
CISCO Live!
ALL IN

The bridge to possible

Dos and Don’ts of Deploying NVMe Over Fabrics

Kamal Bakshi
Director Technical Marketing
BRKDCN -3812

cisco Live!

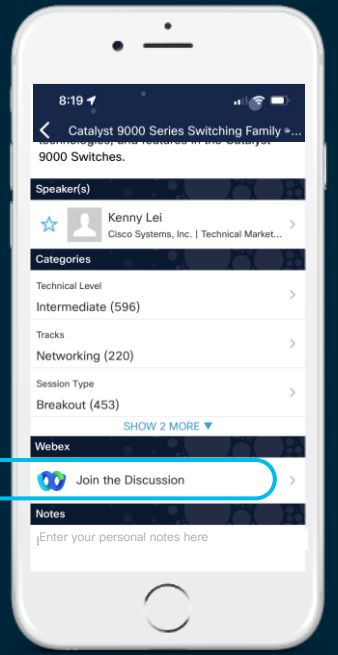
Cisco Webex App

Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space



https://ciscolive.ciscoevents.com/ciscolivebot/#BRKXXX-xxxx

Webex spaces will be moderated by the speaker until June 17, 2022.

NVMe Adoption

- Today (2022) total NVMe market size is over $80 Billion
- By 2030 NVMe market will exceed $175 Billion (CAGR 28%)
- Nearly ALL servers shipping today support NVMe drives
- All enterprise networking adapters sold today are NVMe-oF
- Over 80% of the All Flash Storage Arrays are based on NVM
- By 2026 SSD/flash will be cheaper than enterprise HDD/disk

Sources: G2M Research, Wikibon, & others

Future-Proof your IT Infrastructure by upgrading to NVMe today

What is NVMe/oF?

What problem are we trying to solve?

Why should I care?

What is the value proposition & advantages of this technology?

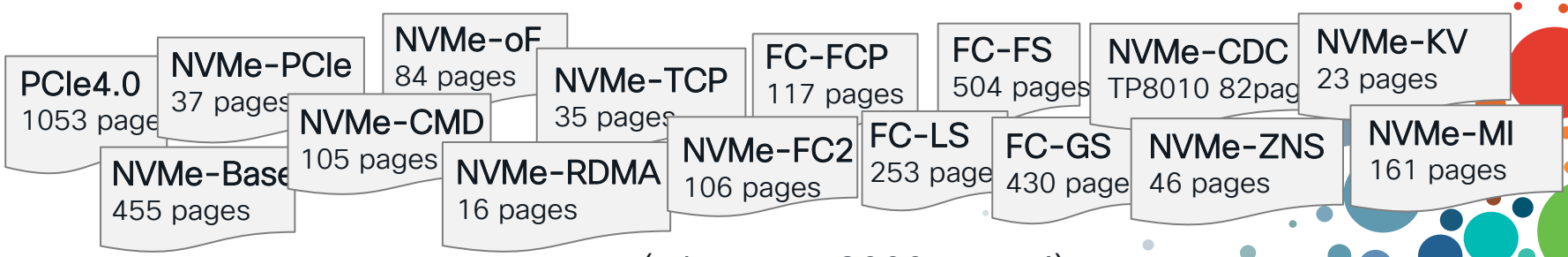
What to watch out for?

What are the Do's & Don'ts for best experience?

Reap Benefits!

Better performance,
Easy to maintain,
High ROI

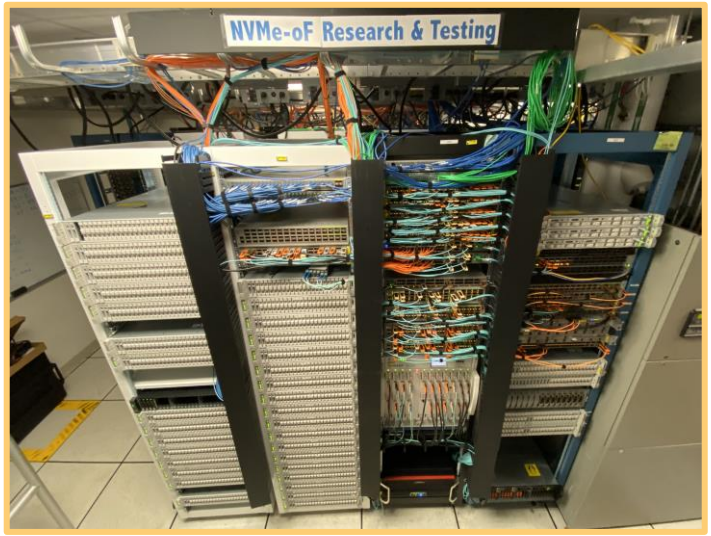
KNOWLEDGE IS THE KEY TO SUCCESS

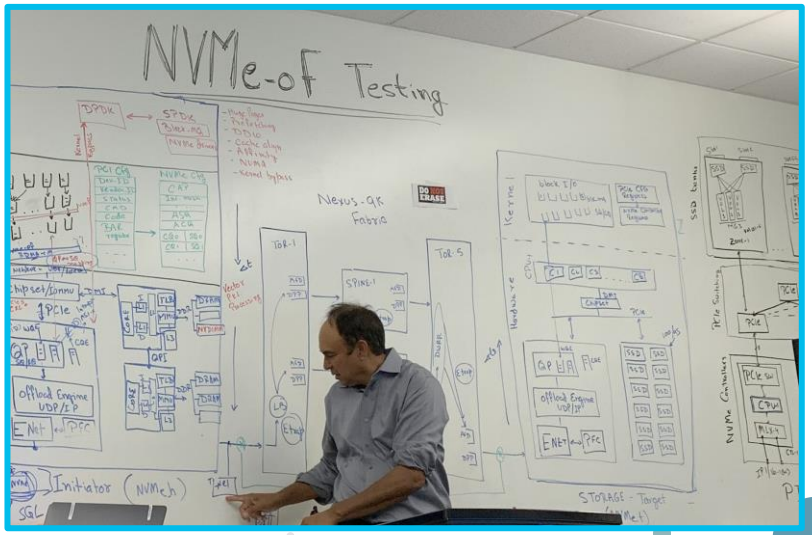


(...but over 3000 pages!)

Background....

For the past couple of years we have been extensively testing NVMe transports related technologies at Cisco DC POC lab. Information presented here is based on those experiences.





Cisco NVMe-oF Research Lab: Kamal Bakshi, Dhanaseker Kandhasamy, Frank Wang, Rithesh Iyer, Nemanja Kamenica, Paresh Gupta

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

3-NVMe Transport Options (FC, TCP, RoCEv2)

4-NVMe Datacenter Design

5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

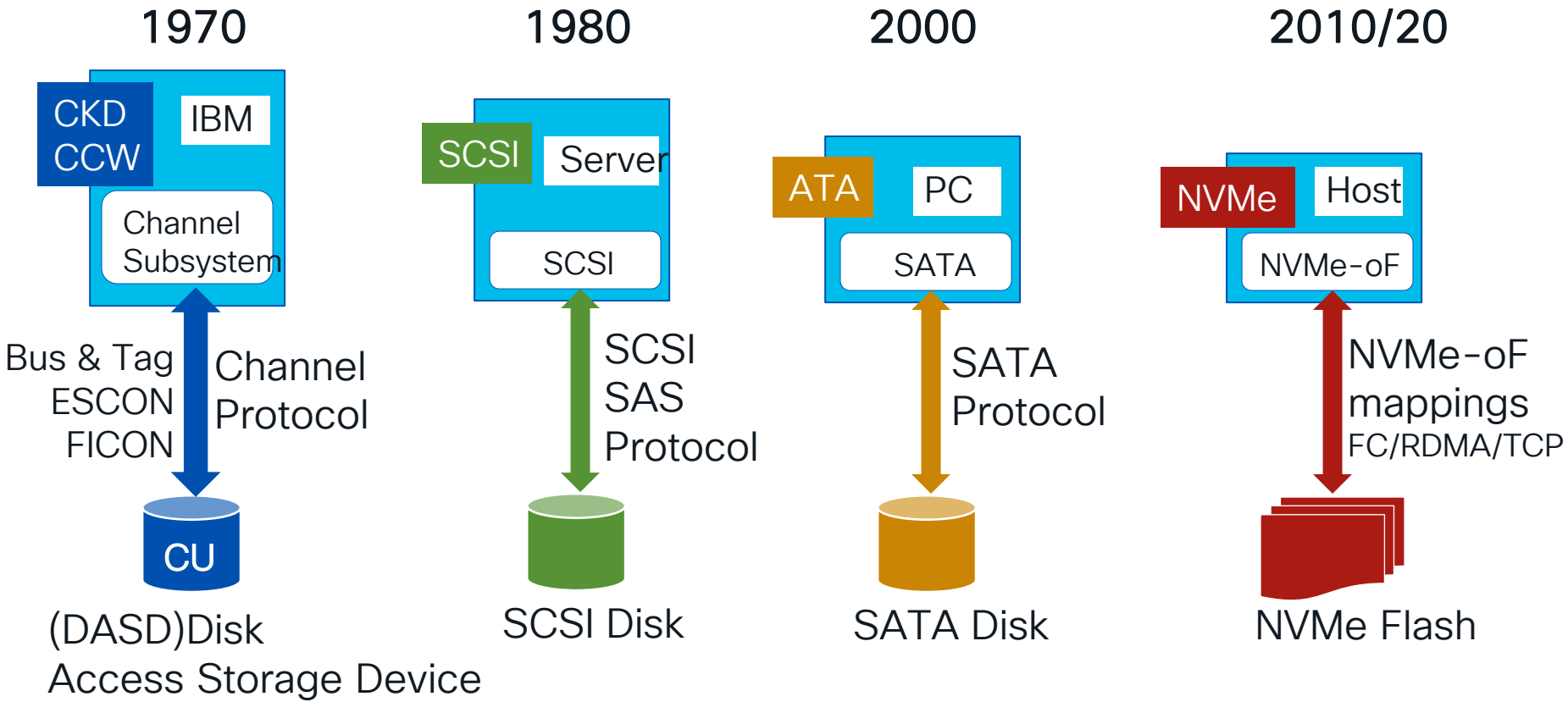
3-NVMe Transport Options (FC, TCP, RoCEv2)

4-NVMe Datacenter Design

5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

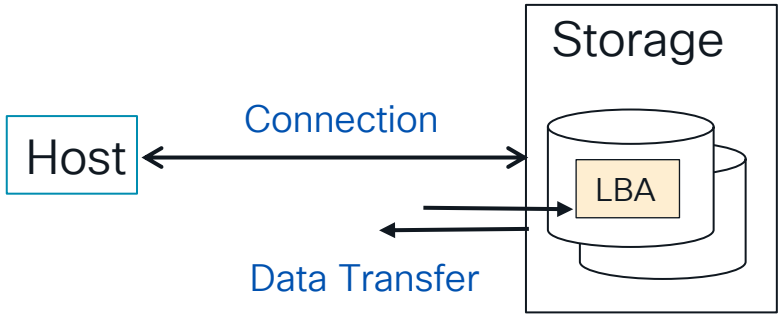
50,000 feet view of NVMe



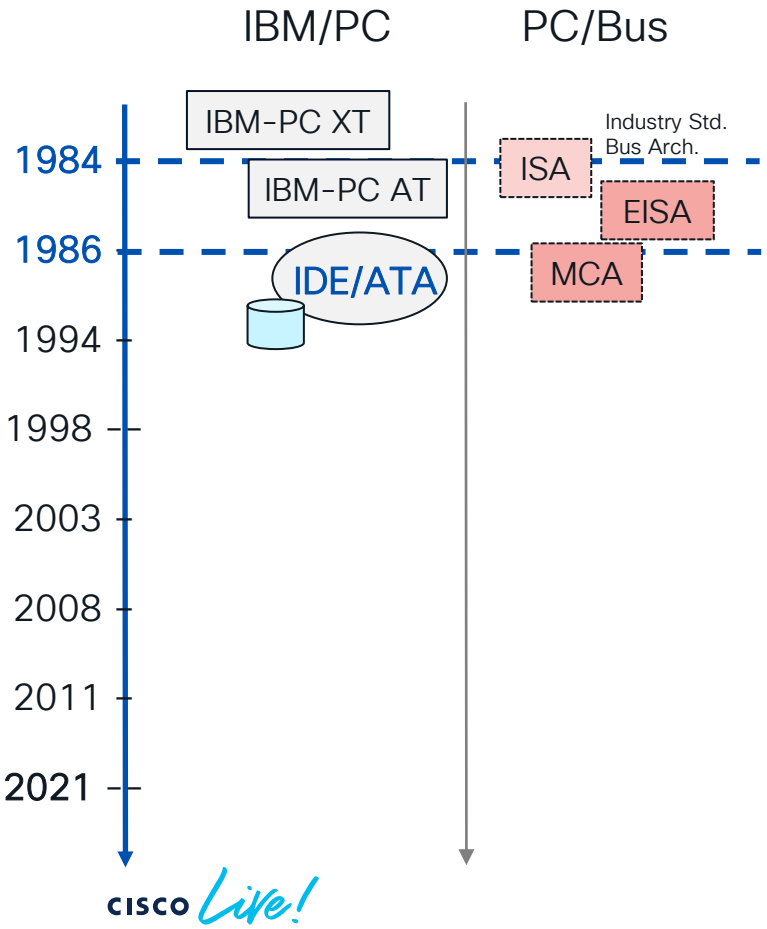
Why NVMe ?

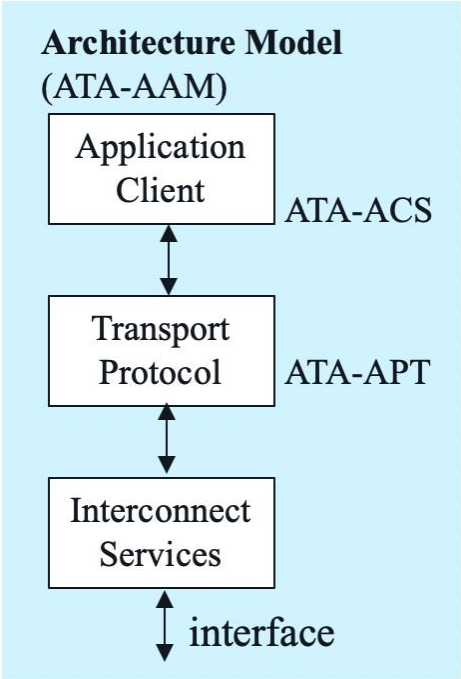
Problem Statement:

How to “connect”, Host to the Storage, and do “Data Transfer” ?



ATA (Advance Technology Attachment)

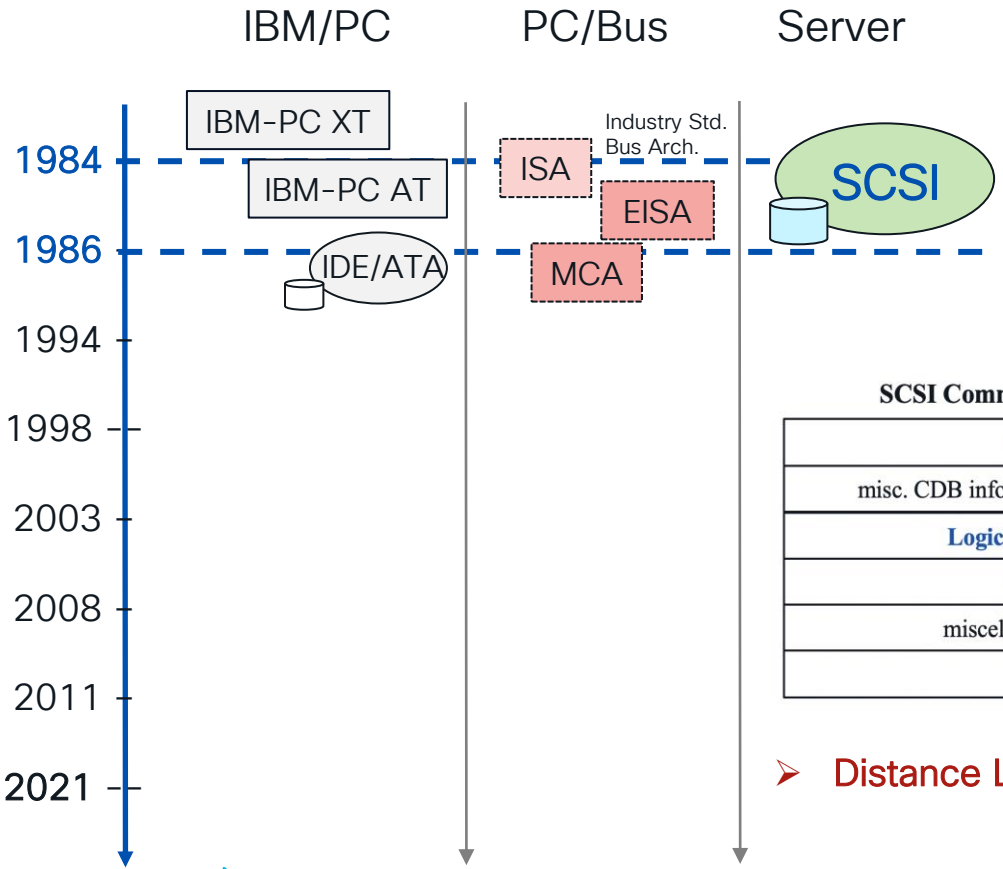




Write Sector Command Input

Field	Description
Feature	N/A
Count	# of Logical Sectors
LBA	Logical Block Address
Command	30h

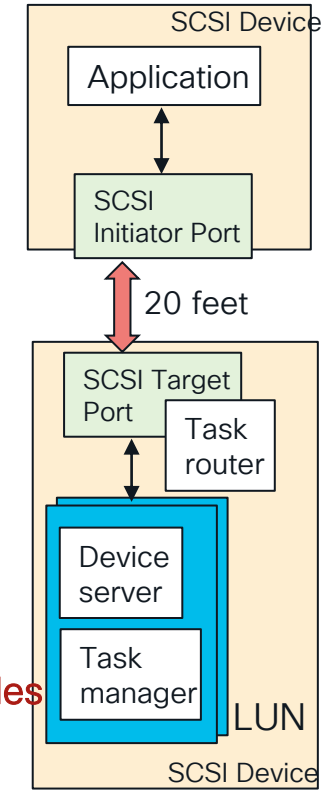
SCSI (Small Computer System Interface)



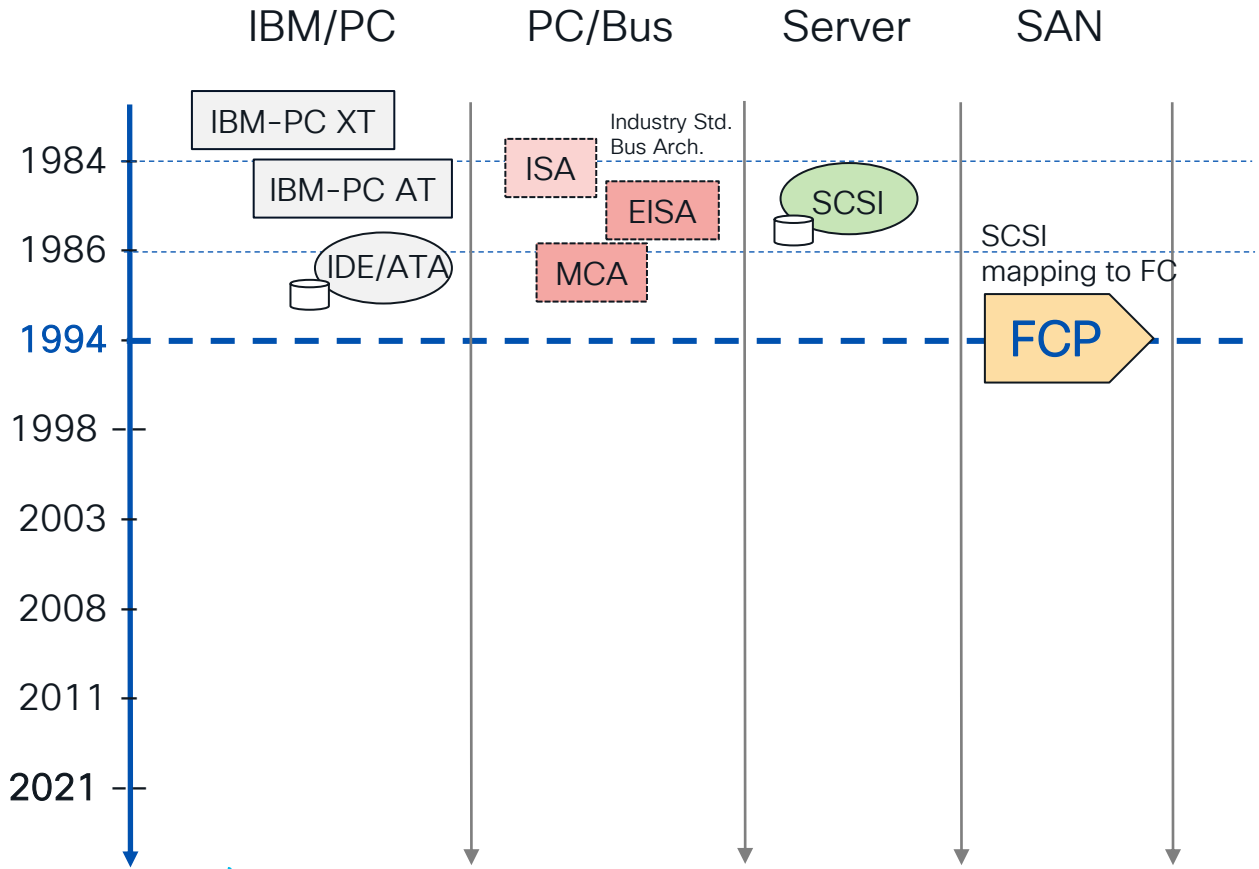
SCSI Command Descriptor Block -CDB

Operation Code	
misc. CDB information	Service Action (if any)
Logical Block Address (LBA)	
Transfer Length	
miscellaneous CDB information	
Control	

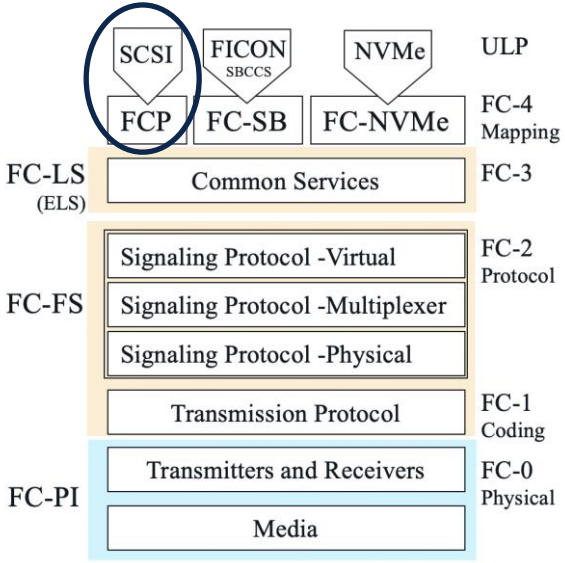
- Distance Limitation with SCSI Cables



FCP (Fibre Channel Protocol)

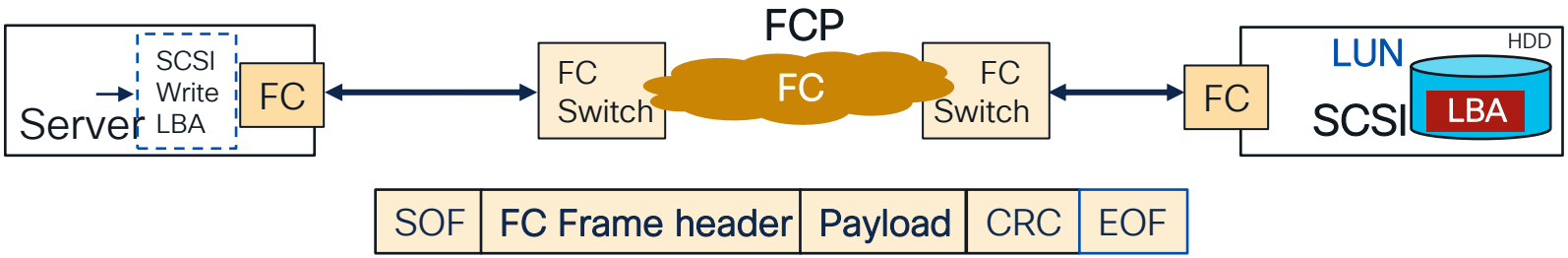


FCP -Fibre Channel Protocc



Fibre Channel Architecture

FCP (SCSI Protocol mapped into Fibre Channel)



SCSI WRITE (16) Command

Operation Code (8Ah)					
WRPROTECT	DPO	FUA	Rsvd	Obsolete	DLD2
Logical Block Address (LBA)					
Transfer Length					
DLD1	DLD0	Group Number			
Control					

