBA/MSIx Interrupts)
FC-NVMe (Discovery Services Subsystem)

- Discover I/O Controllers subsystem name
- Discover Multiple Paths to Subsystems
- Discover Static I/O Controllers
- Manage Async. Event Notifications

Host Bus Adapter

Host Software

Fabric

Host
FC-NVMe (Discovery Services Subsystem)

#1 FC Name Server
identifies the FC-NVMe Ports
including that offers Discovery Services

#2 Discovery Controller
identifies the other NVMe Subsystems
I/O CNTL, Discovery CNTL

#3 I/O Controller
Identify (CNS 02)
Active Namespaces

Host Software
Host Bus Adapter
Host

Storage Array
**FC-NVMe Protocol Flows** (FLOGI)

- **Flogi (Fabric Login)**
  - FCID is assigned
  - B2B are initialized
FC-NVMe Protocol Flows (PLOGI)

Port Logins
- Name Server Login - Registration
- Fabric Controller - SCN

Get ID_FF (FC4 Features Support)
- Type 28/NVMeoFC
- Feature 04/Discovery Services
FC-NVMe Protocol Flows (PRLI)

PRLI NVMeoFC
Service Parameter Page

Service Parameter
• Initiator FC-NVMe
• Target FC-NVMe
  - Discovery Svc.
  - SLER
Communication relationship between a particular controller and a particular host that encompasses the Admin Queue and all I/O Queues of that controller.
Host sends "Create Association" NVMe_LS to the Discovery Service subsystem WKA NQN

nqn.2014-08.org.nvmeexpress.discovery
FC-NVMe (Create Association)

Command (NVMe Create Association)

Controller ID (Dynamic)

Admin Queue depth

Host NQN
Controller Accepts the create association & assigns the “Association ID” and “Connection ID” that would represents the Admin queue 0

Host Software

Host Bus Adapter

FC-NVMe

Accept Create association Descriptor (Payload)
- Association ID#
- Connection ID#
FC-NVMe (Accept Create Association)

Accept

NVMe Association ID

NVMe Connection ID
FC-NVMe Protocol Flows (Connect Command SQE)

Host sends a NVMe Fabric command “Connect” to create the Admin SQ/CQ at the controller.

- Queue ID = 0 (Admin queue)
- Queue Size
- Keep Alive Timeout (KATO)

Note: Discovery CNTL doesn’t have I/O queues.
**FC-NVMe (Connect)**

Fabric Command = Connect

Queue ID = 0 (Admin)

default queue size = 32
FC-NVMe Protocol Flows (Connect Response CQE)

If a connection is established, then the Controller ID allocated to the host is returned.

Status code - Success (CNTL ID)
FC-NVMe Protocol Flows (Get Property CC)

"Get" command’s offset pointed to CC - Controller Configuration buffer
Enable the controller
Set “EN” bit to “1”
FC-NVMe Protocol Flows (Set Property CSTS.RDY)

Set the Controller Status to Ready
Now Controller can start accepting the commands

Set Property
CSTS.RDY
Host sends **Get Log Page-70** command to retrieve the known NVMe Subsystems within the storage target.
The Discovery Log Page provides an inventory of NVM subsystems with which a host may attempt to form an association.

The Discovery Log Page provides an inventory of NVM subsystems with which a host may attempt to form an association.
FC-NVMe Protocol Flows (Create Association with I/O CNTL)

NVMe Admin Queue Association
-LS_CASS
-NVMe Connect (queue #0)

Host Bus Adapter
-LS_CASS
-NVMe Connect

Host Software

Host
FC-NVMe Protocol Flows

(I/O CNTL Ready to accept commands)

Host Software

NVMe OF 0-Admin

Host Bus Adapter

NVMe Association ID

Connection ID0

Get/Set Properties

Get Property CC (Controller Config)
Set Property CC.EN (Enable Controller)
Set Property CSTS.RDY (Controller Ready)
FC-NVMe Protocol Flows (Create I/O Queues)

NVMe I/O Queue #1 Connection
LS_CIOC (queue id #1 / size)
“NVMe Connect” uses Admin queue

Host Bus Adapter
FC-NVMe
NVMe-OF 0-Admin
1-I/O queue
Host Software
Host

Storage Array
CPU
PCIe
Memory
DDR
PCIe Register
BAR Address
MSIx space
NVMe/PCIe Lanes
PCIe
Storage Array
SSD
NVMe Subsystem-NQN#1
NVMe Page
70
Properties Buffers Device Memory
Controller Cntl Flash Cntl NAND Flash FTL
Network Interface
MAC
Port-ID
FC Fabric
Host Software