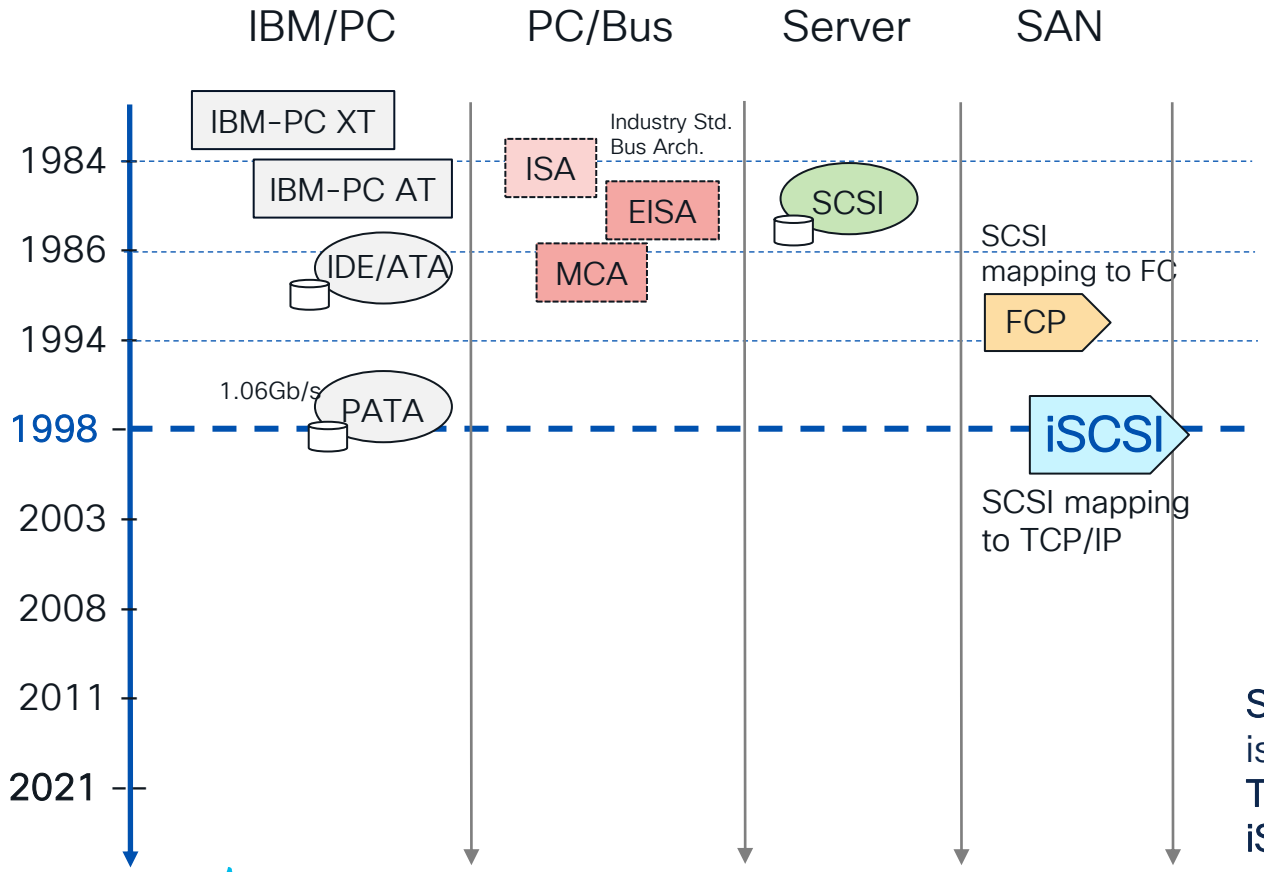
FCP Command IU Payload

FCP_LUN		
Command Reference Number		
Rsvd	Command Priority	Task Attribute
Task Management Flags		
Additional FCP_CDB Length	RDDATA	WRDATA
FCP_CDB		
Additional FCP_CDB (if any)		
FCP_DL		
FCP_Bidirectional_Read_DL (if any)		

FC Frame Header

R_CTL	D_ID	
CS_CTL	S_ID	
TYPE	F_CTL	
SEQ_ID	DF_CTL	SEQ_CNT
OX_ID		RX_ID
Parameter		
FCP Payload		

iSCSI (SCSI over TCP/IP)



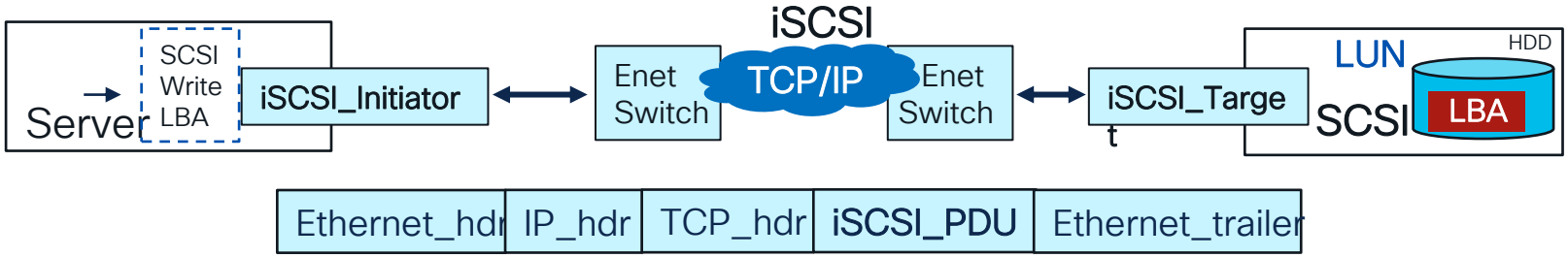
iSCSI Architecture

- iSCSI Initiator
- iSCSI Target
- "iqn" iSCSI Qualified Name
- Login/Logout
- Task Management
- iSNS Server (optional)
  - Name Service
  - Discovery Domain
  - State Change Notification
- Single_queue / Multi_queue(rece

Standard NIC- Performance issues....$
TOE NIC- TCP/IP Offload Engine.....$$
iSCSI HBA- iSCSI & TCP/IP offload..$$$

iSCSI (SCSI Protocol mapped into TCP/IP)

Issue: Limited Performance



Port# 860,3260

SCSI WRITE (16) Command

Operation Code (8Ah)					
WRPROTECT	DPO	FUA	Rsvd	Obsolete	DLD2
Logical Block Address (LBA)					
Transfer Length					
DLD1	DLD0	Group Number			
Control					

SCSI Command PDU

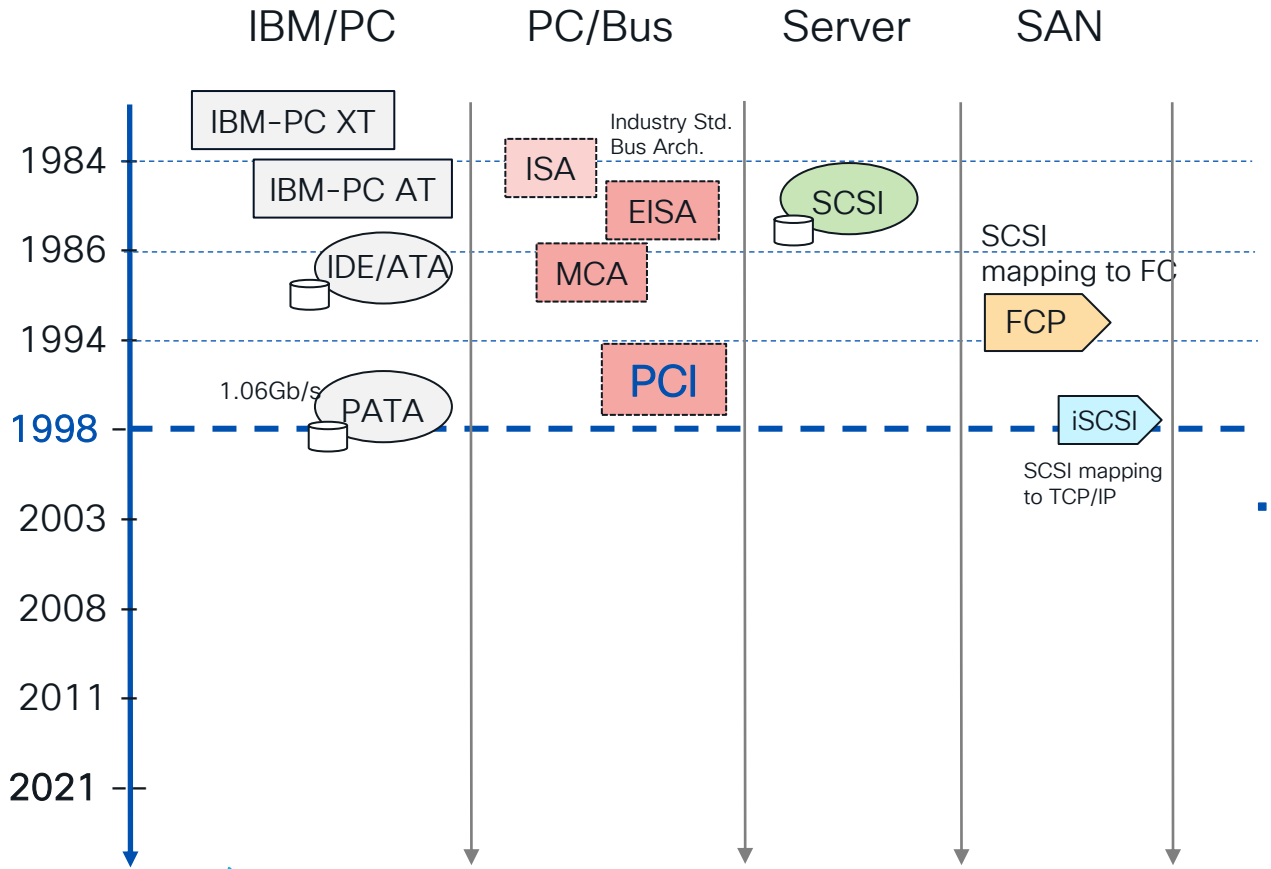
Opcode (0x01)	Opcode specific flags
Total AHS length	Data Segment length
Logical Unit Number (LUN)	
Initiator Task Tag	
Expected Data Transfer Length	
Command Sequence Number	
ExpStatSN	
SCSI Command Descriptor Block (CDB)	

iSCSI PDU

Basic Header Segment (BHS)
Additional header Segments (AHS)*
Header-Digest*
Data Segment*
Data-Digest*

** Optional*

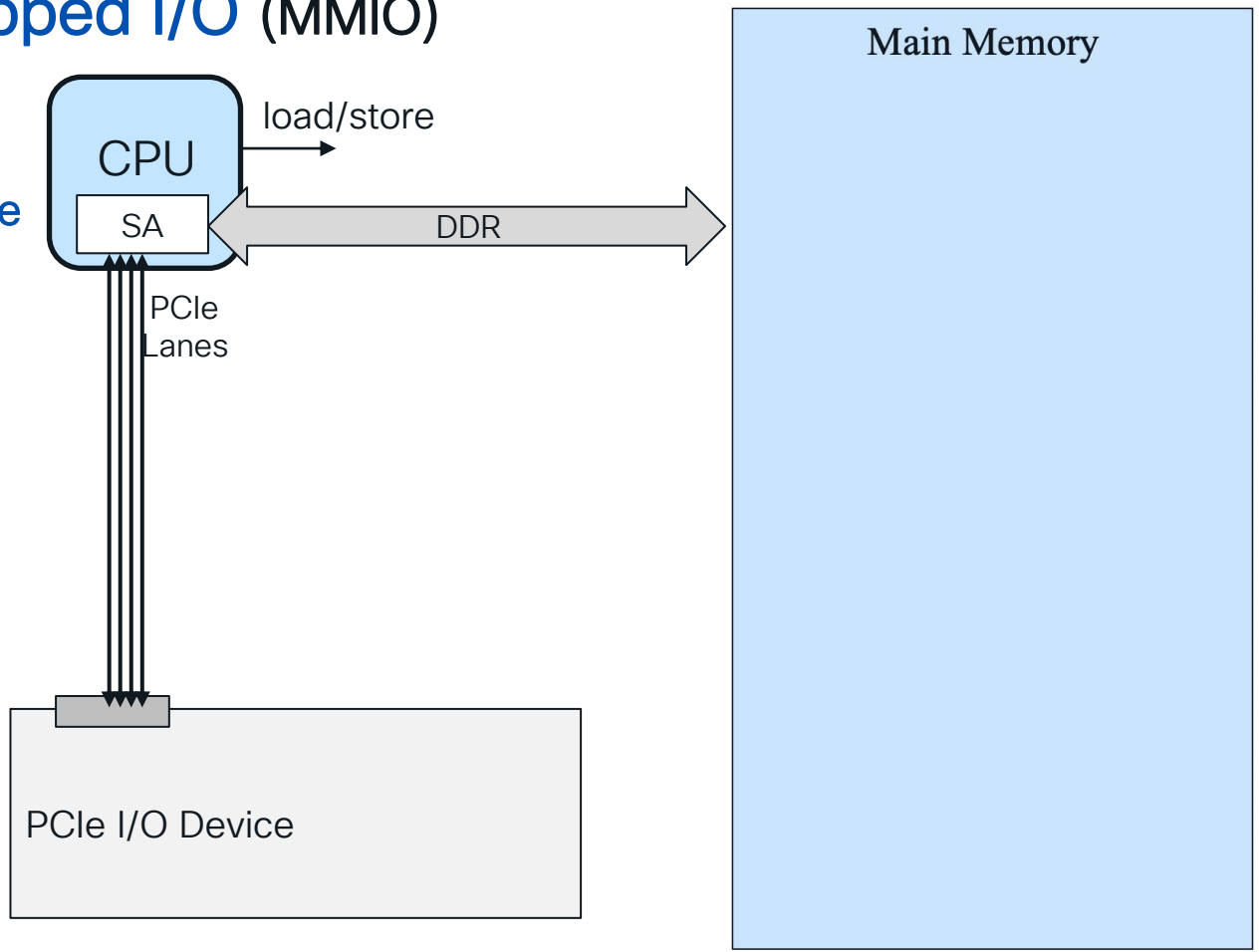
PCI (Peripheral Component Interconnect)



- PCI Architecture
  - Memory Mapped I/O
  - PCI Config. Registers
  - BAR space
  - Capability Registers
  - Message Signaled Interrupt

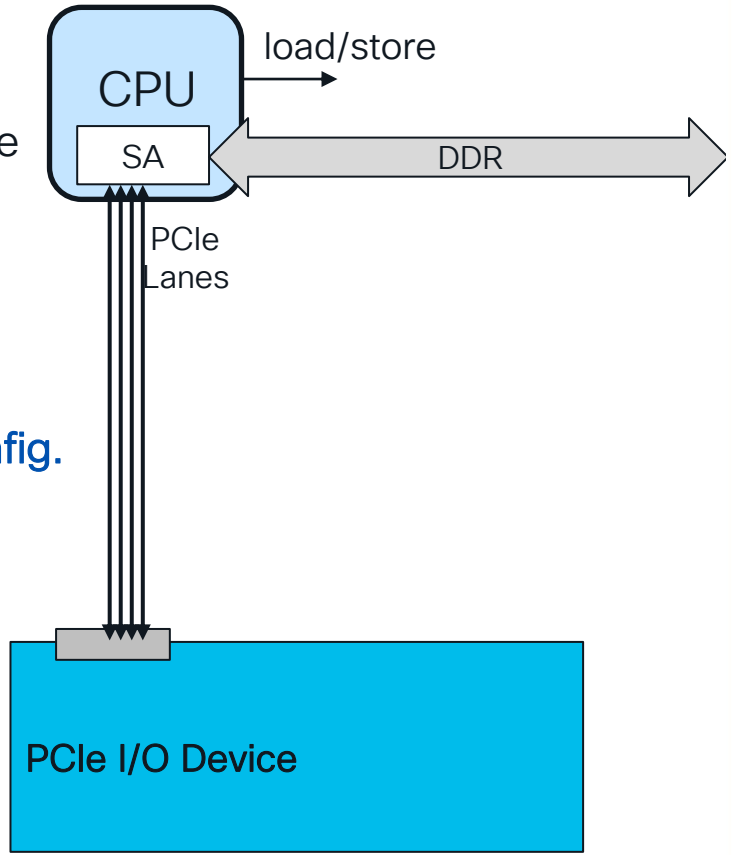
PCIe Memory Mapped I/O (MMIO)

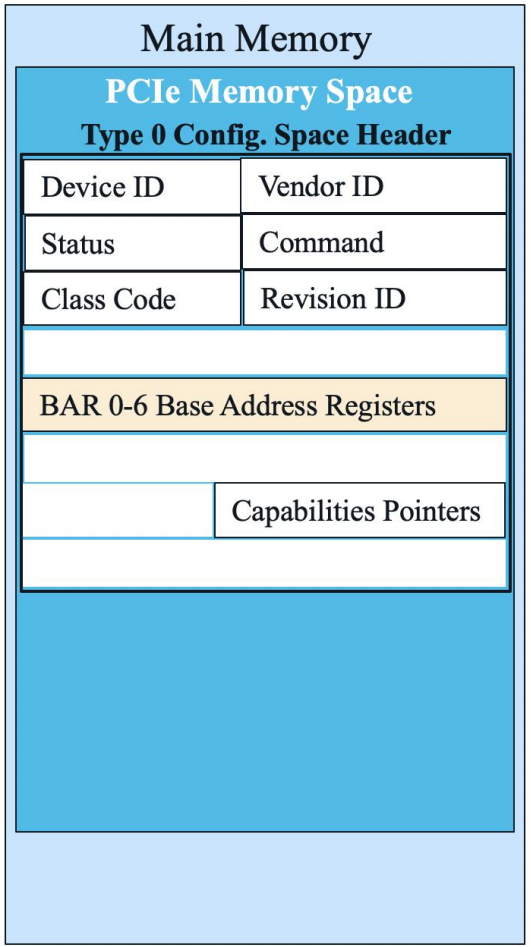
- With MMIO I/O devices are directly mapped into CPU main memory.
- No special set of special CPU instructions needed.



PCIe Memory Mapped I/O (MMIO)

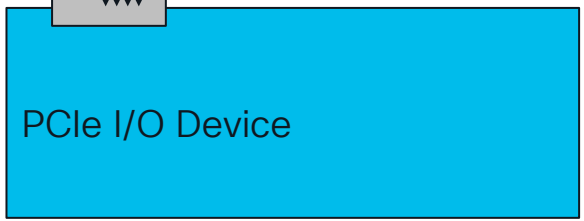
- With MMIO I/O devices are directly mapped into CPU main memory.
- No special set of special CPU instructions needed.
- Each PCIe device has config. space in main memory.

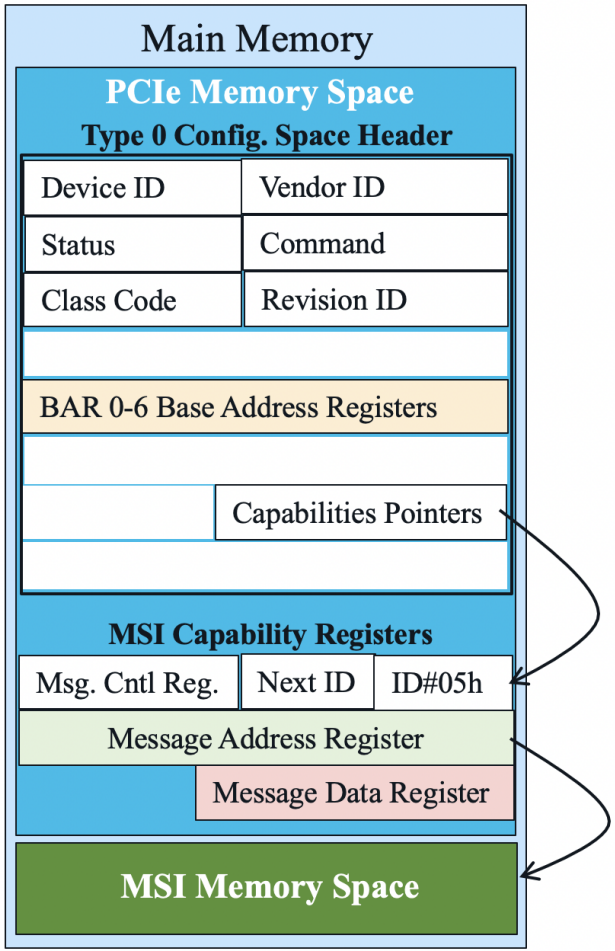




PCIe Memory Mapped I/O (MMIO)

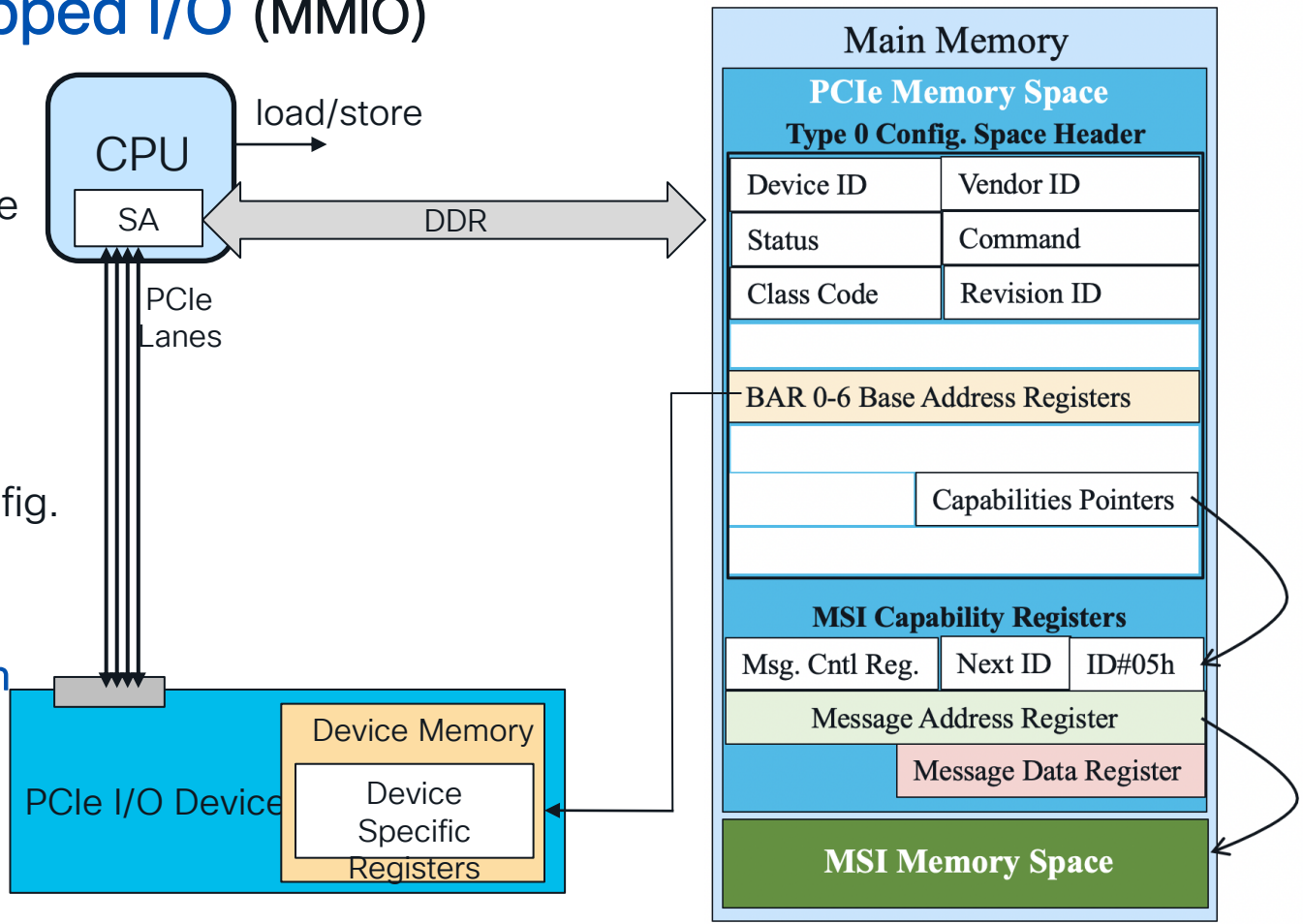
- With MMIO I/O devices are directly mapped into CPU main memory.
- No special set of special CPU instructions needed.
- Each PCIe device has config. space in main memory.





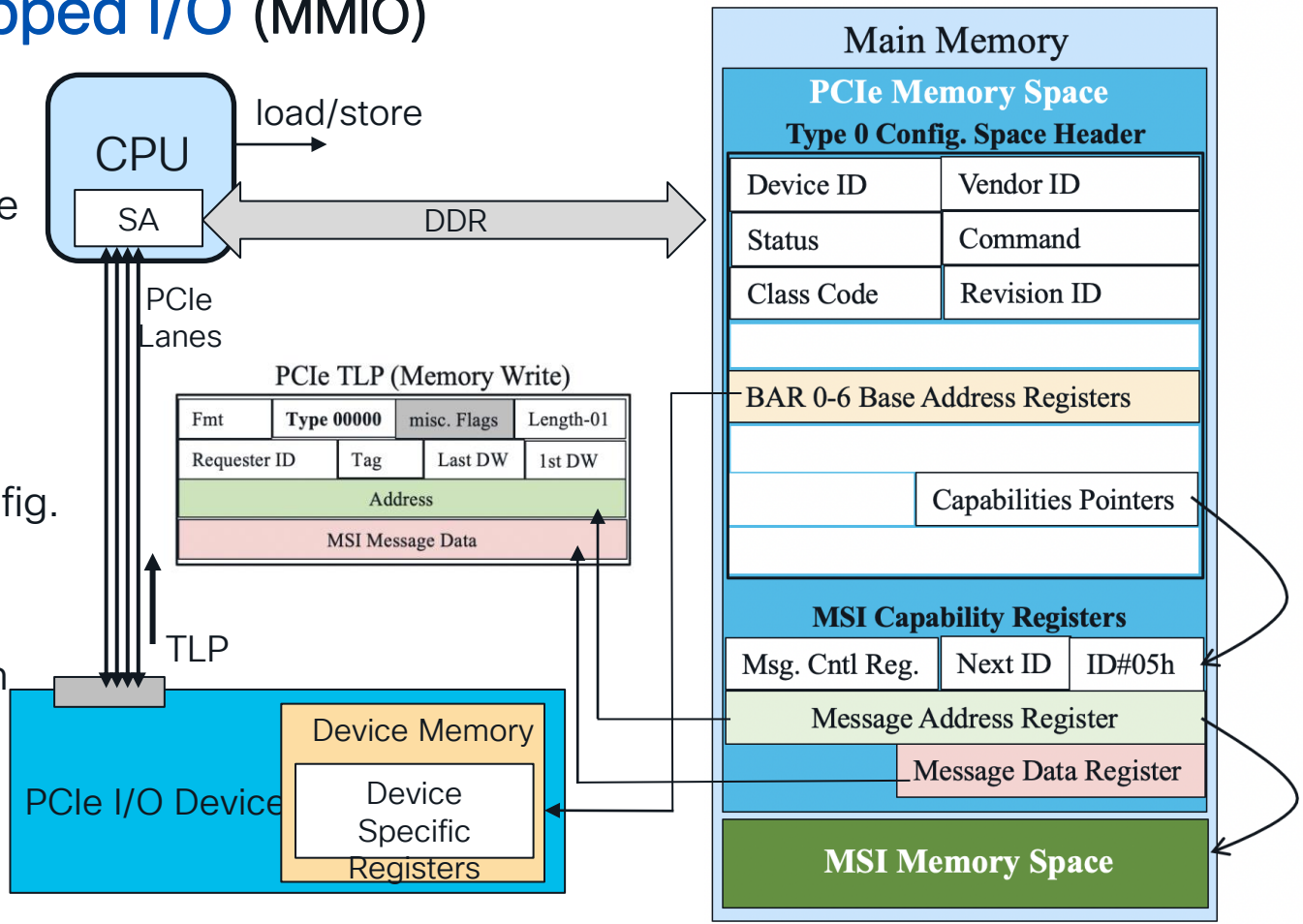
PCIe Memory Mapped I/O (MMIO)

- With MMIO I/O devices are directly mapped into CPU main memory.
- No special set of special CPU instructions needed.
- Each PCIe device has config. space in main memory.
- BAR registers map I/O device memory in the main memory

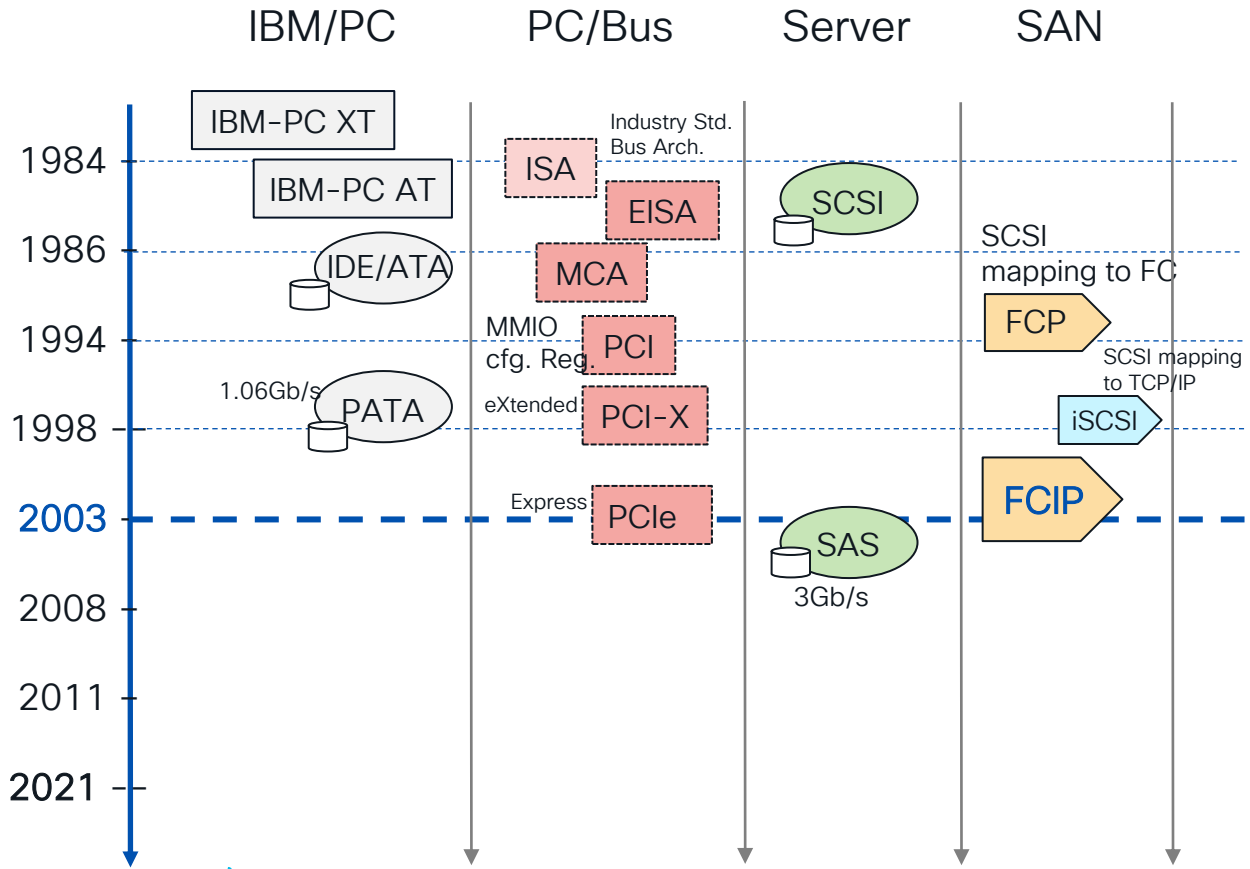


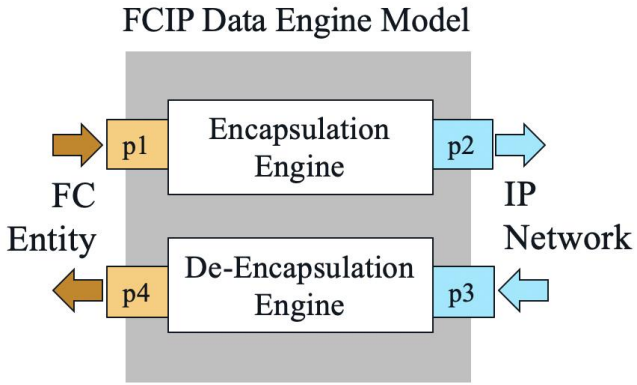
PCIe Memory Mapped I/O (MMIO)

- With MMIO I/O devices are directly mapped into CPU main memory.
- No special set of special CPU instructions needed.
- Each PCIe device has config. space in main memory.
- BAR registers map I/O device memory in the main memory
- MSI Message Signaled Interrupt



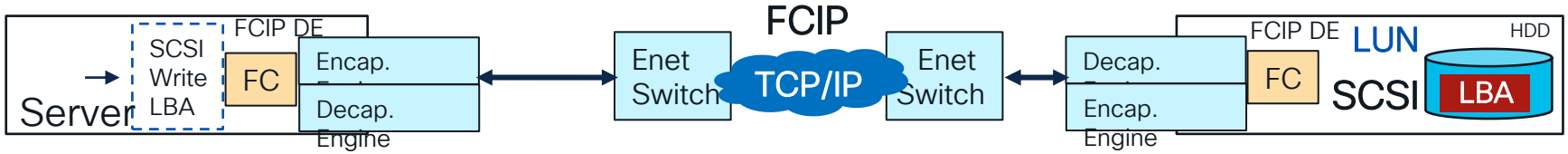
FCIP (Fibre Channel over IP)





FCIP Architecture

FCIP (FC Encapsulated inside TCP/IP Protocol)



Port# 3225

FCIP Header

DW0	Protocol #	Version	-Protocol #	-Version
DW1	Protocol #	Version	-Protocol #	-Version
DW2	pFlags	Reserved	-pFlags	-Reserved
DW3	Flags	Frame Length	- Flags	-Frame Length
DW4	Time Stamp (seconds)			
DW5	Time Stamp (second fraction)			
DW6	CRC			

FCP Header

R_CTL	D_ID	
CS_CTL	S_ID	
TYPE	F_CTL	
SEQ_ID	DF_CTL	SEQ_CNT
OX_ID		RX_ID
Parameter		

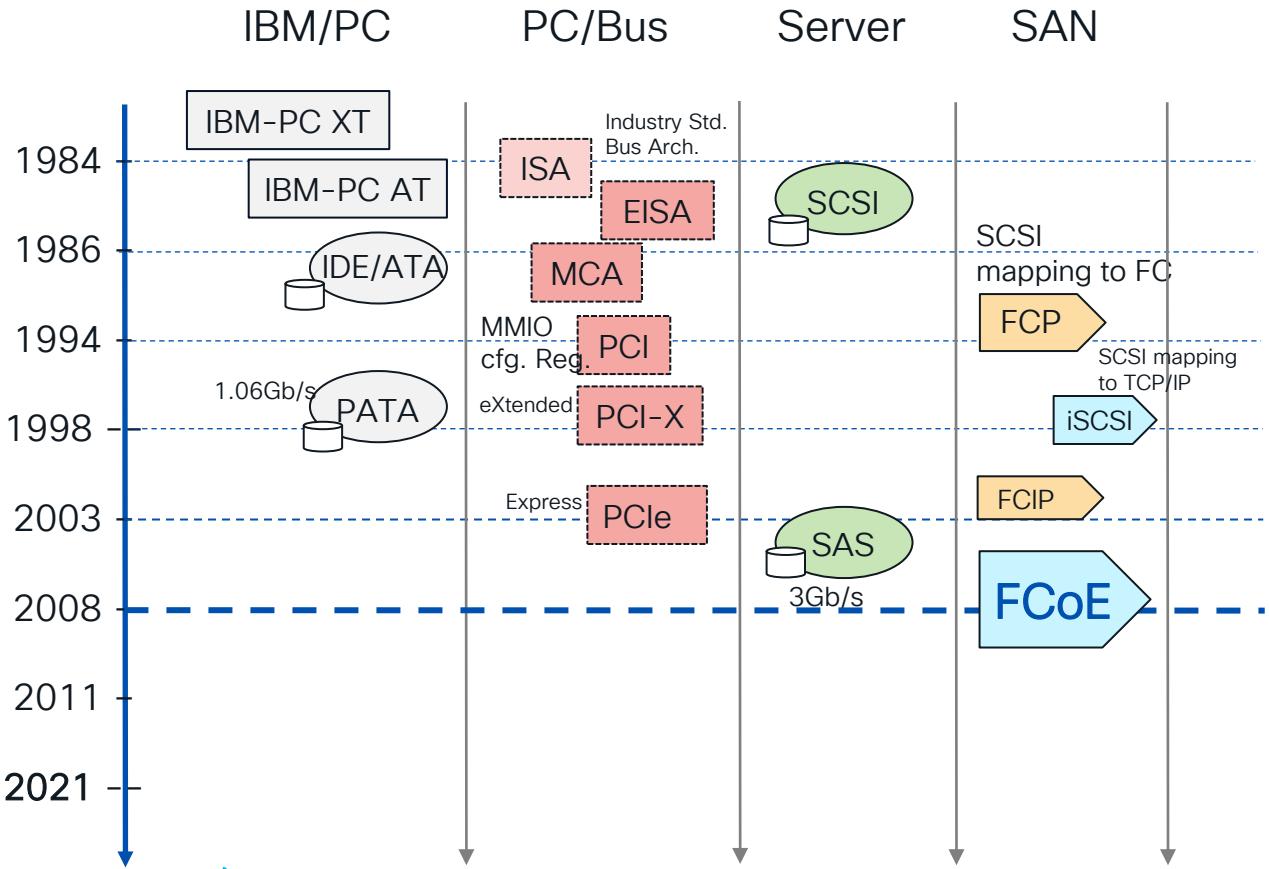
SCSI WRITE (16) Command

Operation Code (8Ah)					
WRPROTECT	DPO	FUA	Rsvd	Obsolete	DLD2
Logical Block Address (LBA)					
Transfer Length					
DLD1	DLD0	Group Number			
Control					

FCP Command IU Payload

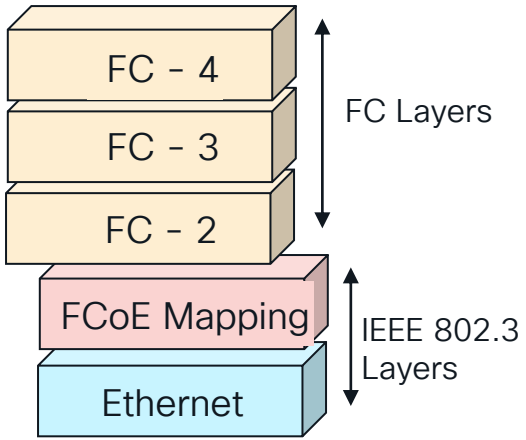
FCP_LUN		
Command Reference Number		
Rsvd	Command Priority	Task Attribute
Task Management Flags		
Additional FCP_CDB Length	RDDATA	WRDATA
FCP_CDB		
Additional FCP_CDB (if any)		
FCP_DL		
FCP_Bidirectional_Read_DL (if any)		

FCoE (Fibre Channel over Ethernet)

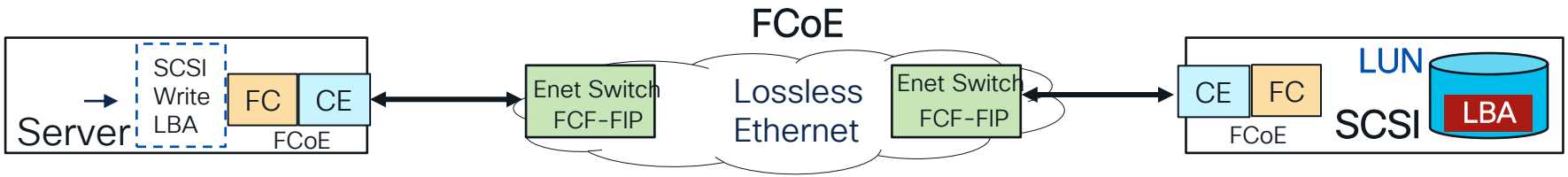


FC over Ethernet

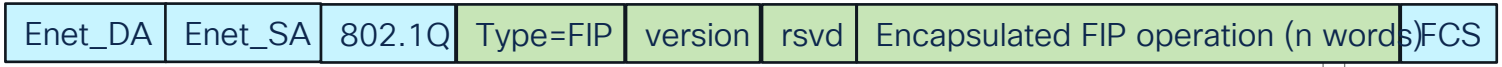
FCoE Protocol Stack

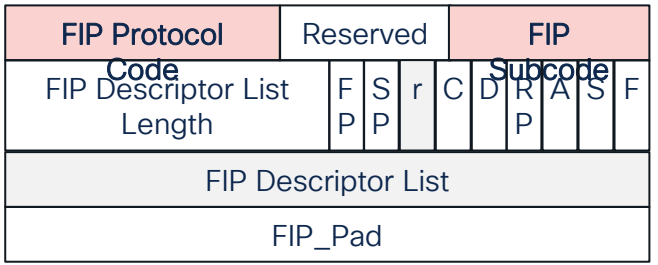


FCoE (Fibre Channel over Ethernet)



FIP (FCoE Initialization Protocol) Frame Format





FIP Operation (code/subcode)

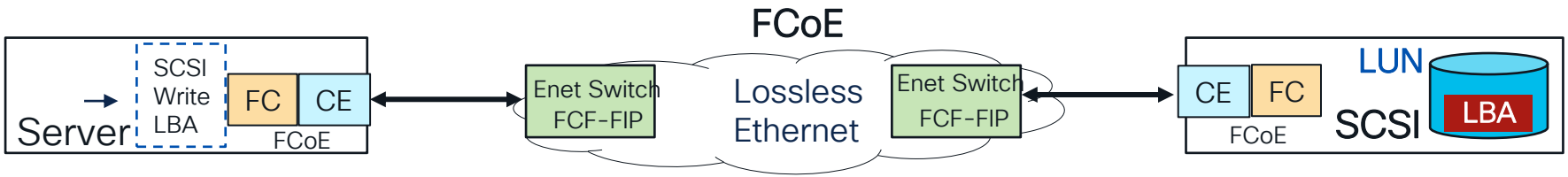
0001/01h Discovery Solicitation
0001/02h Discovery Advertisement
0002/01h Virtual Link Inst. Request
0002/02h Virtual Link Inst. Reply
0003/01h FIP Keep Alive
0003/02h FIP Clear Virtual Links
0004/01h FIP VLAN Request
0004/02h FIP VLAN Notification
0004/03h FIP VN2VN VLAN
0005/01h N_Port_ID Probe Request
0005/02h N_Port_ID Probe Reply
0005/03h N_Port_ID Claim Notification
0005/04h N_Port_ID Claim Response
0005/05h N_Port_ID Beacon
FFF8h - FFFEh Vendor Specific

FIP Descriptor Types

0-Reserved, 1-Priority, 2-MAC Address 3-FC_MAC, 4-Name_Identifier, 5-Fabric, 6-Max F
7-FLOGI, 8-NPIV FDISC, 9-LOGO, 10-ELP, 11-Vx_Port ID, 12-EKA_ADV_Period, 13-Vendo
14-VLAN, 15-VN2VN Attributes, 16-127 Reserved, 128-Clear Virtual Links Reason Code.

FCoE (Fibre Channel over Ethernet)

Issue: Scaling of FCoE protocol to multi-hop



FIP (FCoE Initialization Protocol) Frame Format

Enet_DA	Enet_SA	802.1Q	Type=FIP	version	rsvd	Encapsulated FIP operation (n words)	FCS
			0x8914				

FCoE Frame Format

Enet_DA	Enet_SA	802.1Q	Type=FCoE	version	rsvd	SOF	FC Frame header	Payload	CRC	EOF	rsvd	FCS
			0x8906									

SCSI WRITE (16) Command

Operation Code (8Ah)					
WRPROTECT	DPO	FUA	Rsvd	Obsolete	DLD2
Logical Block Address (LBA)					
Transfer Length					
DLD1	DLD0	Group Number			
Control					

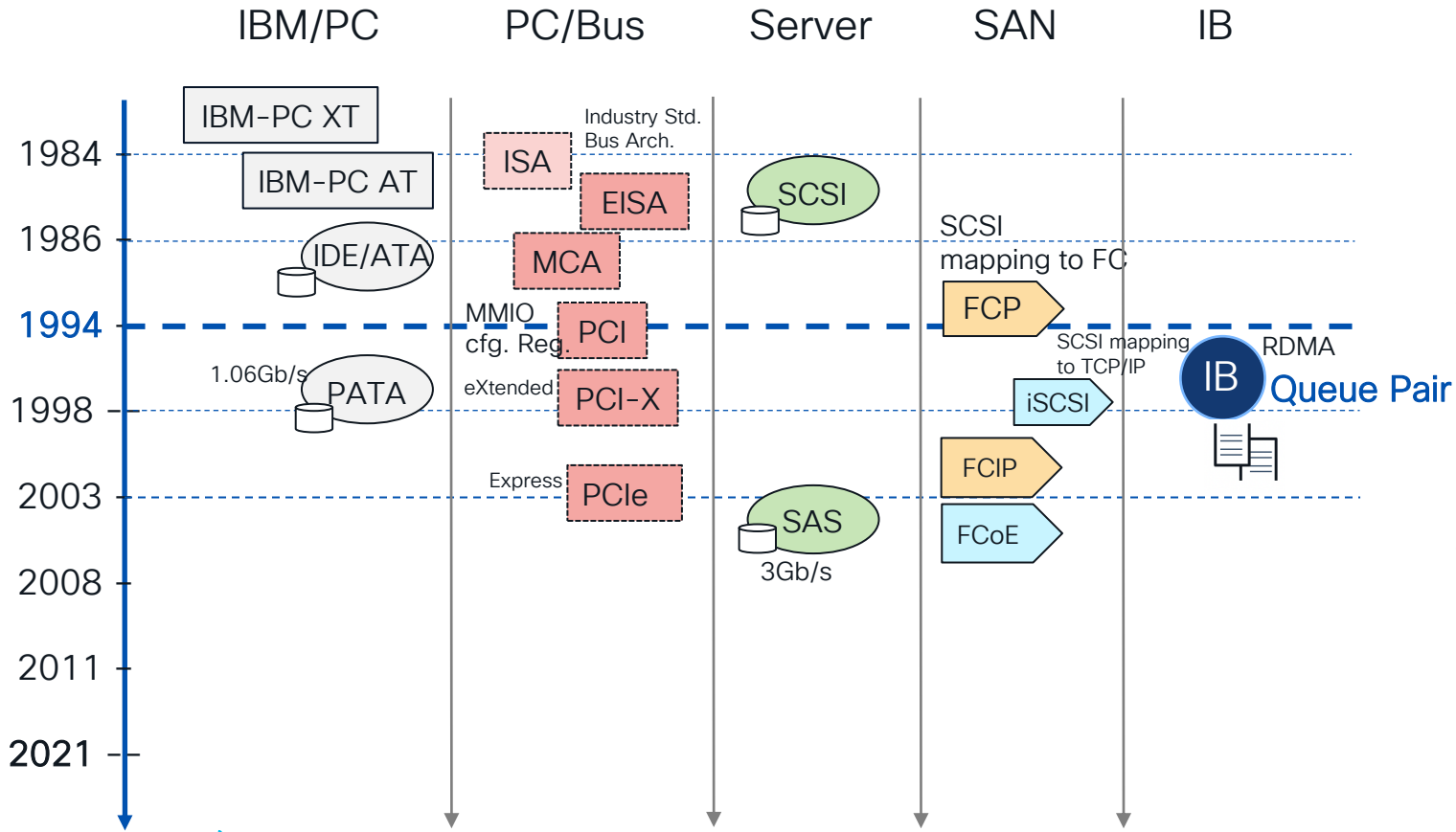
FCP Command IU Payload

FCP_LUN		
Command Reference Number		
Rsvd	Command Priority	Task Attribute
Task Management Flags		
Additional FCP_CDB Length	RDDATA	WRDATA
FCP_CDB		
Additional FCP_CDB (if any)		
FCP_DL		
FCP_Bidirectional_Read_DL (if any)		

FC Frame Header

R_CTL	D_ID	
CS_CTL	S_ID	
TYPE	F_CTL	
SEQ_ID	DF_CTL	SEQ_CNT
OX_ID		RX_ID
Parameter		
FCP Payload		

IB (InfiniBand)

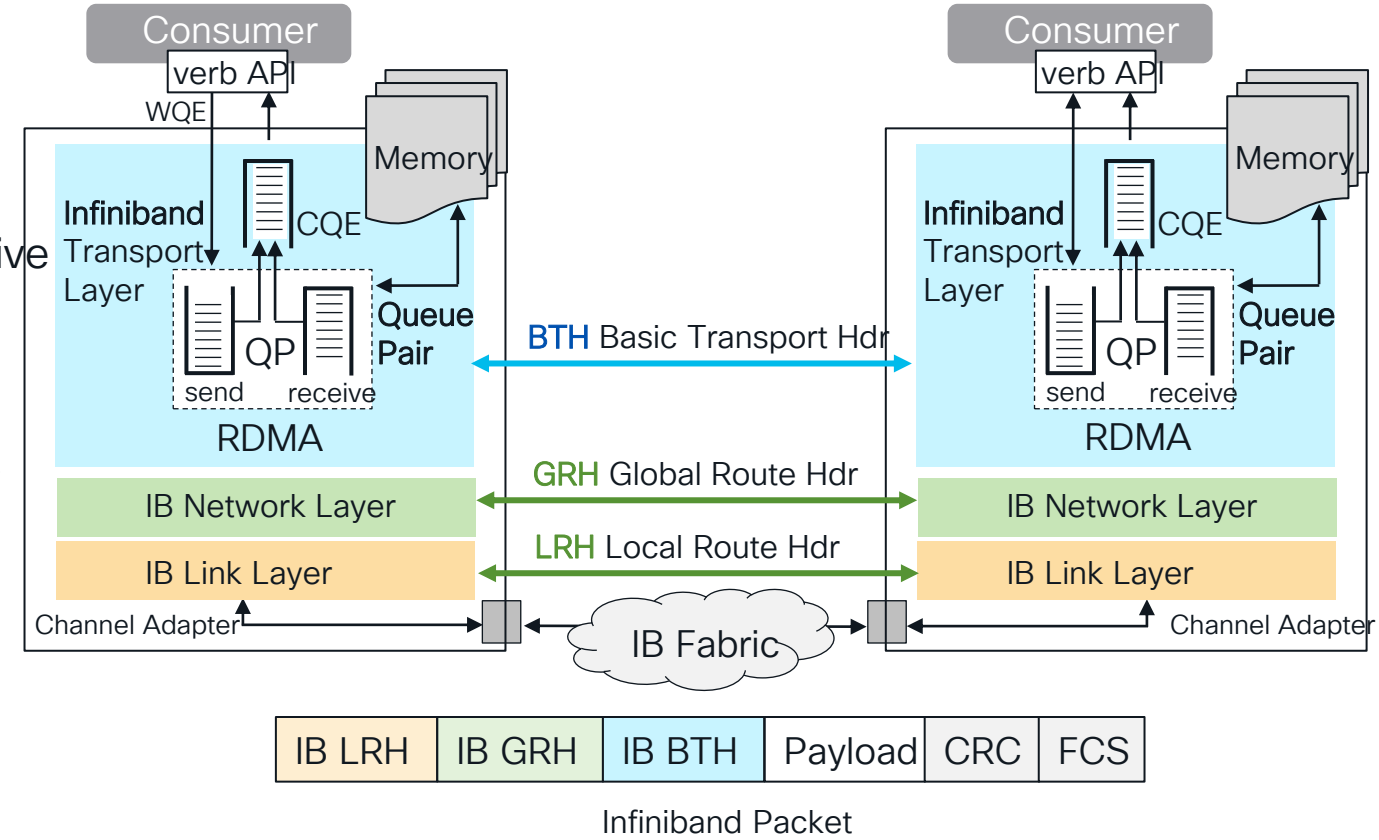


"Initially the IBTA vision for IB was simultaneously a replacement for PCI in I/O, Ethernet in the machine room, cluster interconnect and Fibre Channel." Wikipedia

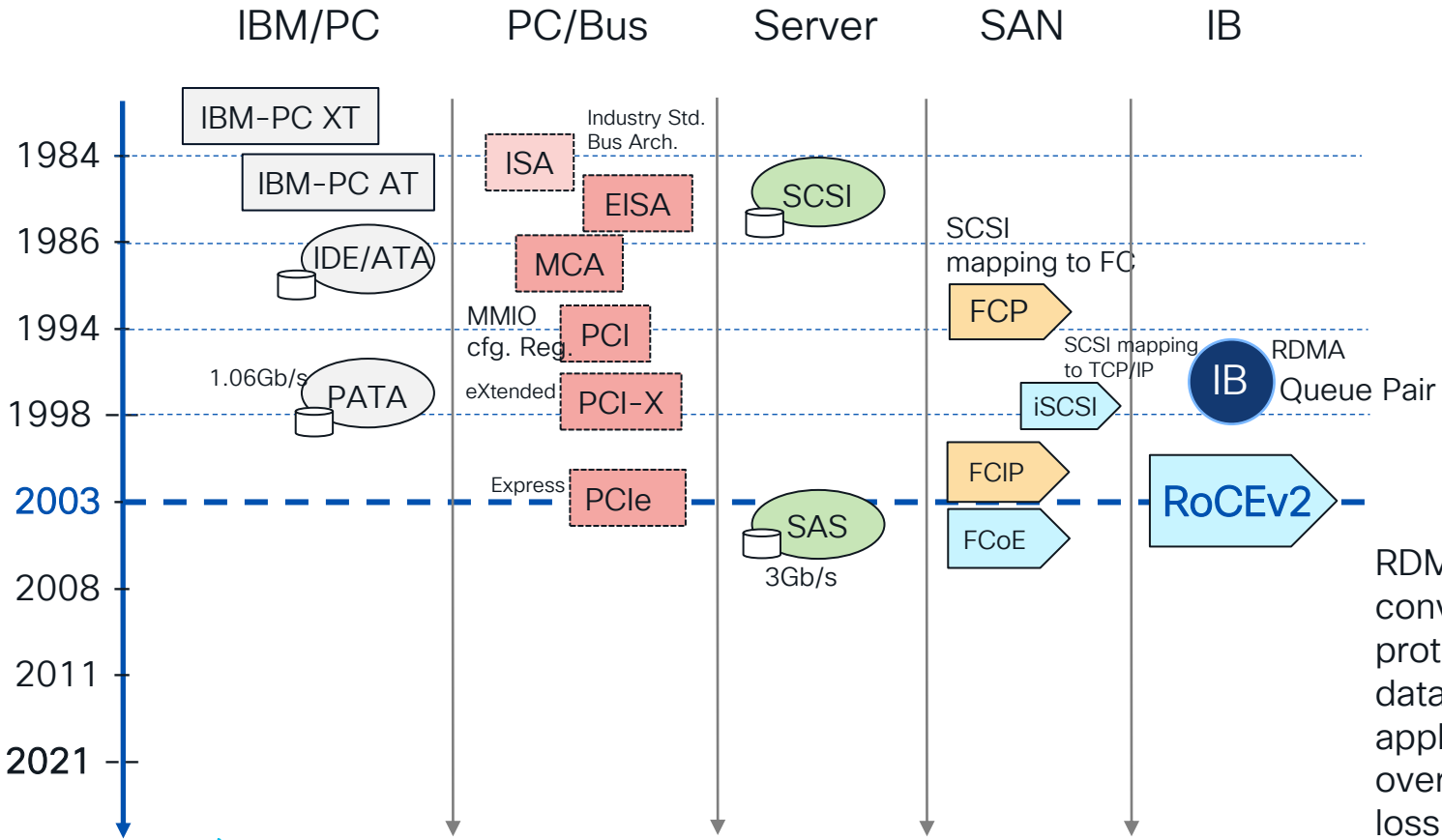
InfiniBand (Queue Pair based Remote Direct Memory Access)