FC-NVMe Protocol Flows

Create I/O Queues
Admin 0 SQ/CQ
NVM SQ/CQ
I/O Cntl#8 NS NSID
NVMe Subsystem Discovery Cntl
NVMe Subsystem Discovery Services
FC-NVMe
Host Log Page
NVMe Association ID
Connection ID0
Connection ID1
-LS_CIOC -NVMe Connect

FC-

NVMe Subsystem-NQN#1
Properties Buffers Device Memory
Controller Cntl Flash Cntl NAND Flash FTL
Network Interface
MAC
Port-ID
FC Fabric
Host Software

FC-NVMe Protocol Flows

Create I/O Queues
Admin 0 SQ/CQ
NVM SQ/CQ
I/O Cntl#8 NS NSID
NVMe Subsystem Discovery Cntl
NVMe Subsystem Discovery Services
FC-NVMe
Host Log Page
NVMe Association ID
Connection ID0
Connection ID1
-LS_CIOC -NVMe Connect

FC-

NVMe Subsystem-NQN#1
Properties Buffers Device Memory
Controller Cntl Flash Cntl NAND Flash FTL
Network Interface
MAC
Port-ID
FC Fabric
Host Software

FC-NVMe Protocol Flows

Create I/O Queues
Admin 0 SQ/CQ
NVM SQ/CQ
I/O Cntl#8 NS NSID
NVMe Subsystem Discovery Cntl
NVMe Subsystem Discovery Services
FC-NVMe
Host Log Page
NVMe Association ID
Connection ID0
Connection ID1
-LS_CIOC -NVMe Connect

FC-
FC-NVMe Protocol Flows (NVMe Identify CNS 02)

Get a list of “Active Namespaces”
FC-NVMe Protocol Flows (NVMe Write)

NVMe “Write” command uses I/O queues to transfer data.

Host Software

Host Bus Adapter

NVMe OF 0-Admin

1-I/O queue

Host

NVMe Write

FC-NVMe

NVMe Association ID

Connection ID1

NVMe Subsystem-NQN#1

NVMe Write

FC Fabric

Host Software
# FC-NVMe (Read command)

## Table

<table>
<thead>
<tr>
<th>Index</th>
<th>Hex</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00</td>
<td>SOF = SOF0;</td>
</tr>
<tr>
<td>00001</td>
<td>00</td>
<td>FC0 = FC0;</td>
</tr>
<tr>
<td>00002</td>
<td>00</td>
<td>NVMe-FC 0;</td>
</tr>
<tr>
<td>00003</td>
<td>00</td>
<td>CMD = CMD0;</td>
</tr>
<tr>
<td>00004</td>
<td>00</td>
<td>Payload 0;</td>
</tr>
<tr>
<td>00005</td>
<td>00</td>
<td>CRC = CRC0;</td>
</tr>
<tr>
<td>00006</td>
<td>00</td>
<td>EOF = EOF0;</td>
</tr>
</tbody>
</table>

## Diagram

- NVMe-CMD “Read”
- Connection-ID
- NSID
- SLB
- NLB

---

Kanu Balaji

FCIA
FREE CHANNEL INDUSTRY ASSOCIATION

48
FC-NVMe Architecture
Summary FC-NVMe

I
Connect to NVMe Cntl.

II
Create NVMe Queues

III
Discover Namespaces

IV
Start NVMe I/O operation
Summary FC-NVMe

I  Connect to NVMe Cntl.

II Create NVMe Queues

III Discover Namespaces

IV Start NVMe I/O operation
Summary FC-NVMe

I. Connect to NVMe Cntl.

II. Create NVMe Queues

III. Discover Namespaces

IV. Start NVMe I/O operation

- Get Controller’s details (NVMe Identify-01, 1C)
- Get Active NS List (NVMe Identify-02)
- Host issues Write Command (NVMe Write)
### FC-NVMe mapping - Command IU

**R_CTL (Routing)**

<table>
<thead>
<tr>
<th>To Target Port</th>
<th>To Initiator Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 NVMe_DATA</td>
<td>01 NVMe_DATA</td>
</tr>
<tr>
<td>03 NVMe_CONF</td>
<td>05 NVMe_XFER_RDY</td>
</tr>
<tr>
<td>06 NVMe_CMND</td>
<td>07 NVMe_RSP</td>
</tr>
<tr>
<td>09 NVMe_SR</td>
<td>08 NVMe_ERSP</td>
</tr>
<tr>
<td>(Seq. Retry)</td>
<td>0A NVMe_SR_RSP</td>
</tr>
</tbody>
</table>