InfiniBand (Queue Pair based Remote Direct Memory Access)

- Verb API
- RDMA Read/Write
- Message Send/Receive
- Kernel Bypass
- Queue Pair
- Completion Queue
- Work Queue Element



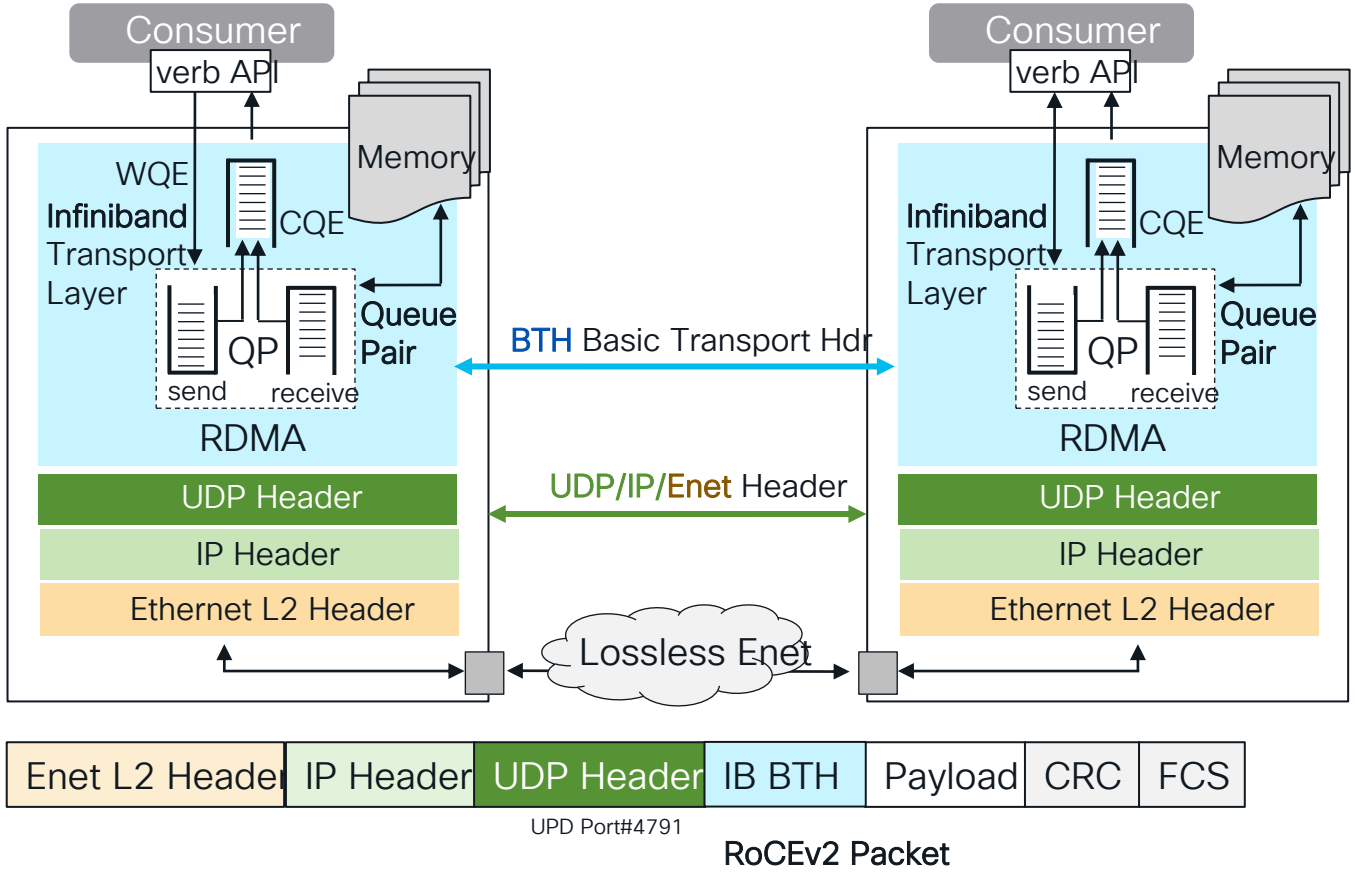
RoCEv2 (RDMA over Converged Ethernet)



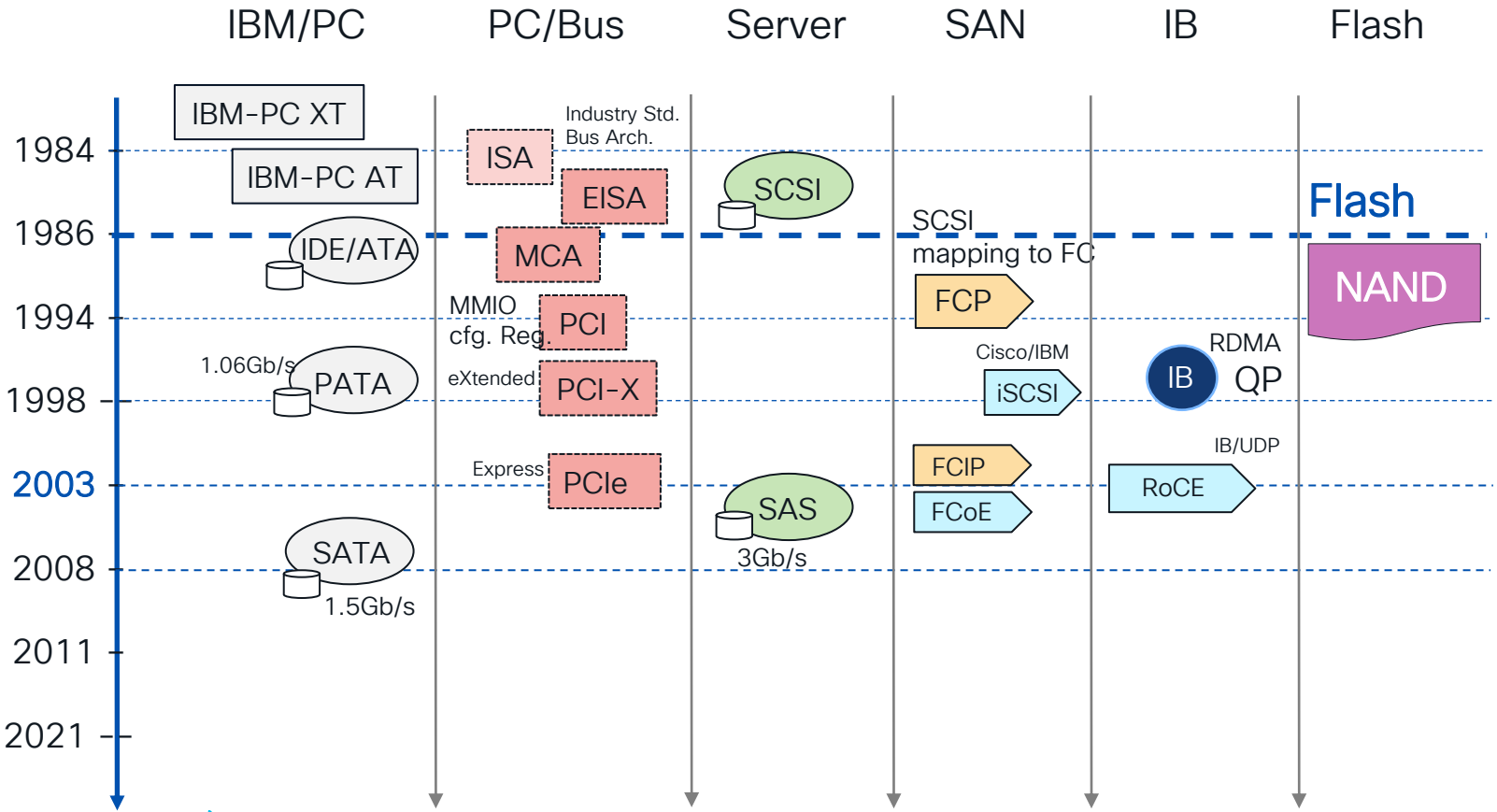
RDMA over converged Ethernet protocol allows data transfer between application memory over lossless Ethernet networks.

RoCEv2 (Architecture)

- Lossless Ethernet
- PFC
- ECN
- DCQCN
- CNP (IBTH)
- Resilient RoCEv2



Flash (Non Volatile Memory)



Flash (Non Volatile Memory)

"Flash memory is an electronic non-volatile computer memory storage medium that can be electrically erased and reprogrammed. The two main types of flash memory, NOR flash and NAND flash, are named for the NOR and NAND logic gates." …Wikipedia

NOR vs NAND:

NOR flash is faster to read but takes longer to write or erase and is mostly used in consumer devices like smartphones. **NAND has higher capacity and is cheaper as compared to NOR.**

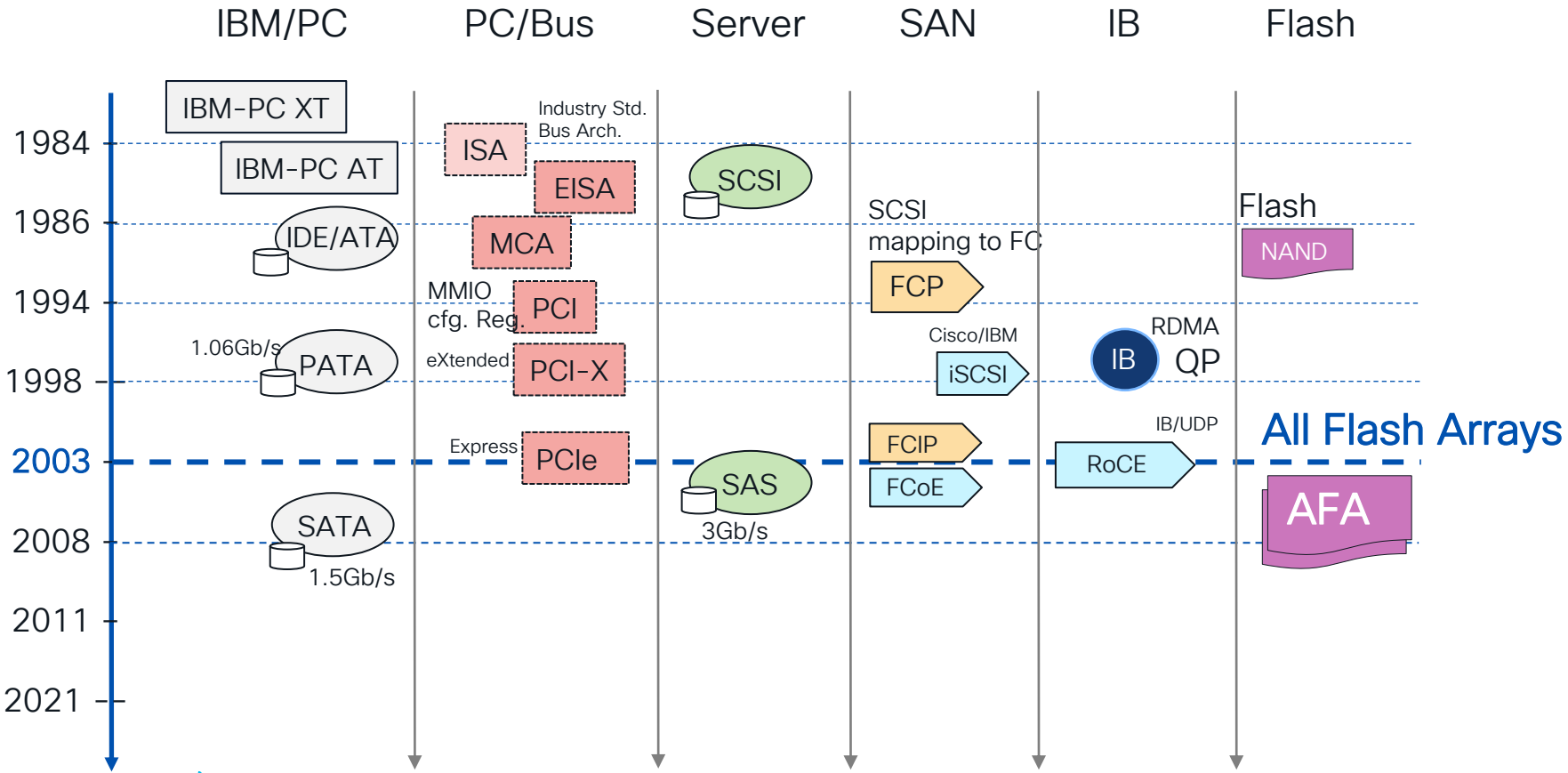
3D/V-NAND (Levels/Layers)

- SLC single level cell stores one bit per cell, MLC multi level cell stores two bits per cell, TLC triple level cell stores three bits per cell, QLC quad level cell stores 4 bits per cell.
- In 2D/planner NAND memory cells are connected in horizontal fashion but in 3D NAND they are stacked vertically in layers. (48, 64, 96, 128…**144-230**…256-layers…1000-layers!)

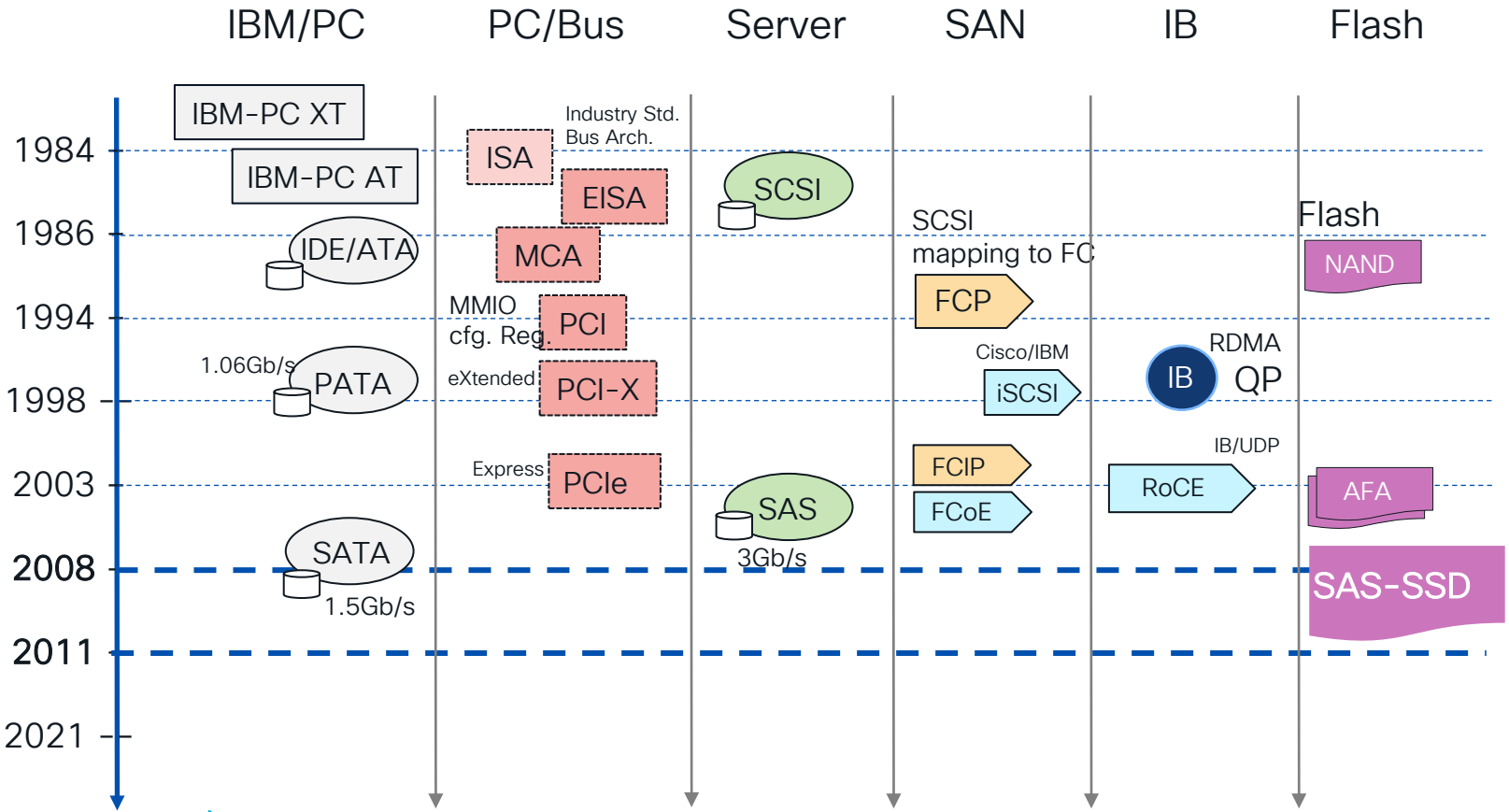
Storage Class Memory -SCM

- PCRAM: Phase Change Random Access Memory (Intel/Optane is based on PCRAM)
- ReRAM: Resistive Random-Access Memory
- MRAM: Magnetic Random-Access Memory
- STT-MRAM: Spin-Transfer Torque Magnetic Random-Access Memory
- Z-NAND: Samsung

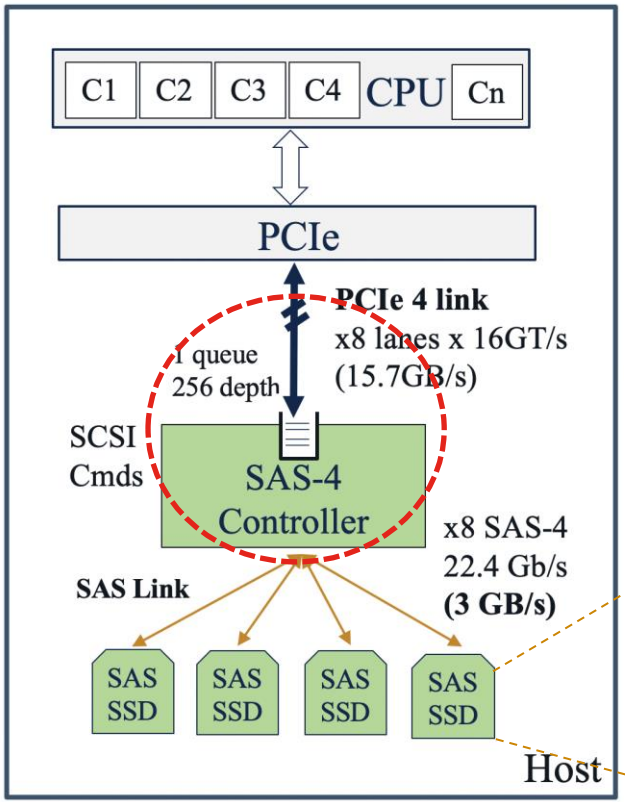
SSD (Solid State Drive)



SSD SAS (Serial Attached SCSI)



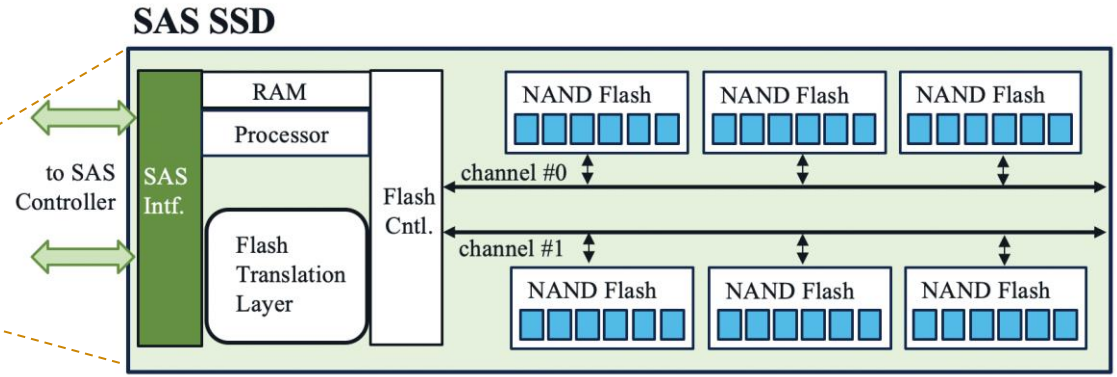
SAS-4 SSD (Maximum Throughput 3GB/s)



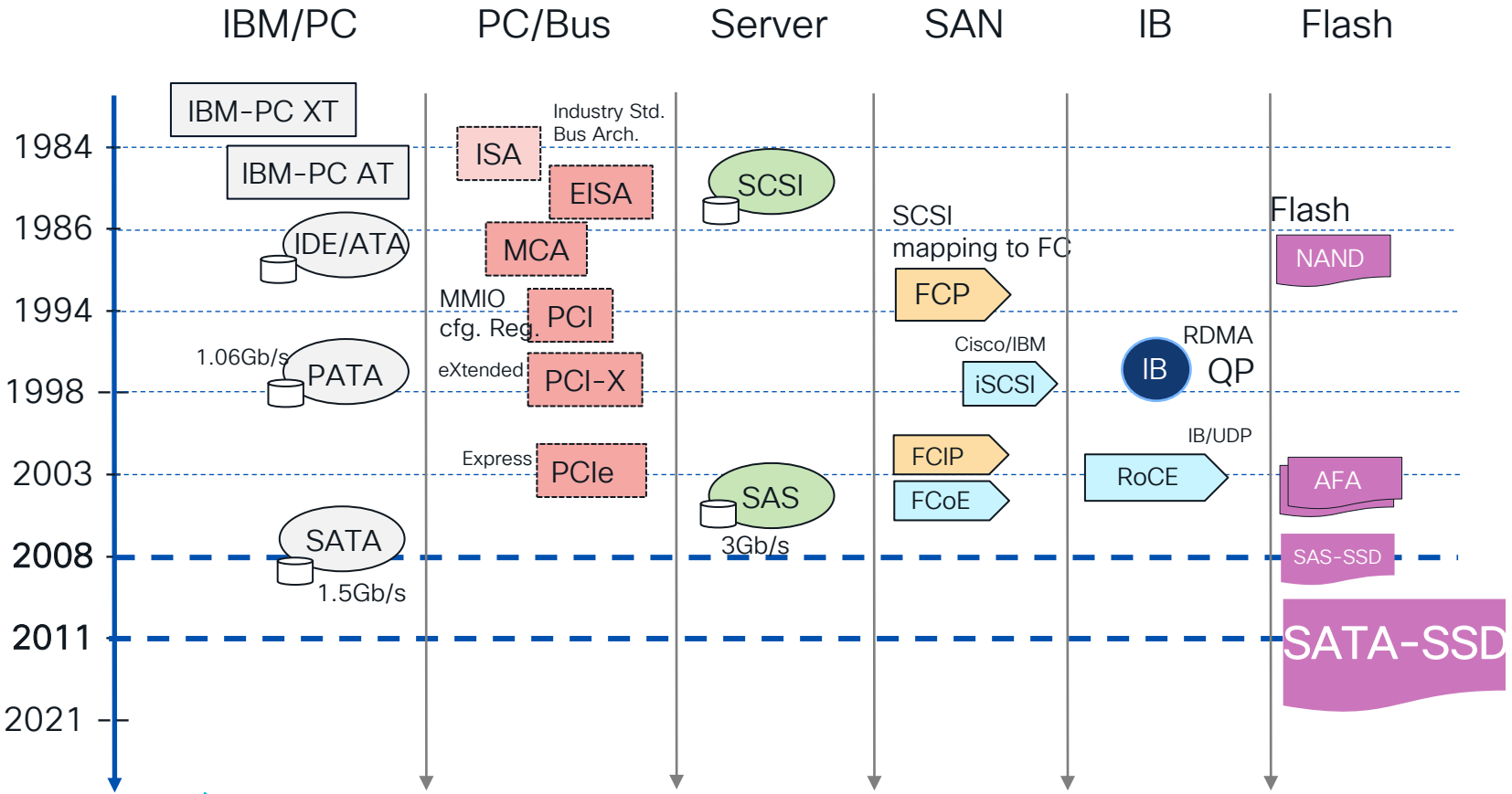
-SCSI Command Set
-SAS Controller/Interface

Limited max. speed

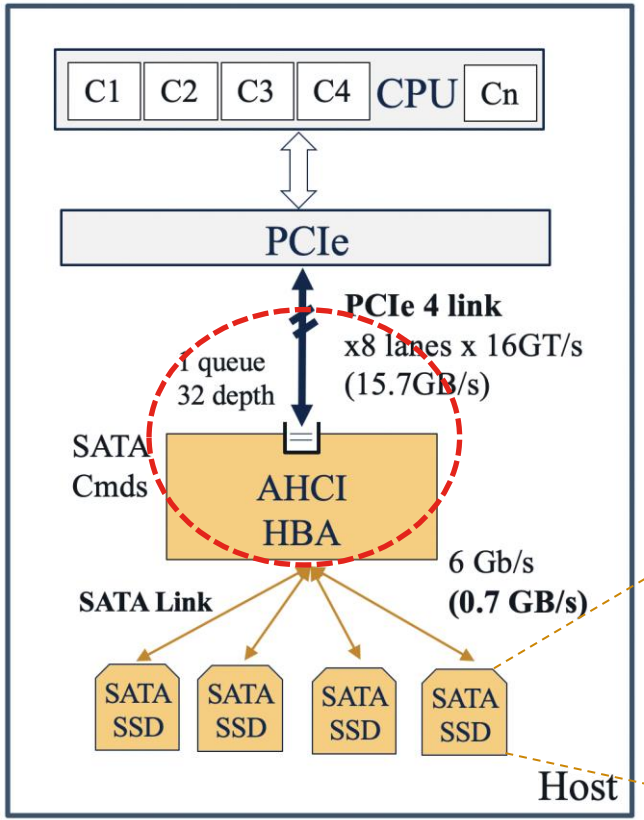
3 Gb/s SAS-1 2004
6 Gb/s SAS-2 2009
12 Gb/s SAS-3 2013
24 Gb/s SAS-4 2017



SSD SATA (Serial ATA)

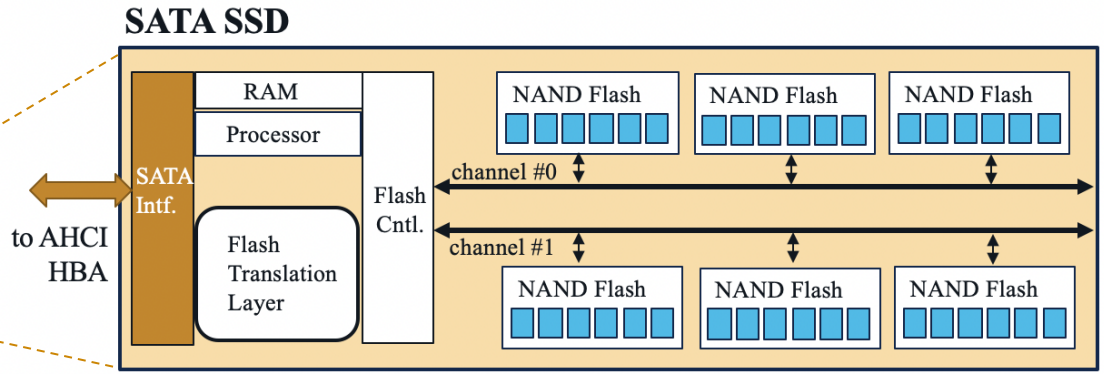


SATA SSD (Maximum Throughput 750MB/s)

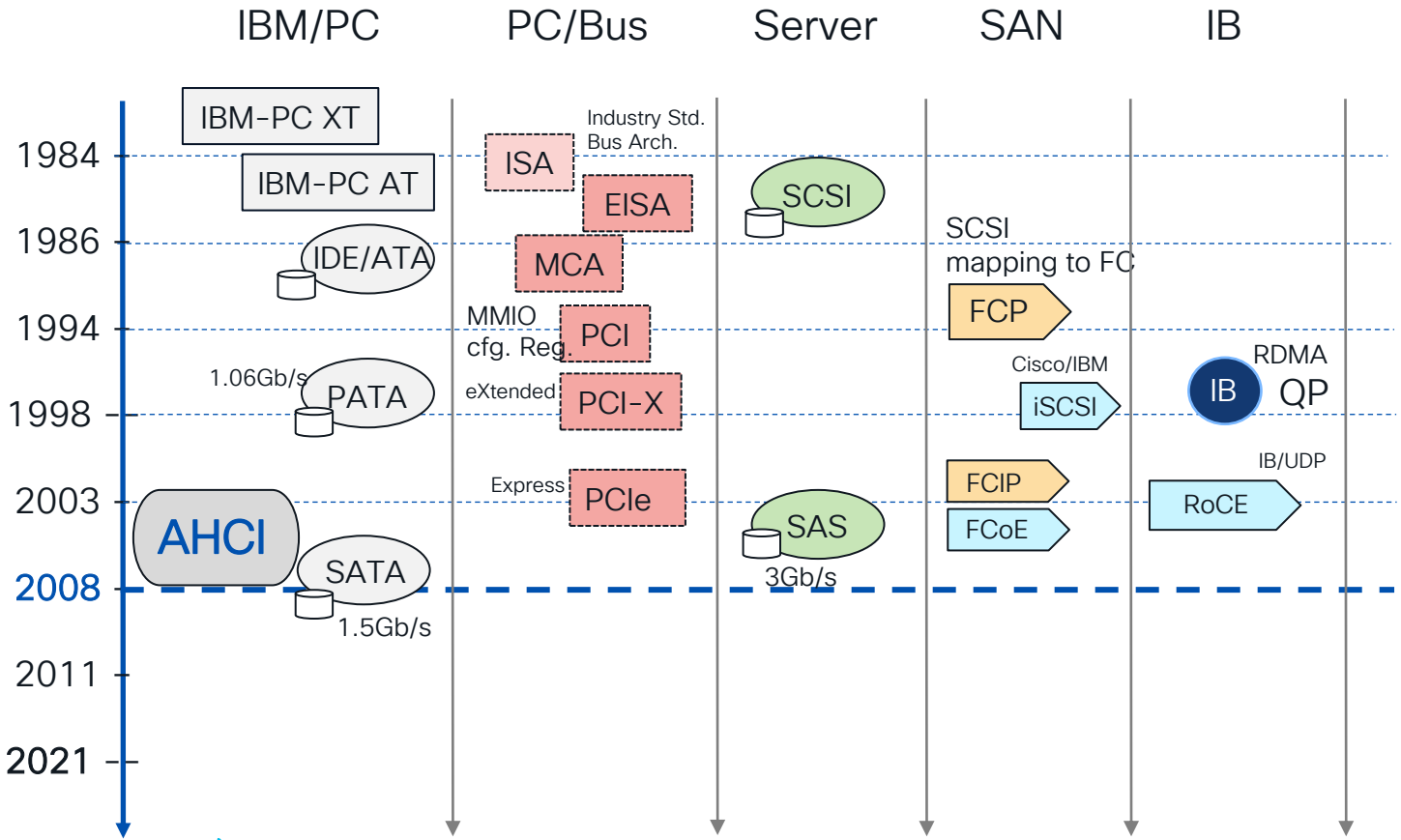


-ATA Command Set
-SATA Controller/Interface

1.5 Gb/s SATA-1 2003
2.0 Gb/s SATA-2 2004 Limited
6.0 Gb/s SATA-3 20[illegible] max. speed
6+ Gb/s SATA-3.2 (SATA Express) 201



AHCI (Advance Host Controller Interface)



Advance Host Cntl. Intf.

"AHCI is a PCI class device that acts as a data movement engine between system memory and Serial ATA device."

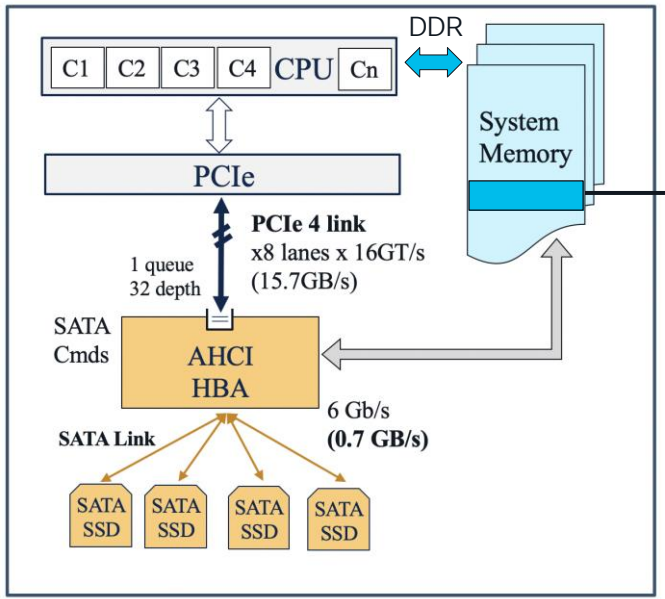
AHCI Advantages

-AHCI device allows **data movement between system memory and SATA device**

-It makes HBA implementation simpler as they are not required to parse ATA commands

-Data transfers between SATA device and system memory uses DMA thus offloading the CPU

-AHCI also



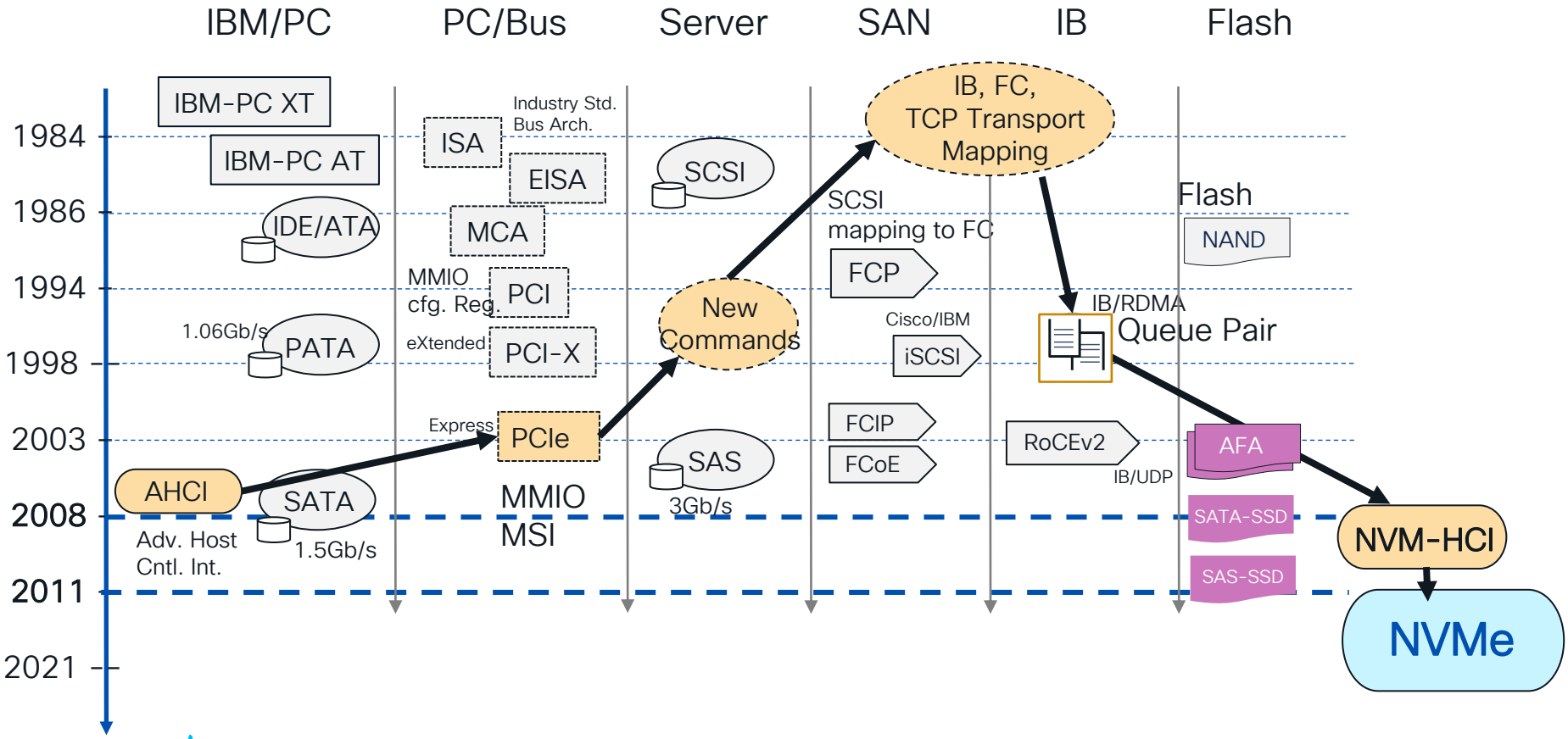
PCIe Register

00h	3Fh	PCI Header
PMCAP	PMCAP+7	PCI Power Mgmt. Capability
MSICAP	MSICAP+9	Msg. Signaled Intr. Capability

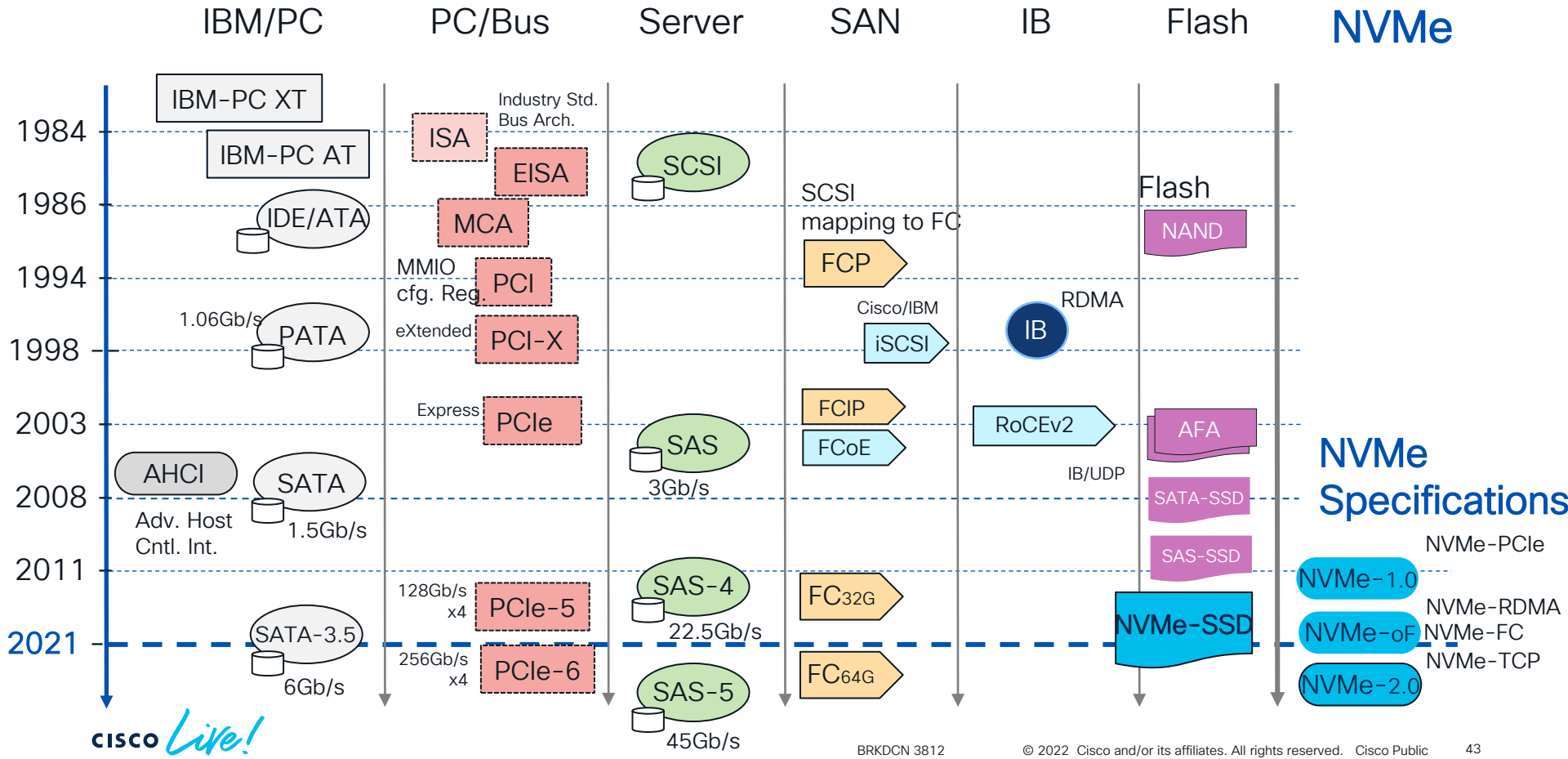
PCIe Header

00-03h	ID	Identifier
04-05h	CMD	Command Register
06-07h	STS	Device Status
08-08h	RID	Revision ID
09-0Bh	CC	Class Code
10-23h	BARS	Base Address Registers 0-4
24-27h	ABAR	AHCI BAR - 05
2C-2Fh	SS	Subsystem Identifiers
34-34h	CAP	Capability Pointer

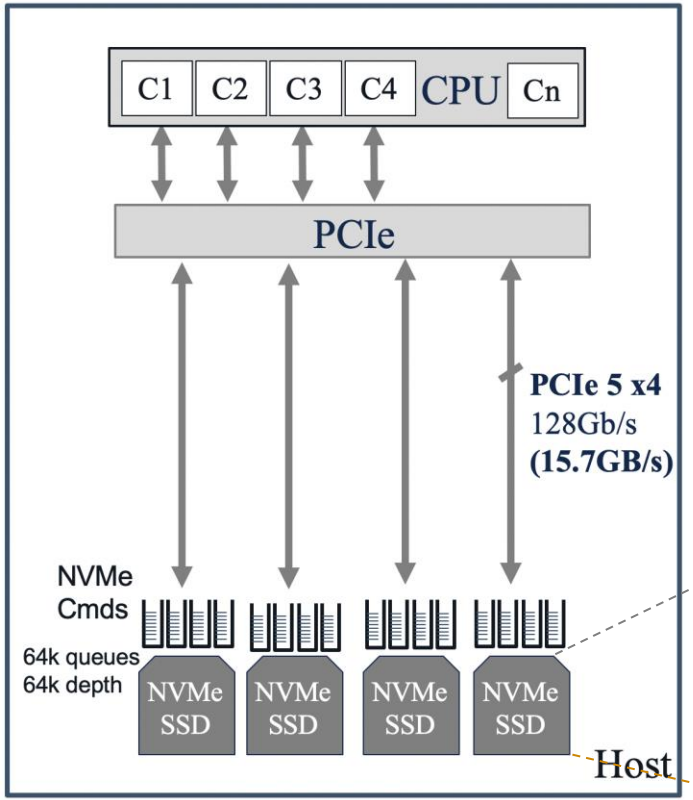
Best of all worlds....



NVMe (Non Volatile Memory Express)



NVMe SSD (15GB with PCIe-5)



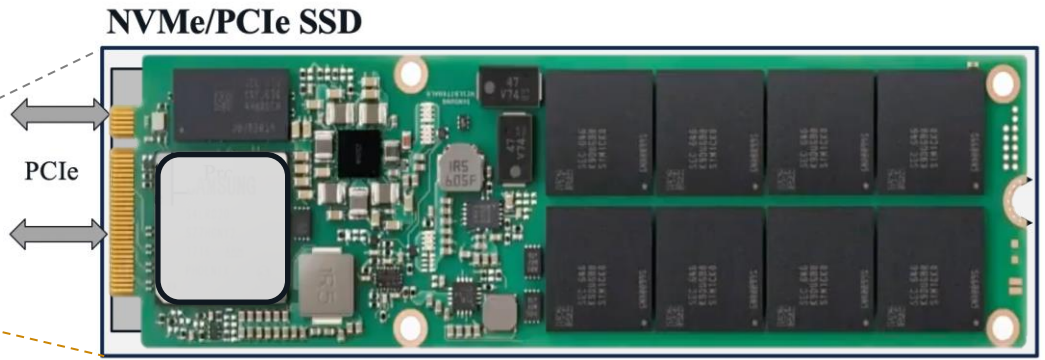
NVMe

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

Fabric Command
- Connect/Disconnect
- Set/Get Property

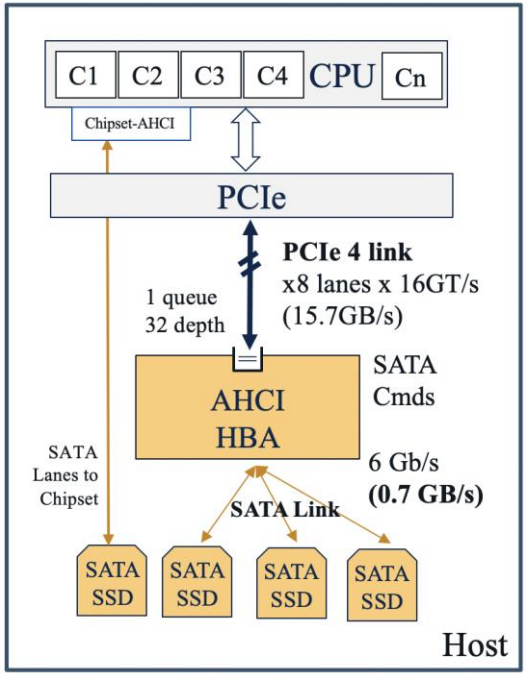
I/O Command
- Read/Write
- Flush

Admin Command
- Create/Delete I/O SC
- Create/Delete I/O CC
- Get Log Page
- Identify
- Abort
- Set/Get Feature
- Async. Event Reques

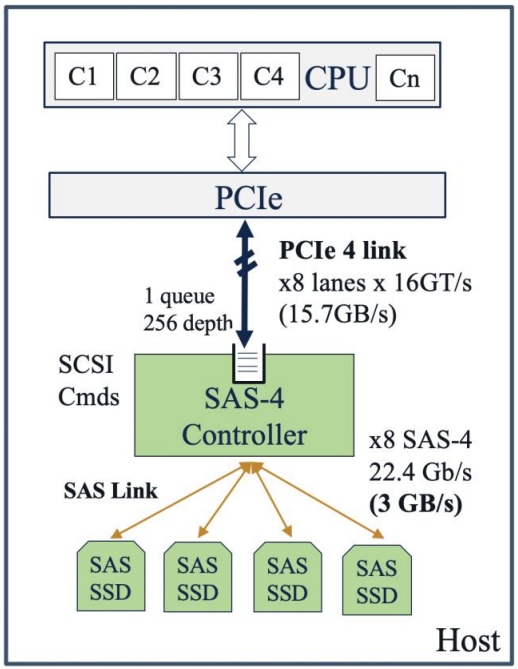


M.2 form factor

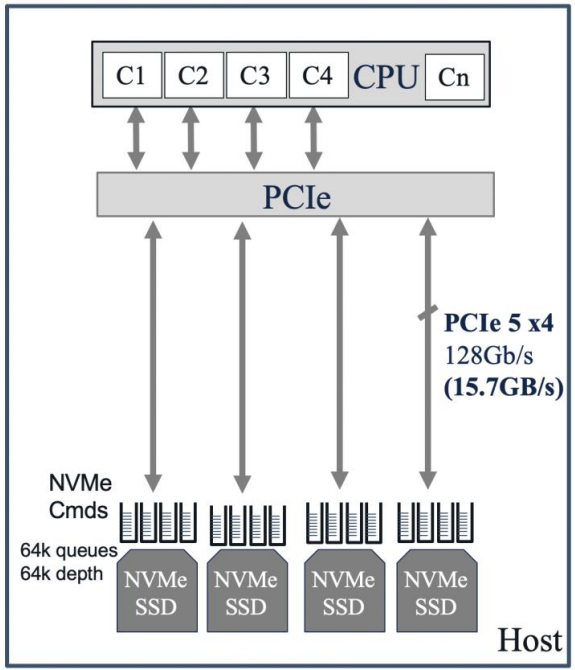
SATA, SAS, PCIe/NVMe



SATA-SSD



SAS-SSD



PCIe/NVMe-SSD

Best Practices (Do's & Don'ts)

-Higher "Levels" of cell usually provide less reliability due to higher voltages levels requirements.

SLC -one voltage levels 2^1 (Single Level Cell)
TLC -eight voltage levels 2^3 (Triple Level Cell)
QLC -sixteen voltage levels 2^4 (Quad Level Cell)
PLC -thirty two voltage levels 2^5 (Penta Level Cell)

-Intel Optane & Samsung Z-SSD provide the highest performance & lowest latency, but the prices are on the high end.