**FC ID = (28) NVMe/FC**

#### NVMe_CMND IU

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_CTL(06)</td>
<td>D_ID</td>
</tr>
<tr>
<td>CS_CTL/Pri</td>
<td>S_ID</td>
</tr>
<tr>
<td>TYPE (8)</td>
<td>F_CTL</td>
</tr>
<tr>
<td>SEQ_ID</td>
<td>DF_CTL</td>
</tr>
<tr>
<td>SEQ_CNT</td>
<td></td>
</tr>
<tr>
<td>OX_ID</td>
<td>RX_ID</td>
</tr>
</tbody>
</table>

**Format ID (FD)**

- 0001b for Admin queue
- 1xxx for I/O queue

- xxx = CSS

**Flag**

- 0 Write, 1 Read

**Connection ID**

- Host queues mapping to the Controller’s NVMe queues

**DPS**

- Data Protection Type Setting

**LBADS**

- LBA Data Size

**MS**

- Meta Data Size

**NVMe SQE (64 bytes)**
FC-NVMe mapping - Read Command IU

NVMe-Read SQE
- Opcode (02) Read
- CID Command ID
- NSID Namespace ID
- SGL Descriptor
- SLBA Starting LBA
- NLB Number of LBs

NVMe_Transport
- Type (05)
- Sub Type(A)
- Length of Data Blocks(0)

Transport SGL

SGL

NVMe_CMND IU
- R_CTL(06)
- CS_CTL/Pri
- TYPE (8)
- SEQ_ID
- DF_CTL
- SEQ_CNT
- OX_ID
- RX_ID

Parameter
- Format ID (FD)
- FC ID (28)
- CMND IU Length
- Reserved
- Category
- Flags

Connection ID
- Command Sequence Number
- Data Length

NVMe_SQE (64 bytes)
- DPS
- LBADS
- MS
- Reserved
FC-NVMe mapping - Data Transfer

The start of the range is indicated by the Parameter field in the first frame of the Sequence. Relative offset value multiple of x4.

Data Series:
- Each frame in the Sequence is a continually increasing portion of the Data Series range.
- The length of the range is the Sequence payload length.
- If more than one NVMe_DATA IU is used to transfer the data, the relative offset value in the Parameter field is used to ensure that the NVM data is reassembled in the proper order.

NVMe Data is transferred as FCP Data.

---

<table>
<thead>
<tr>
<th>Parameter (offset)</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_CTL(01)</td>
<td></td>
</tr>
<tr>
<td>CS_CTL/Pri</td>
<td></td>
</tr>
<tr>
<td>TYPE (8)</td>
<td></td>
</tr>
<tr>
<td>SEQ_ID</td>
<td></td>
</tr>
<tr>
<td>DF_CTL</td>
<td></td>
</tr>
<tr>
<td>SEQ_CNT</td>
<td></td>
</tr>
<tr>
<td>OX_ID</td>
<td></td>
</tr>
<tr>
<td>RX_ID</td>
<td></td>
</tr>
</tbody>
</table>
NVMe Advantage#1: Higher Throughput/IOPS

Legacy SATA

SATAexpress 3.2

SAS-4 SSD

PCle/NVMe-SSD
NVMe Advantage#2: Lower Tail Latency (ZNS SSD)

Zoned Namespaces

By dividing an NVMe namespace into zones, which are required to be sequentially written, ZNS offers essential benefits to hyper-scale organizations, all-flash array vendors and large storage-system vendors wishing to take advantage of storage devices optimized for sequential write workloads. ZNS reduces device-side write amplification, over-provisioning and DRAM while improving tail latency, throughput and drive capacity.

(FDP* Flexible Data Placement – TP4146)
**NVMe Advantage#3: Error Detection & Recovery (SLER)**

FC-NVMe-2 (Sequence Level Error Recovery)

---

**Long Running Command**

<table>
<thead>
<tr>
<th>Host</th>
<th>Initiator</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>NVMe_CMND</td>
<td>FLUSH_CMND</td>
</tr>
<tr>
<td>FLUSH_TOV1</td>
<td>FLUSH_CMND</td>
<td>FLUSH_RSP2</td>
</tr>
</tbody>
</table>

1. FLUSH timeout occurs after transmitting the NVMe_CMND and a FLUSH BLS is sent to determine the status of the Exchange.

2. The FLUSH_RSP indicates the Exchange is open and the target NVMe_Port holds Sequence Initiative.

No error recovery is required.
Lost Command

1. FLUSH_CMND indicates the state of the Exchange;
2. transmit a FLUSH_RSP indicating the state of the Exchange; and
3. wait for an FC-4 specific event before further processing of the Exchange.