-Flash Drive Endurance determine the total amount of data that can be written

$$\text{Drive Endurance} = \frac{\text{Flash cell endurance}}{\text{STF x AT x WAF}}$$

Flash Cell Endurance = maximum P/E cycles (program erase)

STF = Storage Time Factor (length of time in storage)
AT = Acceleration factor for Temperature
WAF = Write Amplification Factor

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

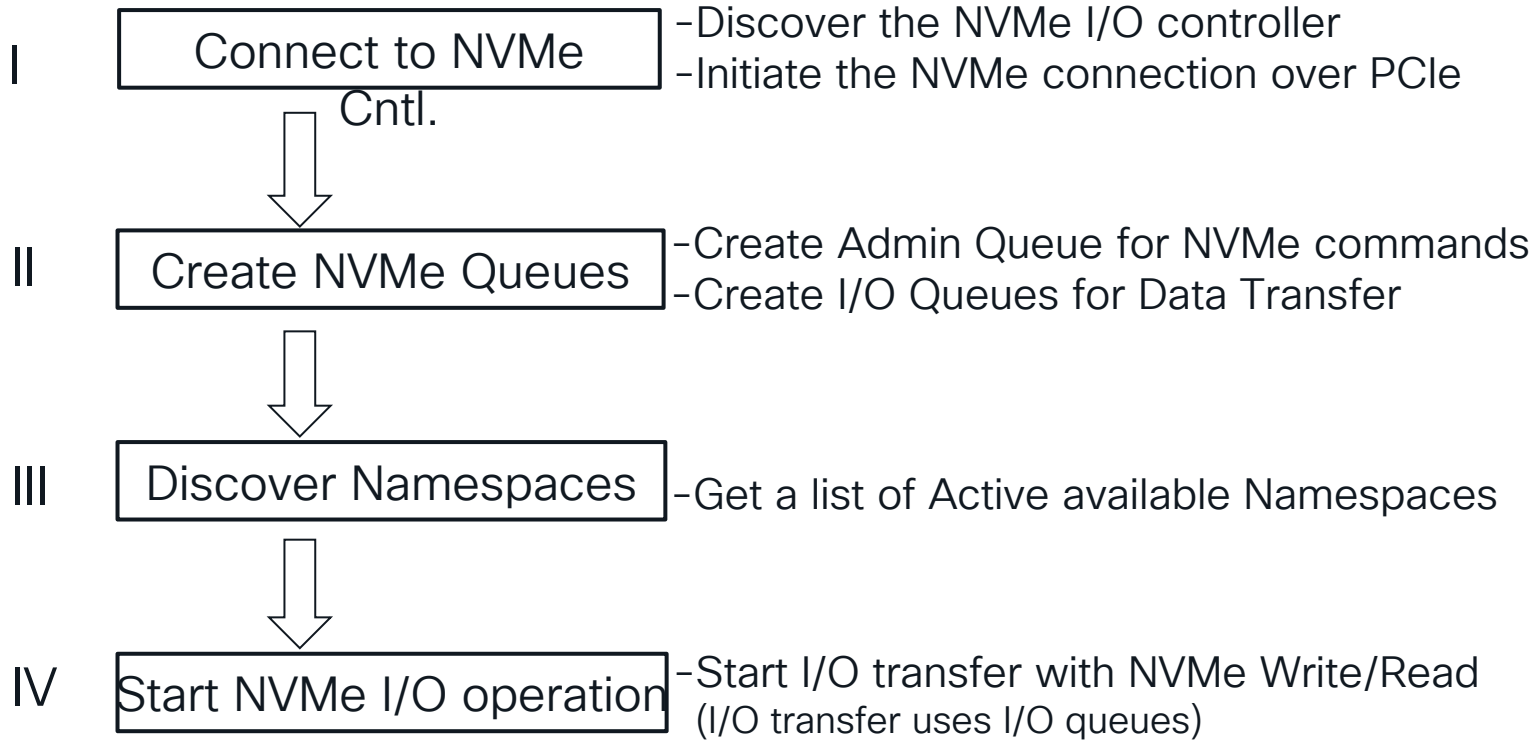
3-NVMe Transport Options (FC, TCP, RoCEv2)

4-NVMe Datacenter Design

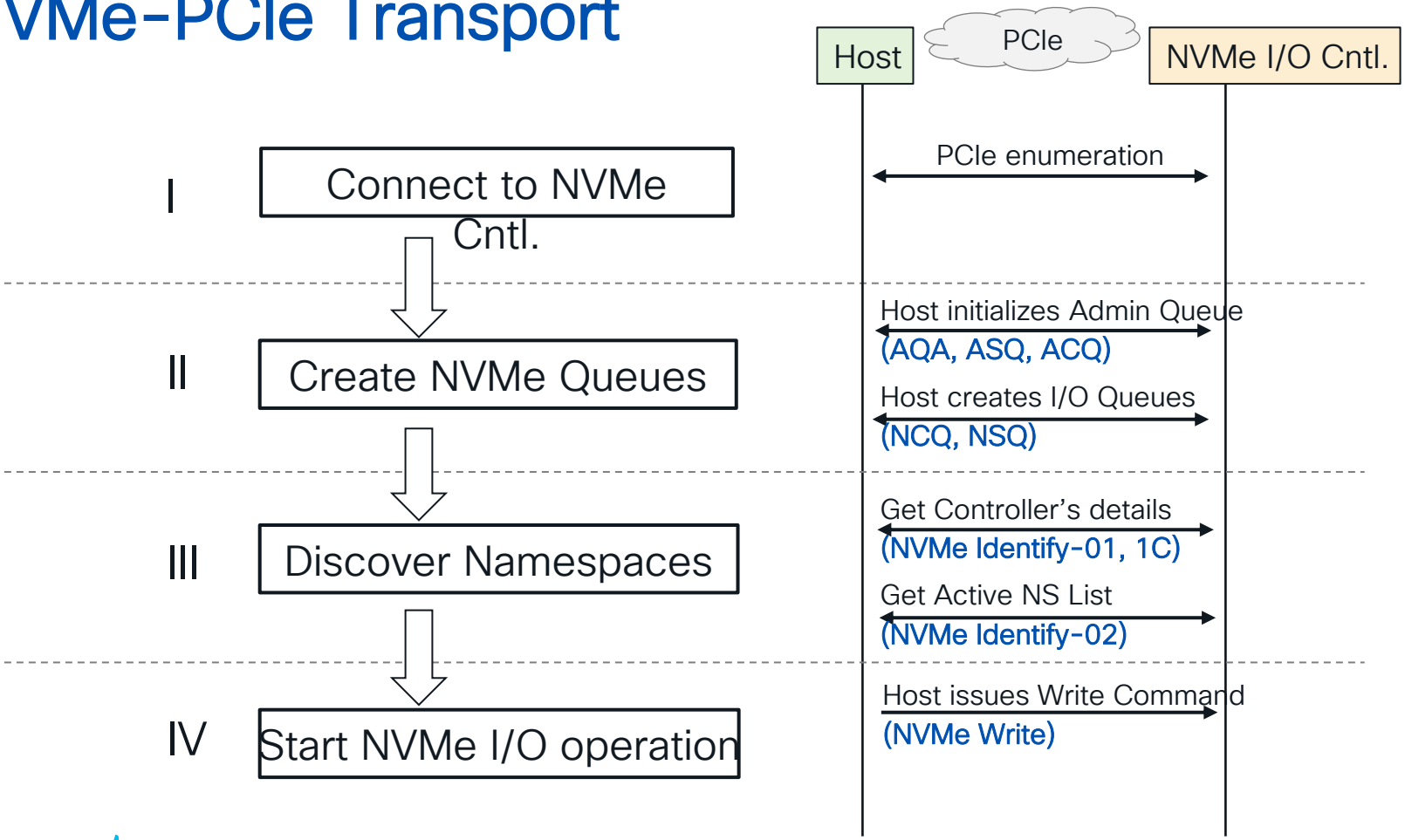
5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

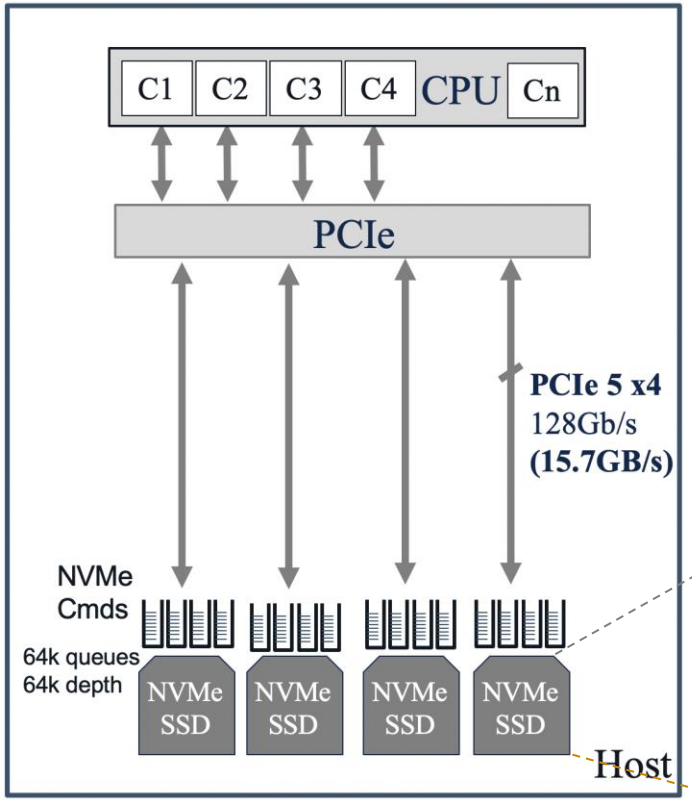
NVMe-PCIe Transport



NVMe-PCIe Transport

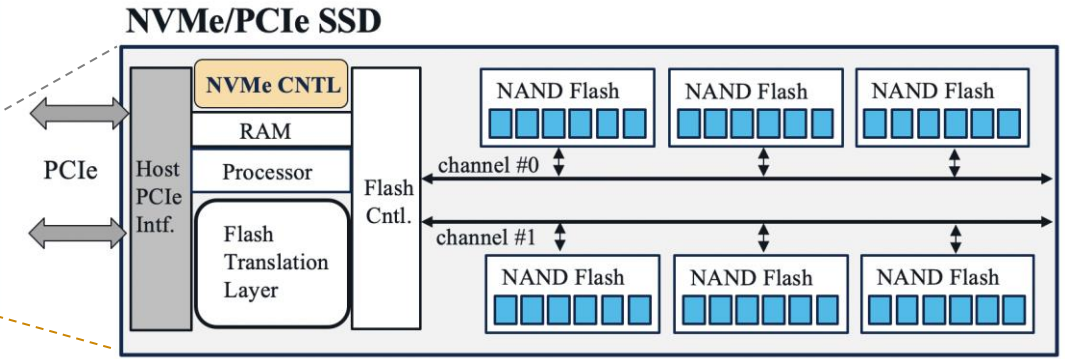


NVMe-PCIe Transport

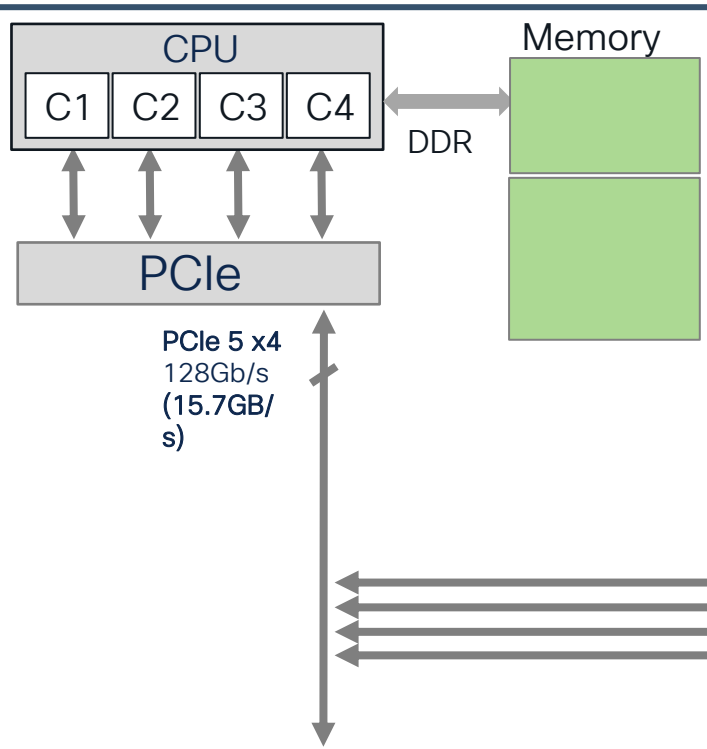


The PCIe transport provides reliable mechanisms for memory mapped data transfer of Admin and I/O command data through memory mapped I/O transactions.

....NVMe-PCIe spec. 1.0

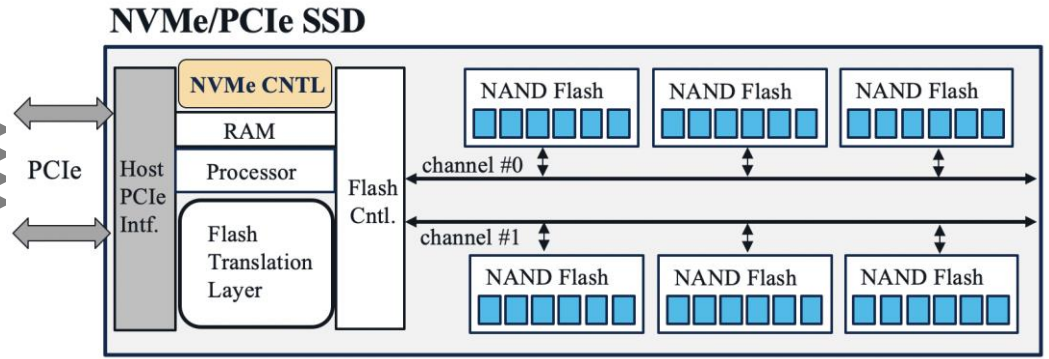


NVMe-PCIe Transport



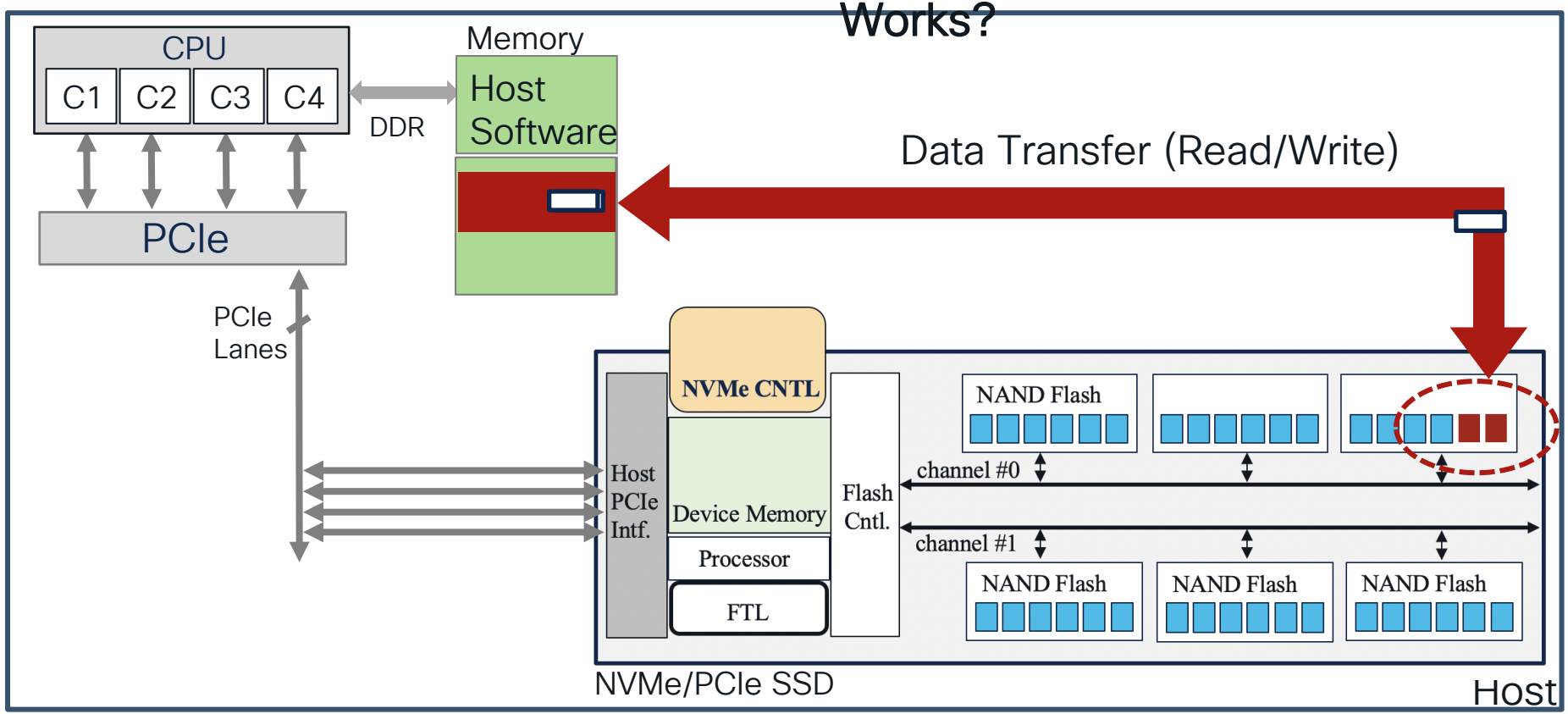
Building Blocks of NVMe

- PCIe Config Registers
- BAR Address, Capability Pointers
- Messaged Signaled Interrupt, Doorbell
- NVMe Queues, Admin/IO (SQ/CQ)
- NVMe Subsystem/Controller



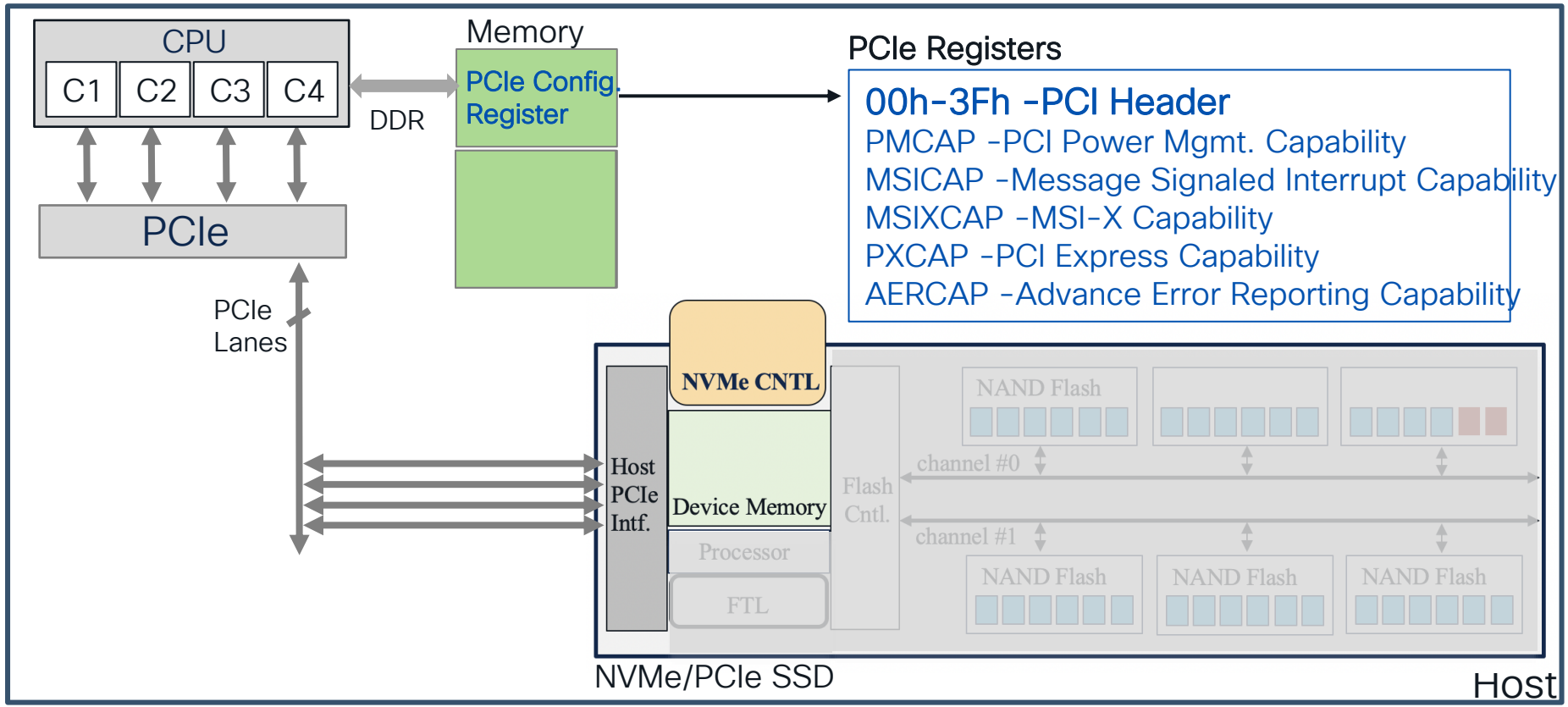
NVMe-PCIe Transport

How does Data Transfer Works?



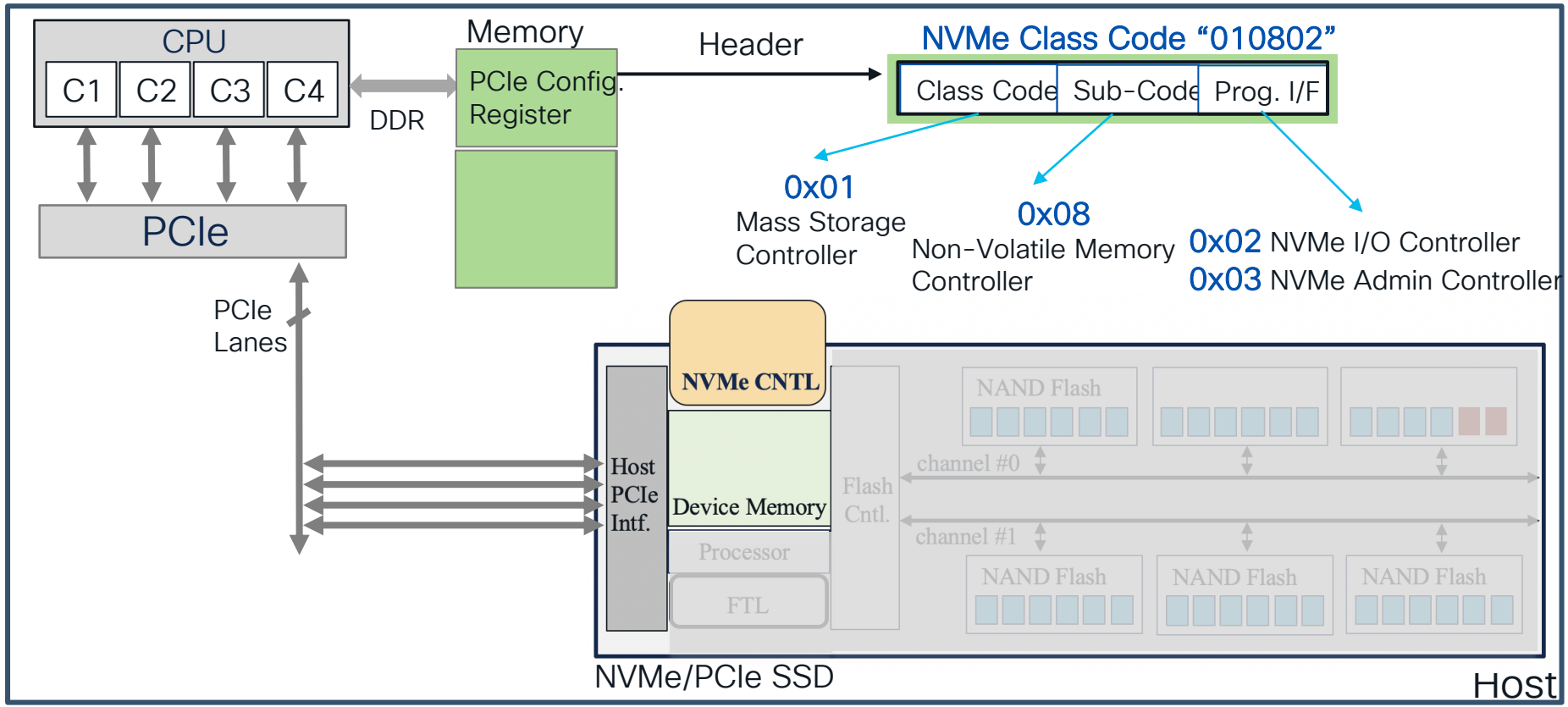
NVMe-PCIe (Registers)

PCIe devices have set of registers that are mapped to memory locations



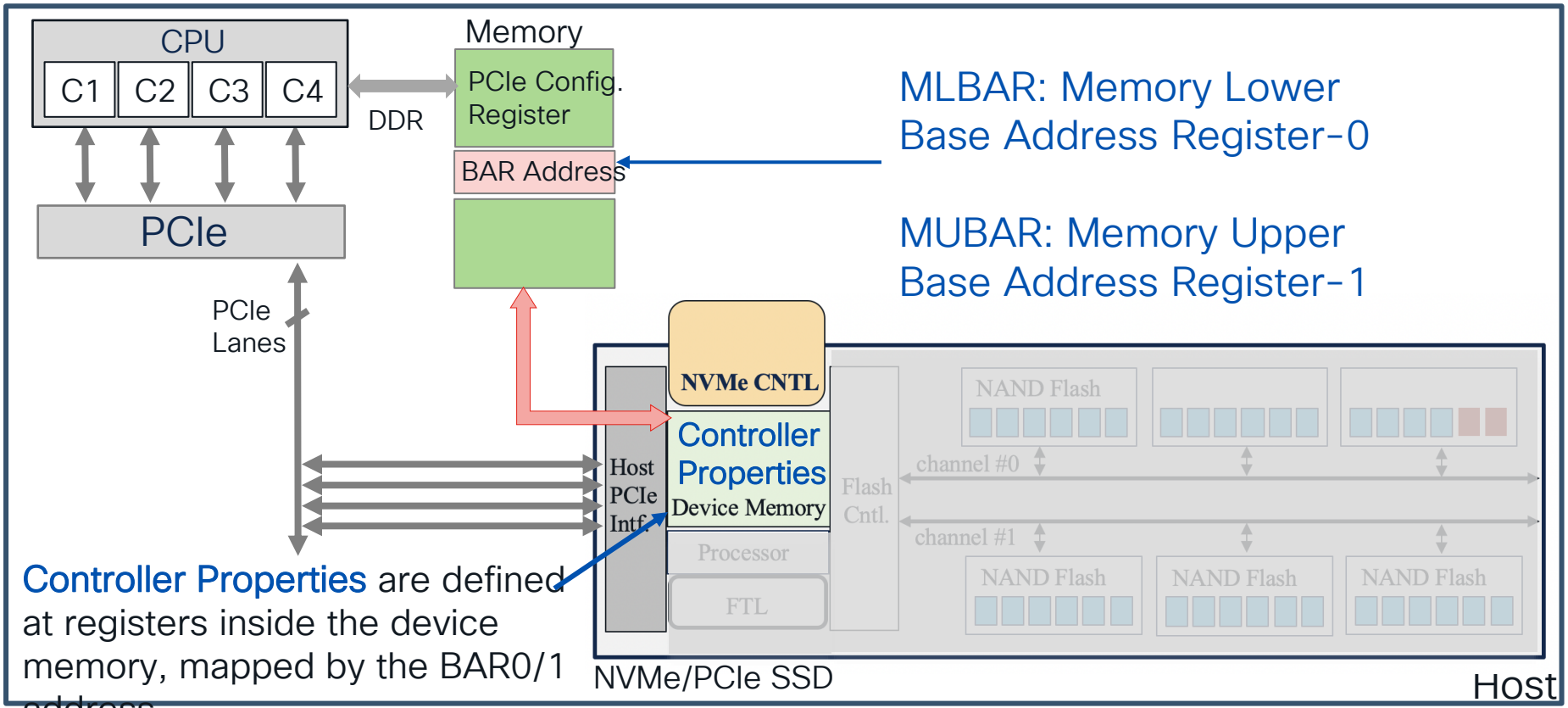
NVMe-PCIe (Registers)

During PCIe enumeration
"Class Code" is read



NVMe-PCIe (Registers)

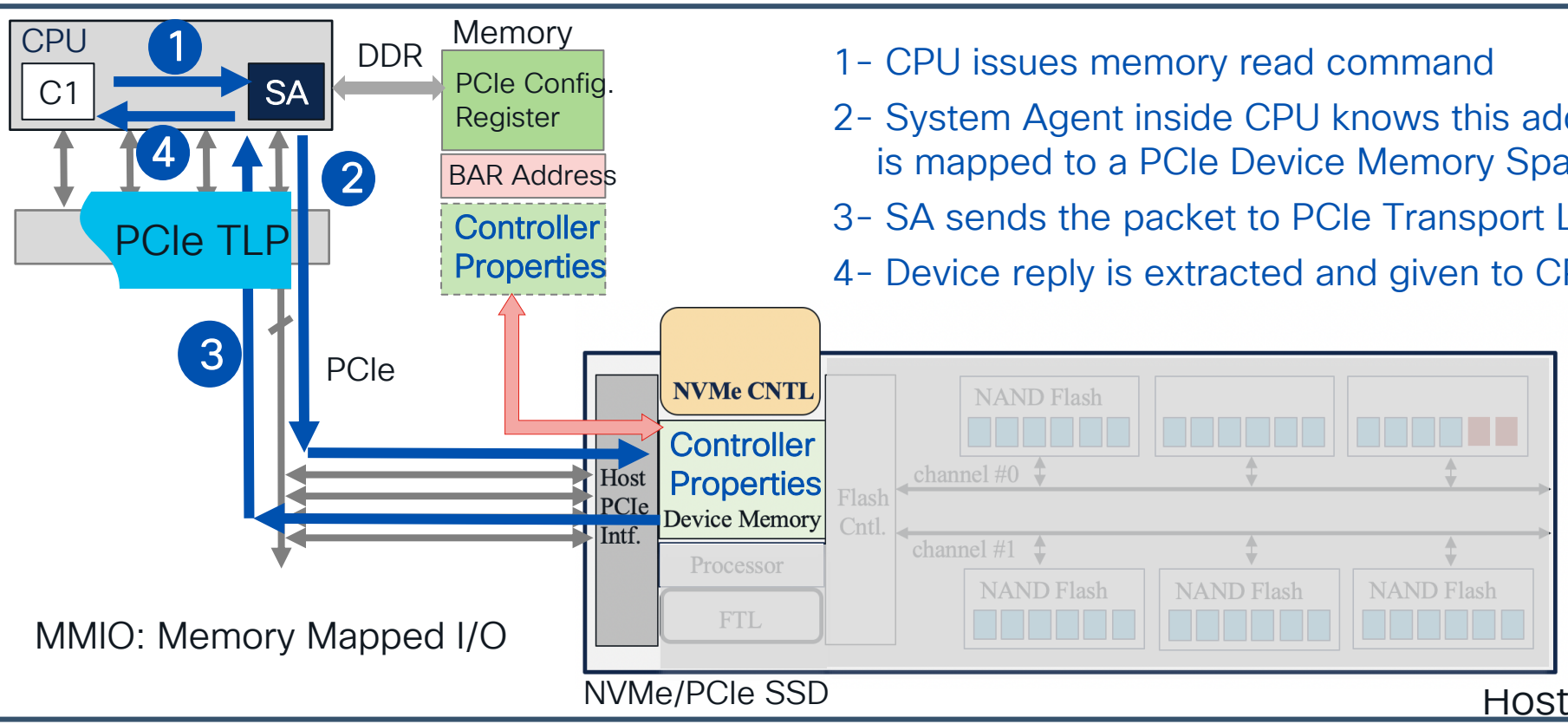
BAR registers maps Device Memory Registers into CPU memory



Controller Properties are defined at registers inside the device memory, mapped by the BAR0/1 address

NVMe-PCIe (Properties)

How does CPU reads the Controller Properties Register ?