HT: Halt Transmission bit

- HT='1' indicates a FLUSH_CMND is sent to determine the status of the Exchange.
- HT='0' indicates the Exchange is not open at the target, and the Exchange is closed.

*1 FLUSH timeout occurs after transmitting the NVMe_CMND and a FLUSH BLS is sent to determine the status of the Exchange.

*2 The FLUSH_RSP indicates the Exchange is not open at the target, and the Exchange is closed.

*3 The NVMe_CMND IU is retransmitted using the same OX_ID, SLER qualifier, and CSN.
**NVMe Advantage#3: Error Detection & Recovery (SLER)**

**FC-NVMe-2 (Sequence Level Error Recovery)**

---

### Lost Response

<table>
<thead>
<tr>
<th>Host</th>
<th>Initiator</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>NVMe_CMND</td>
<td>NVMe_RSP</td>
</tr>
<tr>
<td></td>
<td>FLUSH_CMND</td>
<td>NVMe_RSP</td>
</tr>
<tr>
<td></td>
<td>FLUSH_RSP^1</td>
<td>NVMe_RSP^2</td>
</tr>
<tr>
<td></td>
<td>NVMe_SR^3</td>
<td>NVMe_SR_RSP</td>
</tr>
</tbody>
</table>

- *1 FLUSH timeout occurs after transmitting the NVMe_CMND and a FLUSH BLS is sent to determine the status of the Exchange.
- *2 The FLUSH_RSP indicates the initiator NVMe_Port holds Sequence Initiative and the Exchange is open.
- *3 The initiator NVMe_Port transmits an NVMe_SR IU specifying the NVMe_RSP be resent.

---

**NVMe_SR**

<table>
<thead>
<tr>
<th>Word</th>
<th>Bit 1</th>
<th>Bit 2</th>
<th>Bit 3</th>
<th>Bit 4</th>
<th>Bit 5</th>
<th>Bit 6</th>
<th>Bit 7</th>
<th>Bit 8</th>
<th>Bit 9</th>
<th>Bit 10</th>
<th>Bit 11</th>
<th>Bit 12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opcode (0xh)</td>
<td>Reserved</td>
<td>Retry R_CTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>FC ID (28h)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The NVMe_SR IU contains a request for Sequence retransmission from the initiator NVMe_Port to the target NVMe_Port to recover from an Exchange error.

**NVMe_SR_RSP**

<table>
<thead>
<tr>
<th>Word</th>
<th>Bit 1</th>
<th>Bit 2</th>
<th>Bit 3</th>
<th>Bit 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>FC ID (28h)</td>
<td>Opcode (0xh)</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

00h: Accepted  
01h: Invalid FC ID  
02h: Invalid Opcode  
03h: Logical error  
04h: Invalid qualifiers  
09h: Unable to perform retransmission request

---
NVMe Advantage#3: Error Detection & Recovery (SLER)

FC-NVMe-2 (Sequence Level Error Recovery)

**RED: Responder Error Detected**

The RED command may be transmitted by an Exchange Responder to indicate to the Exchange Originator that a Sequence error was detected on an open Exchange.

*1 After detecting a Sequence error the target transmits a RED BLS.

*2 After receiving the RED BLS, the initiator NVMe_Port transmits an NVMe_SR IU specifying the NVMe_XFER_RDY be resent.

*3 The target NVMe_Port retransmits the NVMe_XFER_RDY with Relative Offset set to zero.

*4 The initiator NVMe_Port retransmits the data.
NVMe Advantage#4: Performance Advantage for KV Databases (KV SSD)

Today all storage protocols (Block, NFS or Object) use LBA block addressing scheme.

KV protocol maps an address (Key, 32 bytes max.) to a physical location where (Value, 4GB max) is storage. No LBA, hence no translation in FTL.

Key Value API (SNIA)
- Open/Retrieve Device
- Create/Delete Key Space
- Store, Retrieve, Delete,
- List, Delete Group

NVMe KV I/O Commands
(Store, Retrieve, List, Exist, Delete)
NVMe Advantage#5: Lower CPU utilization (CPU Offload with CMB)

CPU Offload with NVMe Controller Memory Buffer (CMB)
NVMe Advantage#6: Computational Storage (Future)

1. NVMe Read (NS) is issued to CNTL
2. CNTL moves the (NS) data to CM
3. Execute PGM-0 on Compute E.-2
4. Read CM Output Data back to Host

TP4091  Computational Programs
        I/O Command Set
        -Execute Program
        -Load Program
        -Activate Program