1- CPU issues memory read command

2- System Agent inside CPU knows this addre
is mapped to a PCIe Device Memory Space

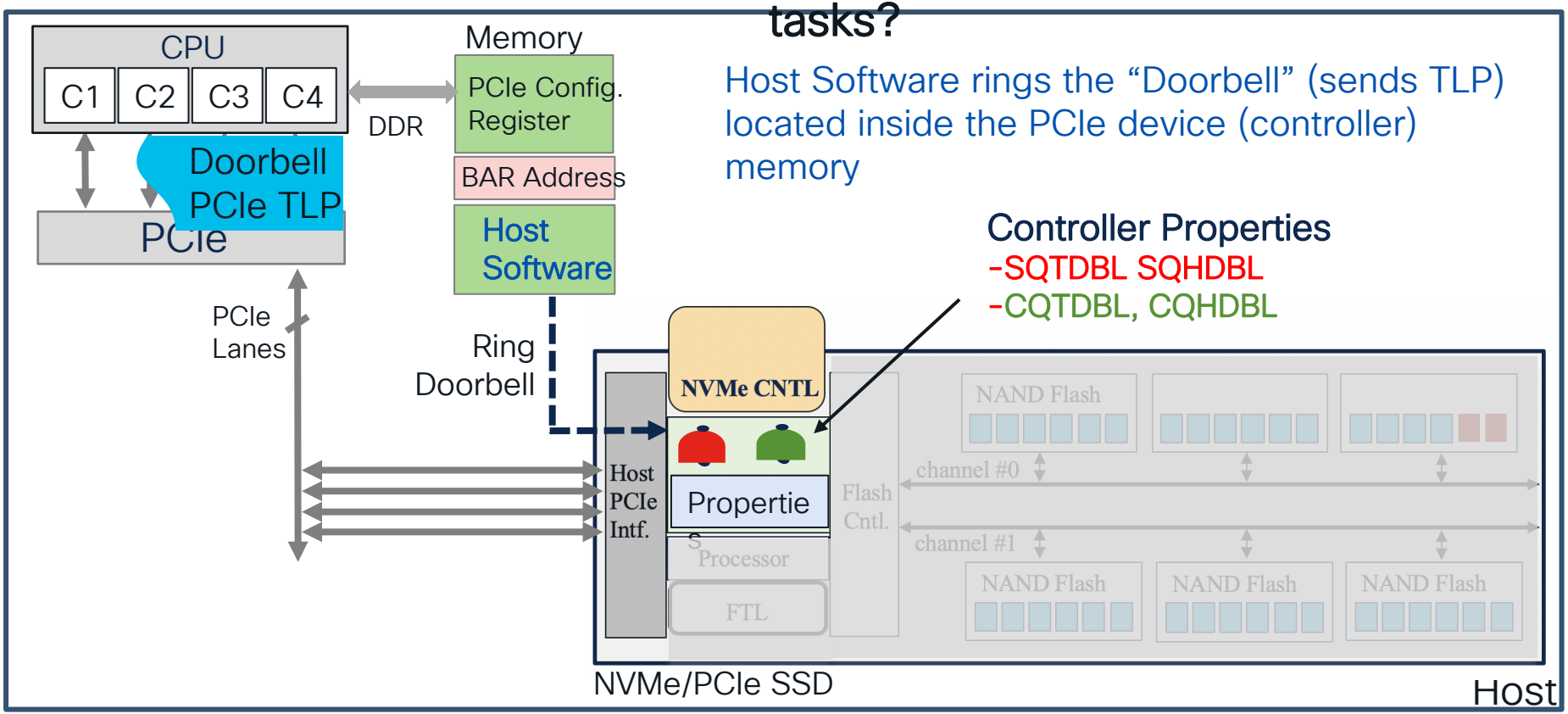
3- SA sends the packet to PCIe Transport Lay

4- Device reply is extracted and given to CPU

NVMe-PCIe (Doorbell)

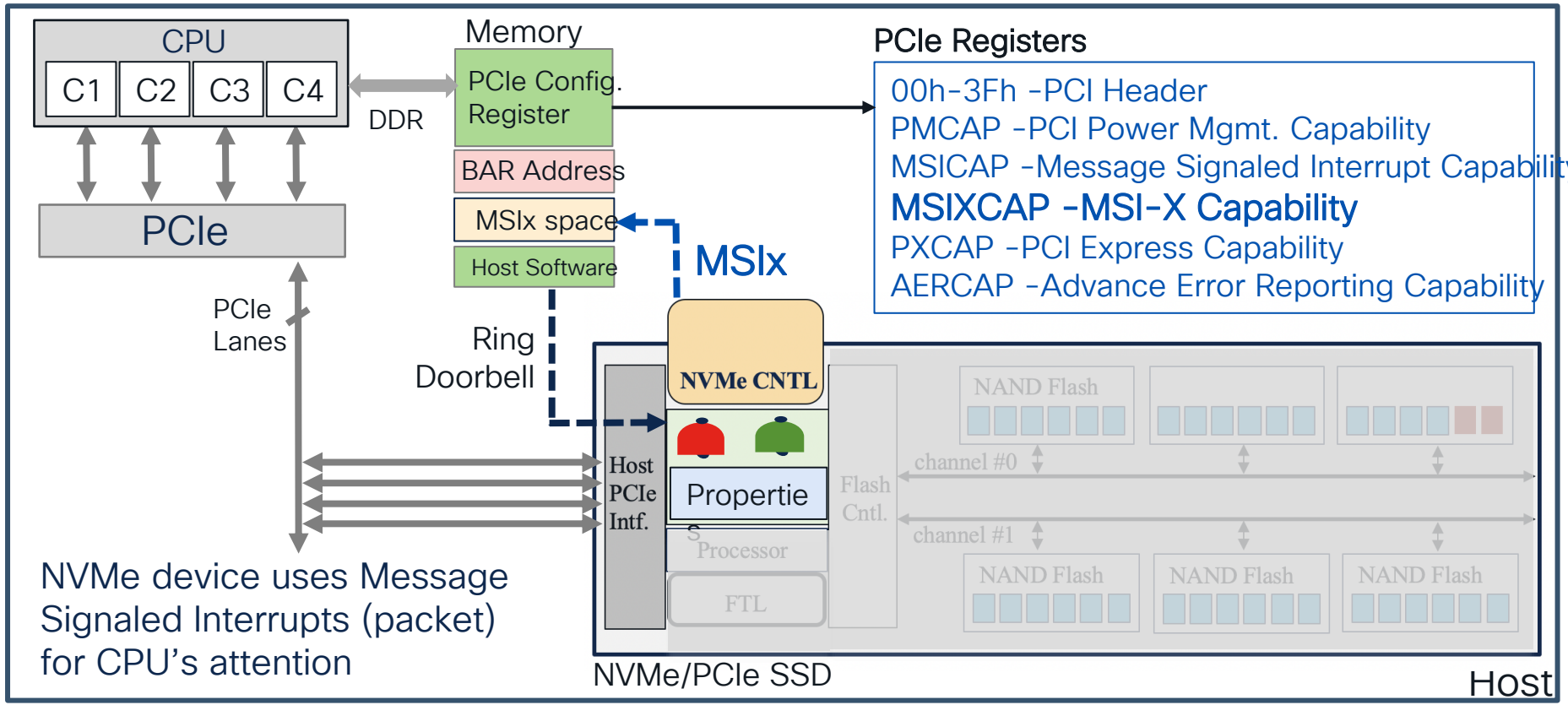
How does "Host Software" informs Controller about pending tasks?

Host Software rings the "Doorbell" (sends TLP) located inside the PCIe device (controller) memory



NVMe-PCIe (MSIx)

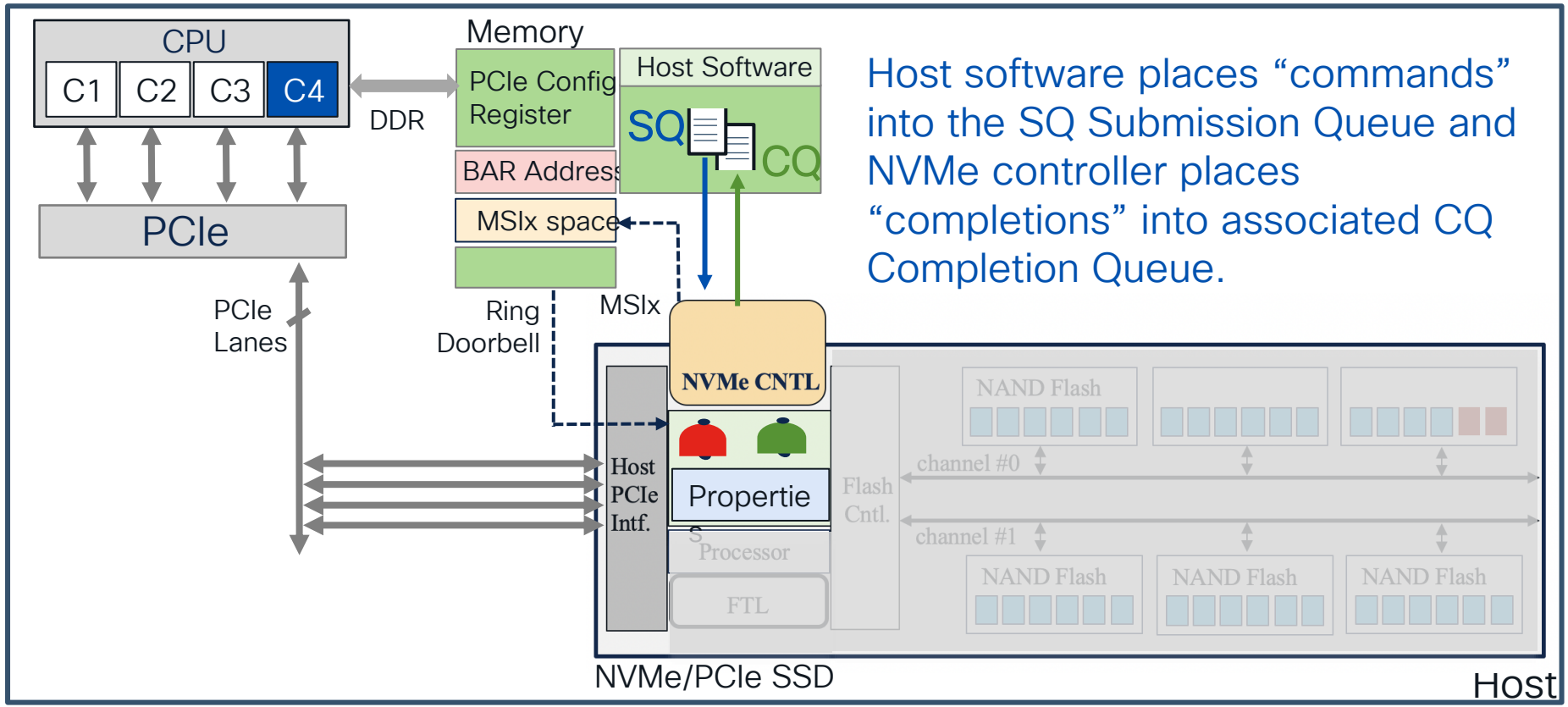
How does "Controller" informs Host about pending tasks ?



NVMe device uses Message Signaled Interrupts (packet) for CPU's attention

NVMe-PCIe (SQ/CQ Pair)

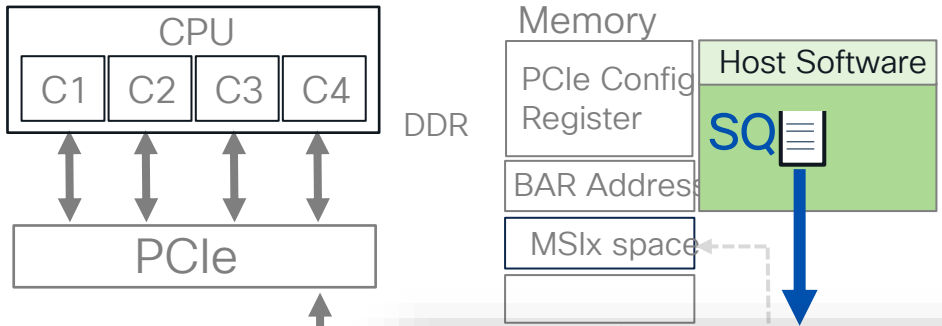
NVMe is based on a paired Submission and Completion Queue mechanism.



Host software places “commands” into the SQ Submission Queue and NVMe controller places “completions” into associated CQ Completion Queue.

NVMe-PCIe (SQE)

Submission Queue Entry



NVMe SQ Admin Commands 64B -Queue #0

NVMe/PCIe (SQE 64 bytes)

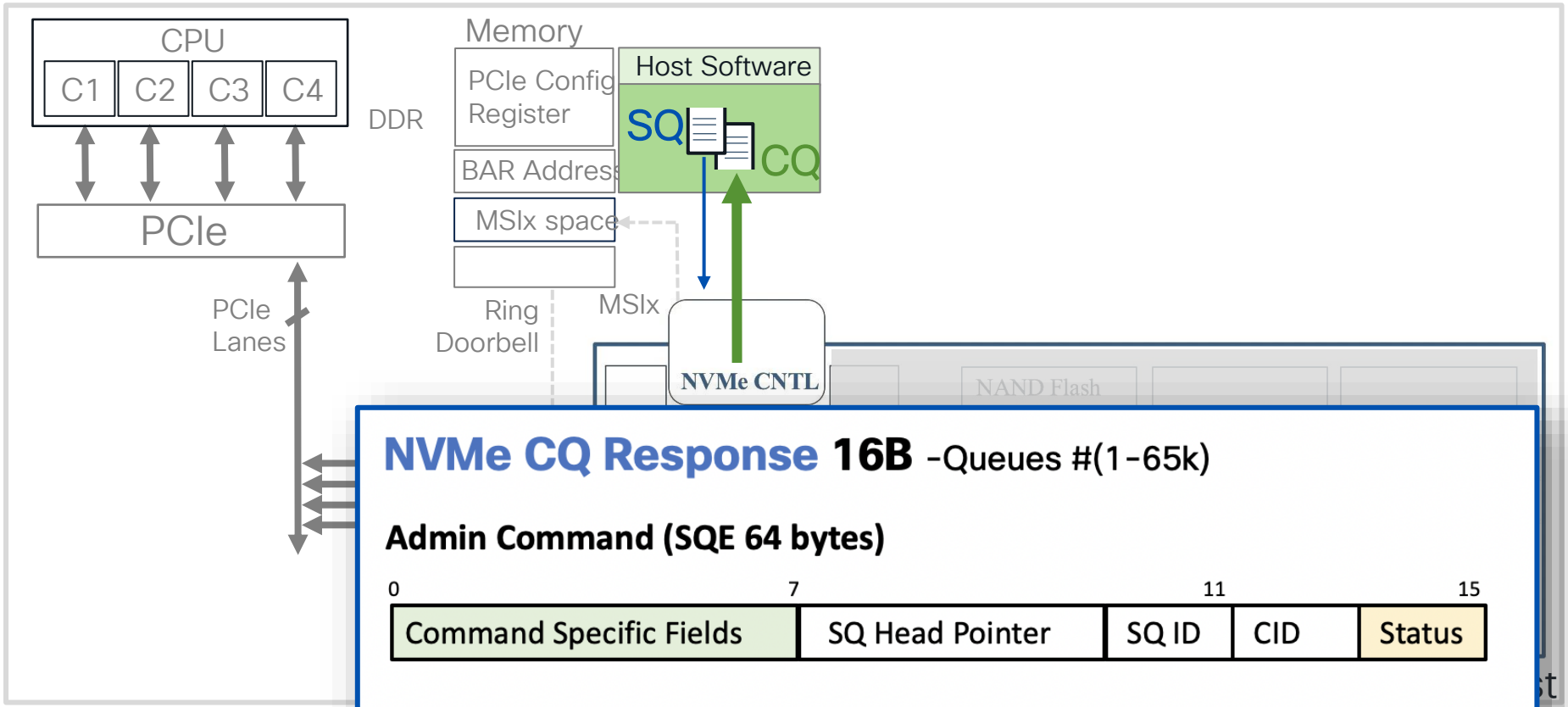
0	4	8	16	24	40	63
Opcode	FUSE	PSDT	CID	NSID	DW2-3	MPTR	DPTR/SGL/PRP	DW 10-15

-(06) Identify
-(0A) Get Feature
-(09) Set Feature
-(02) Get Log Page
-(0C) Async. Event Req.
-(08) Abort

-(01) Create I/O SQ
-(00) Delete I/O SQ
-(05) Create I/O CQ
-(04) Delete I/O CQ

NVMe-PCIe (CQE)

Completion Queue Entry



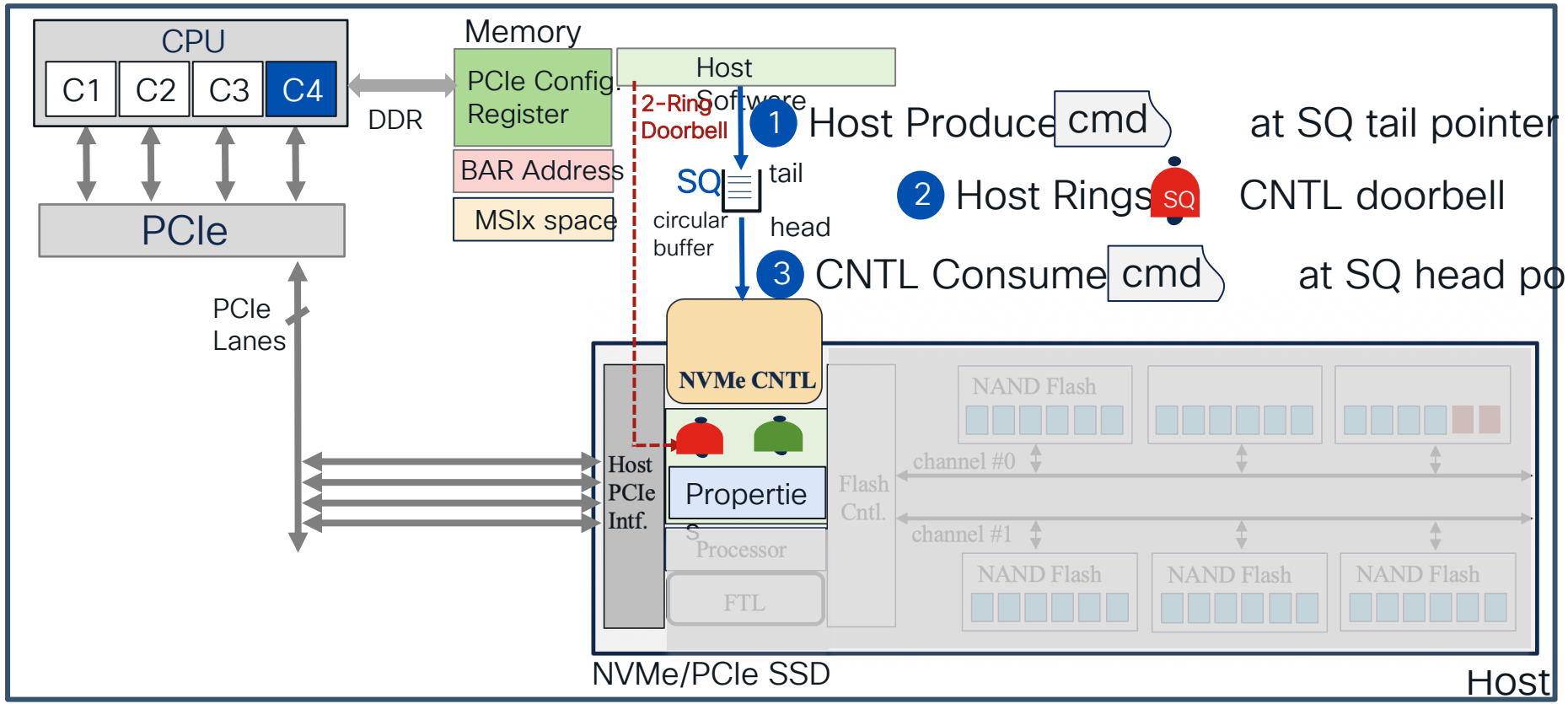
NVMe CQ Response 16B -Queues #(1-65k)

Admin Command (SQE 64 bytes)

0	7	11	15
Command Specific Fields	SQ Head Pointer	SQ ID	CID	Status

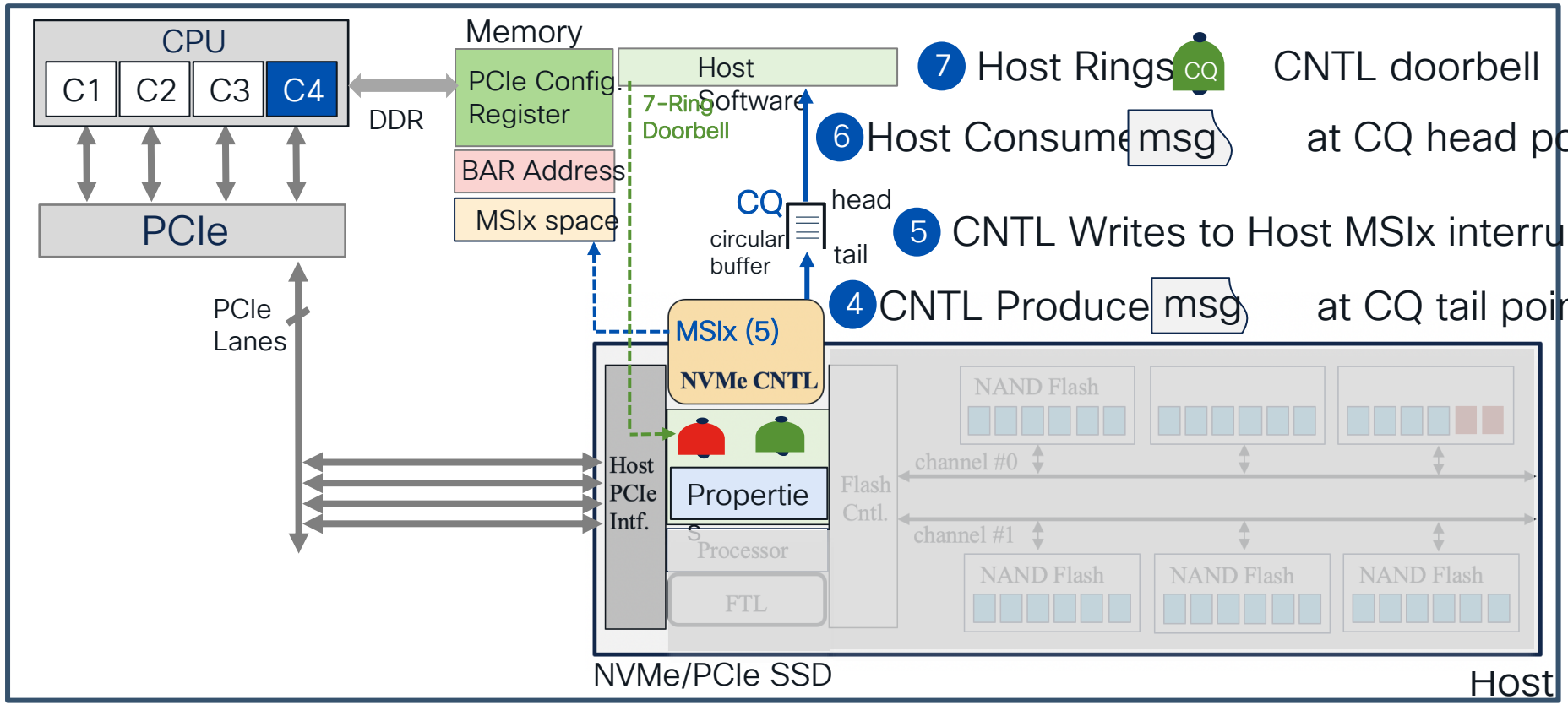
NVMe-PCIe (Host to CNTL)