TP4131  Controller Local Memory
Adoption and Use Cases

Nishant Lodha
Director, Emerging Technologies
Marvell Semiconductor
FC-NVMe Adoption - on the-rise

Low-latency business-critical applications that transact with all-flash arrays are driving the adoption of FC-NVMe

Industry estimates that ~10% of Fibre Channel deployments leverage FC-NVMe

Preserve and leverage existing investments with full backward compatibility with FCP-SCSI
FC-NVMe Hardware Ecosystem

HBAs
- Broadcom Emulex
- Marvell QLogic

Switches
- Cisco MDS
- Broadcom Brocade

Storage Array
- DELL EMC, NetApp, HPE, Pure Storage, IBM, others
# FC-NVMe: Operating System Support

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Support Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux</strong></td>
<td>Fully supported</td>
</tr>
<tr>
<td></td>
<td>RHEL 7.8 / 8.0 onwards</td>
</tr>
<tr>
<td></td>
<td>SLES 12 SP4 / SLES 15 SP2 onwards</td>
</tr>
<tr>
<td></td>
<td>Multipathing and Boot</td>
</tr>
<tr>
<td><strong>VMware ESXi</strong></td>
<td>Fully supported</td>
</tr>
<tr>
<td></td>
<td>VMware ESXi 7.0 onwards</td>
</tr>
<tr>
<td></td>
<td>VMware ESXi 8.0 adds vVols support</td>
</tr>
<tr>
<td></td>
<td>Multipathing and Boot</td>
</tr>
<tr>
<td><strong>Microsoft Windows</strong></td>
<td>Emulation Mode Support</td>
</tr>
<tr>
<td></td>
<td>No native support for FC-NVMe</td>
</tr>
<tr>
<td></td>
<td>HBA Vendors provide emulated drivers</td>
</tr>
<tr>
<td></td>
<td>Does not impact Windows VMs on VMware ESXi or KVM</td>
</tr>
</tbody>
</table>

*FC-NVMe adoption is growing, join in!*
NVMe-FC Performance Example
Improves Performance of Containerized AI/ML Models

• From Feb 2023 FCIA Webcast
https://www.brighttalk.com/webcast/14967/57217

<table>
<thead>
<tr>
<th>Dataset size</th>
<th>887MB</th>
<th>1.8GB</th>
<th>2.6GB</th>
<th>27GB</th>
<th>35GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container- NVMe over Fibre Channel</td>
<td>86</td>
<td>192</td>
<td>409</td>
<td>6920</td>
<td>10618</td>
</tr>
<tr>
<td>Container- FCP</td>
<td>112</td>
<td>296</td>
<td>563</td>
<td>8121</td>
<td>12394</td>
</tr>
<tr>
<td>VM- NVMe over Fibre Channel</td>
<td>124</td>
<td>337</td>
<td>616</td>
<td>9356</td>
<td>15882</td>
</tr>
<tr>
<td>VM- FCP</td>
<td>184</td>
<td>404</td>
<td>795</td>
<td>11232</td>
<td>17,964</td>
</tr>
</tbody>
</table>

Performance Improvement with container on NVMe over Fibre Channel %

30%      54%      37%      37%      17%
NVMe/FC vs FCP - Database OLTP Performance

64GFC/NVMe-FC Database Server
All Flash Storage Array with 4x 32GFC NVMe/FC

- **OLTP TPM**
  - 53%
  - Faster Transactions than FCP/SCSI Protocol
  - HammerDB TPROC-C OLTP TPM

- **Stored Procedure LATENCY**
  - 58%
  - Lower Latency than FCP/SCSI Protocol
  - HammerDB TPROC-C Time Profile 95th percentile Stored Procedure latency

- **CPU Efficiency**
  - 106%
  - Better Server CPU Efficiency than FCP/SCSI Protocol
  - Greater HammerDB TPROC-C TPM per Server CPU Utilization
After this Webcast

• Please rate this event – we value your feedback
• We will post a Q&A blog at http://fibrechannel.org/ with answers to today’s questions
• Our library of FCIA on-demand webcasts http://fibrechannel.org/webcasts/ includes:
  • Fibre Channel Fundamentals
  • FC-NVMe
  • FC-NVMe for ML
  • Long Distance Fibre Channel
  • Fibre Channel Speedmap
  • FCIP (Extension): Data Protection and Business Continuity
  • Fibre Channel Performance
  • FICON
  • Fibre Channel Cabling
  • 64GFC and 128GFC
  • FC Zoning Basics
• Follow us @FCIAnews for updates on future FCIA webcasts
Thank You