NVMe Queuing mechanism details



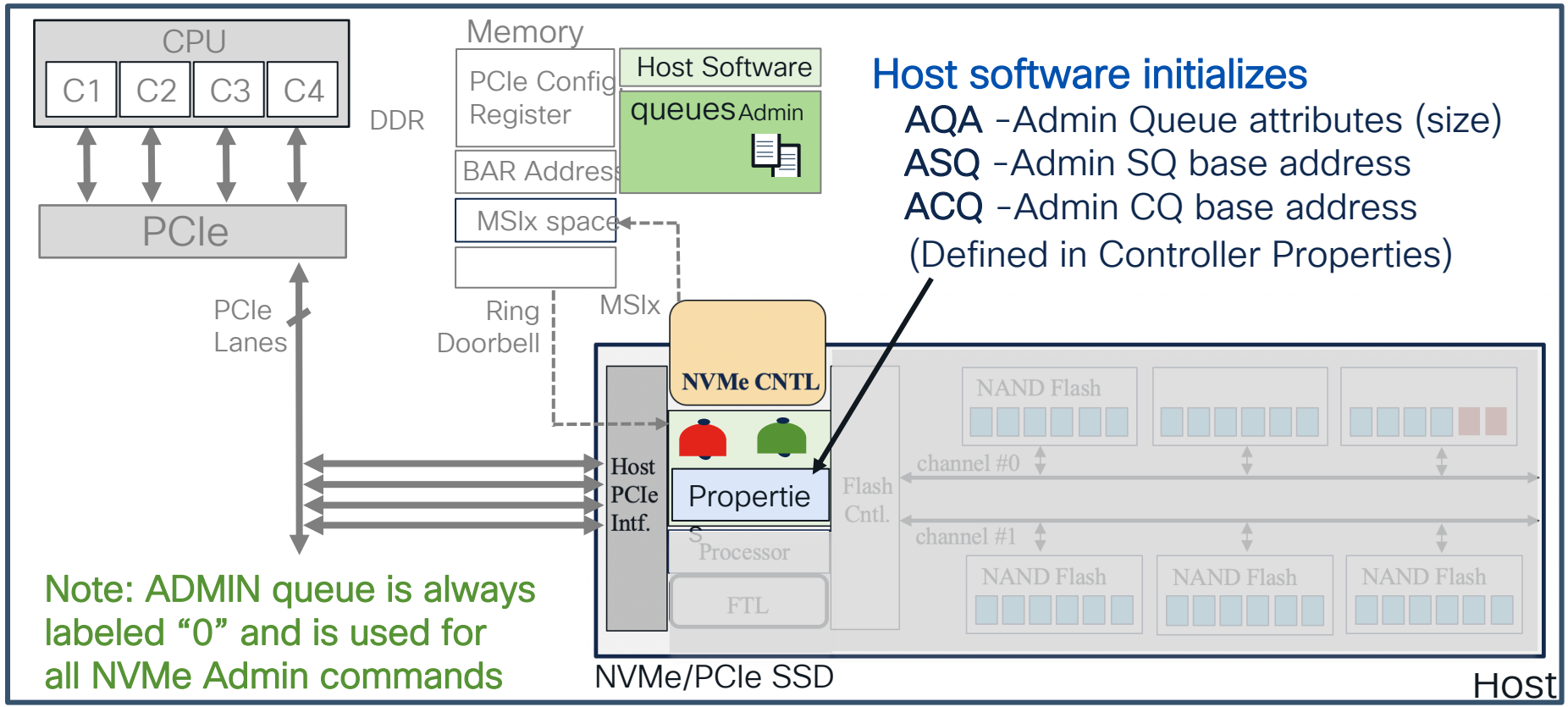
NVMe-PCIe (CNTL to Host)

NVMe Queuing mechanism details



NVMe-PCIe (Admin_Q)

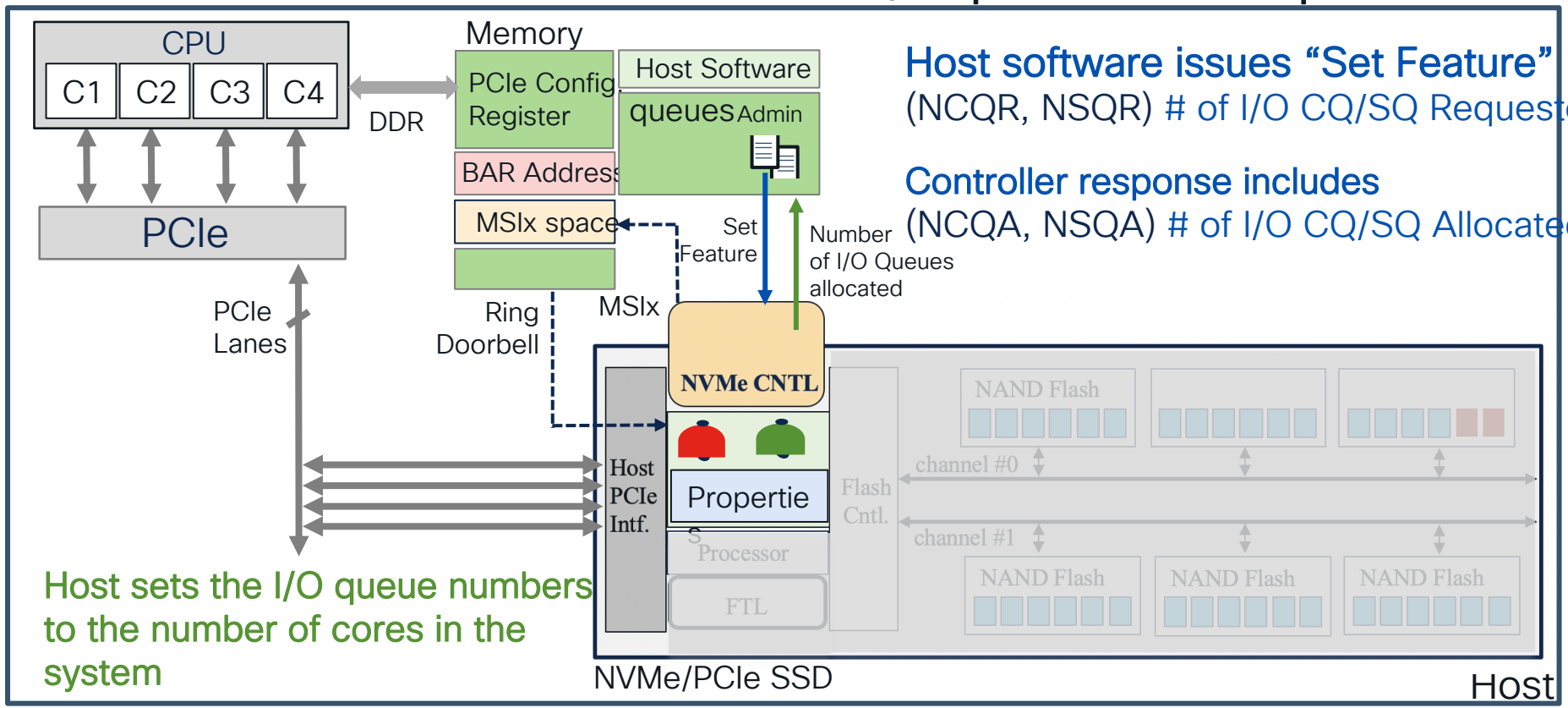
Admin queues are created first and are used for "Administrative Tasks"



Note: ADMIN queue is always labeled "0" and is used for all NVMe Admin commands

NVMe-PCIe (I/O queues)

Using Admin Queues Host starts the I/O queues creation process



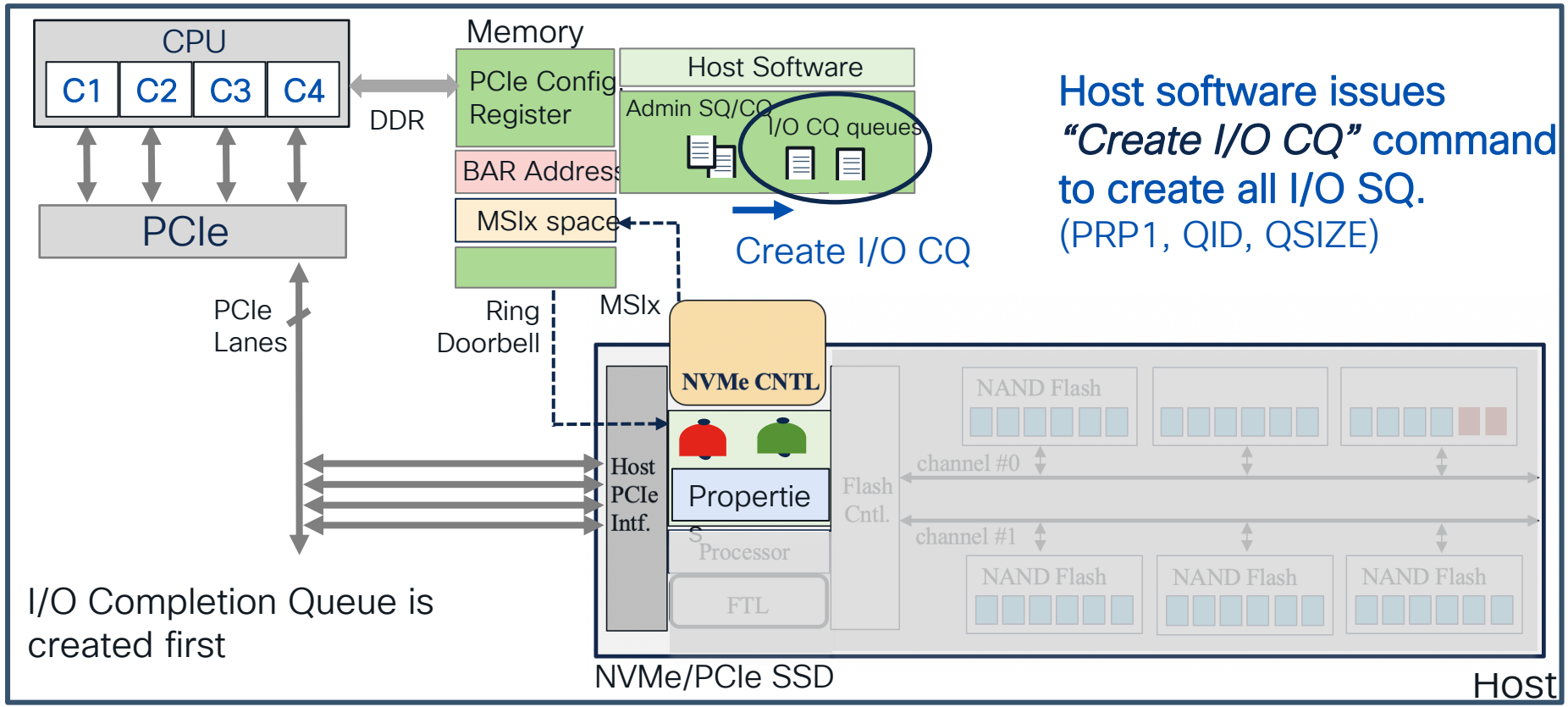
Host software issues "Set Feature" (NCQR, NSQR) # of I/O CQ/SQ Requested

Controller response includes (NCQA, NSQA) # of I/O CQ/SQ Allocated

Host sets the I/O queue numbers to the number of cores in the system

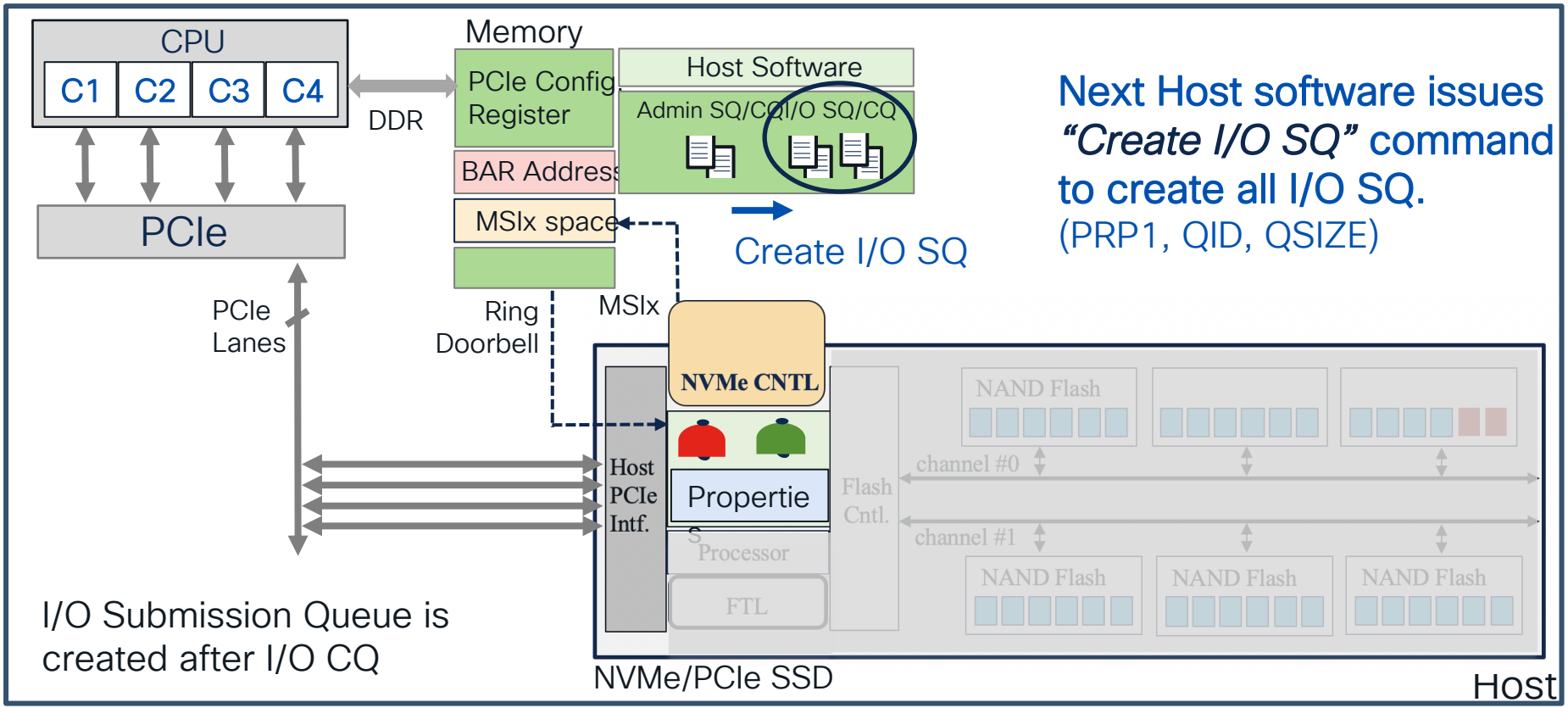
NVMe-PCIe (I/O queues)

Primary purpose for I/O queues is to transfer data (read/write)



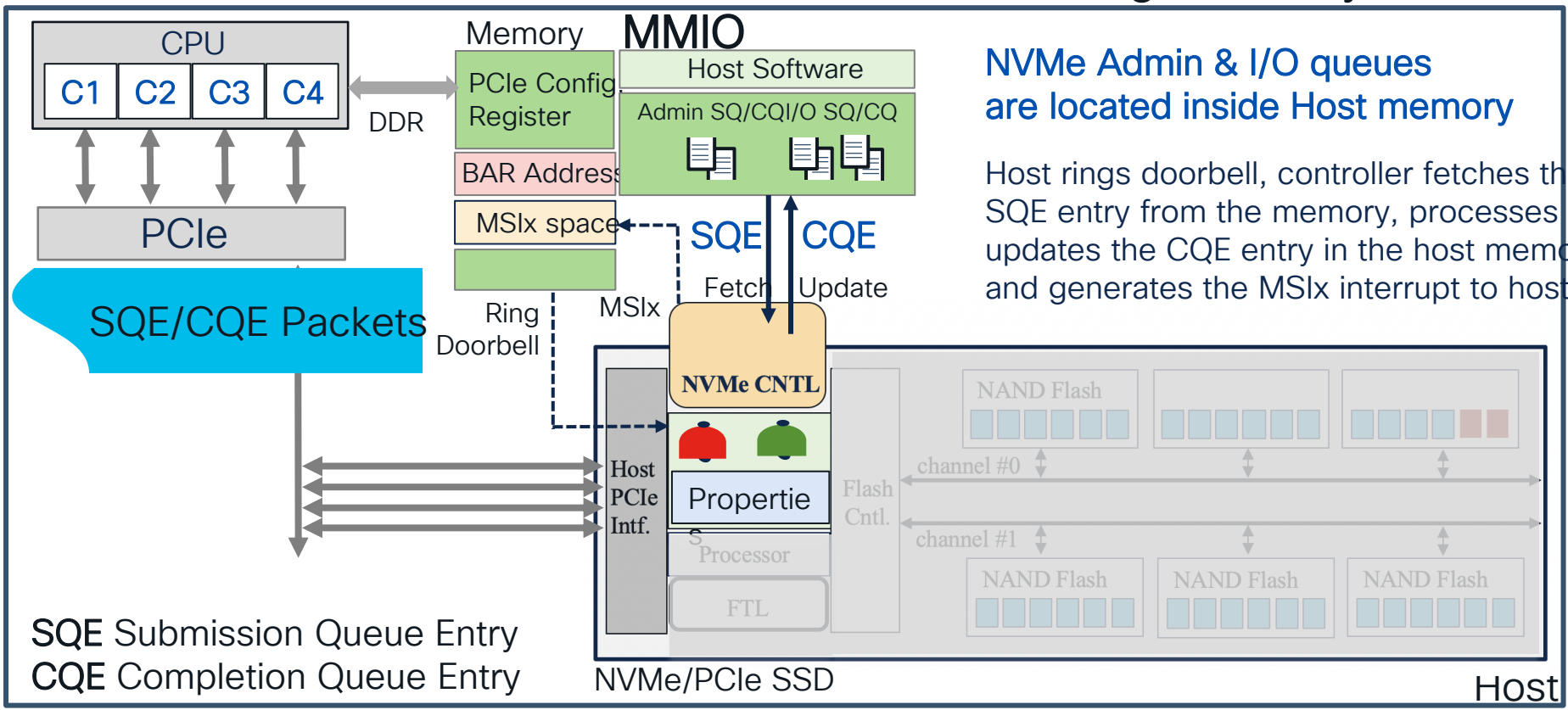
NVMe-PCIe (I/O queues)

Primary purpose for I/O queues is to transfer data (read/write)



NVMe-PCIe (MMIO)

Data transfer mechanism for Admin and I/O command data through memory MMIO



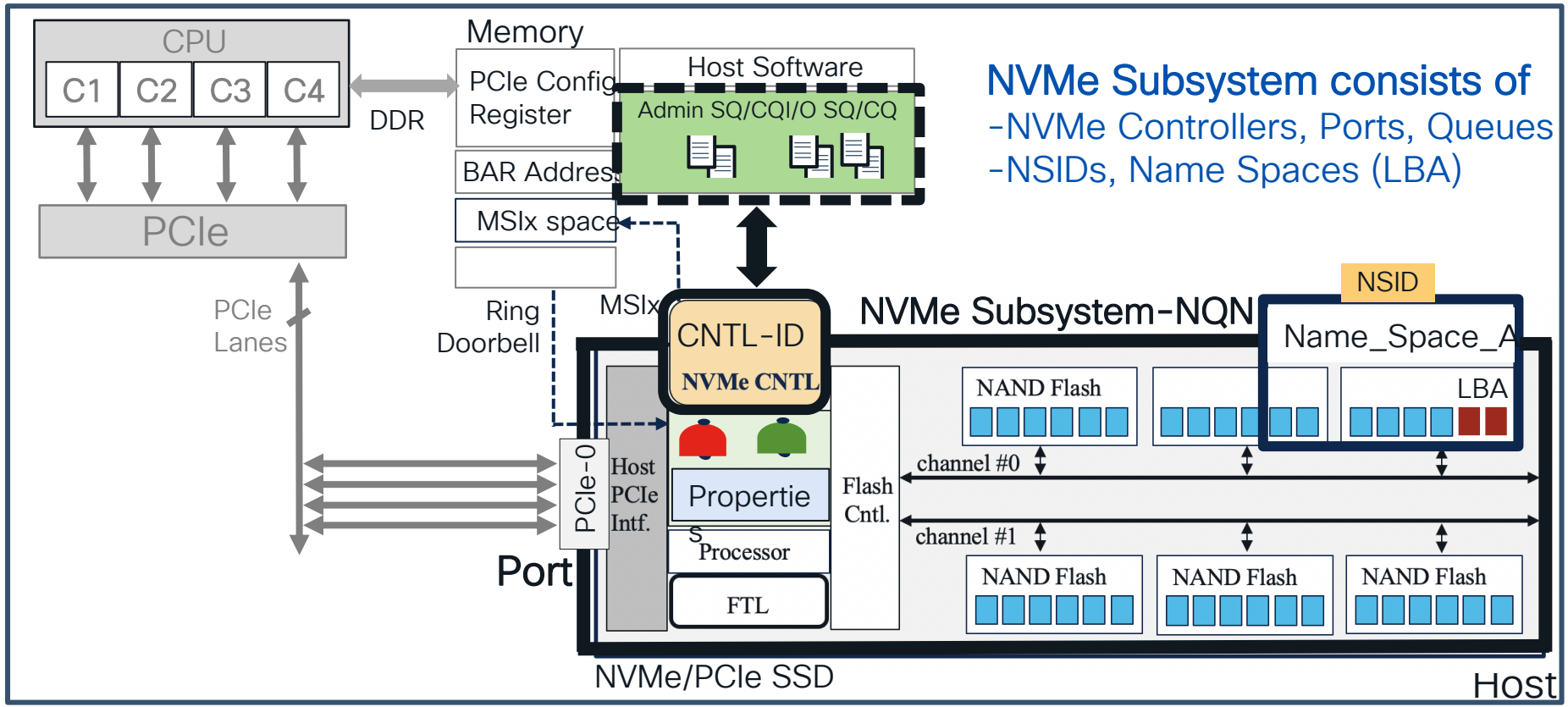
NVMe Admin & I/O queues are located inside Host memory

Host rings doorbell, controller fetches the SQE entry from the memory, processes it, updates the CQE entry in the host memory and generates the MSIx interrupt to host.

SQE Submission Queue Entry
CQE Completion Queue Entry

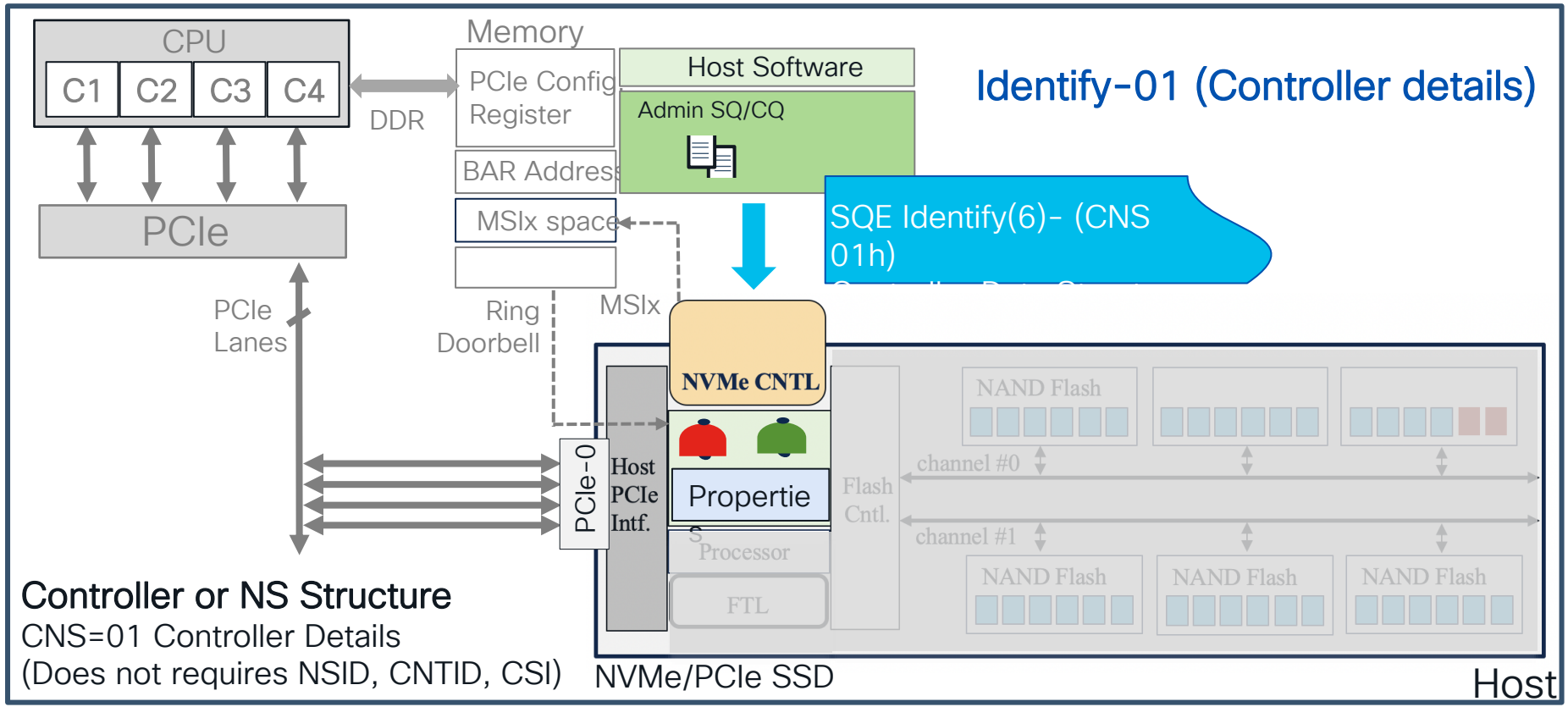
NVMe-PCIe (NVMe Subsystem)

What is NVMe Subsystem?

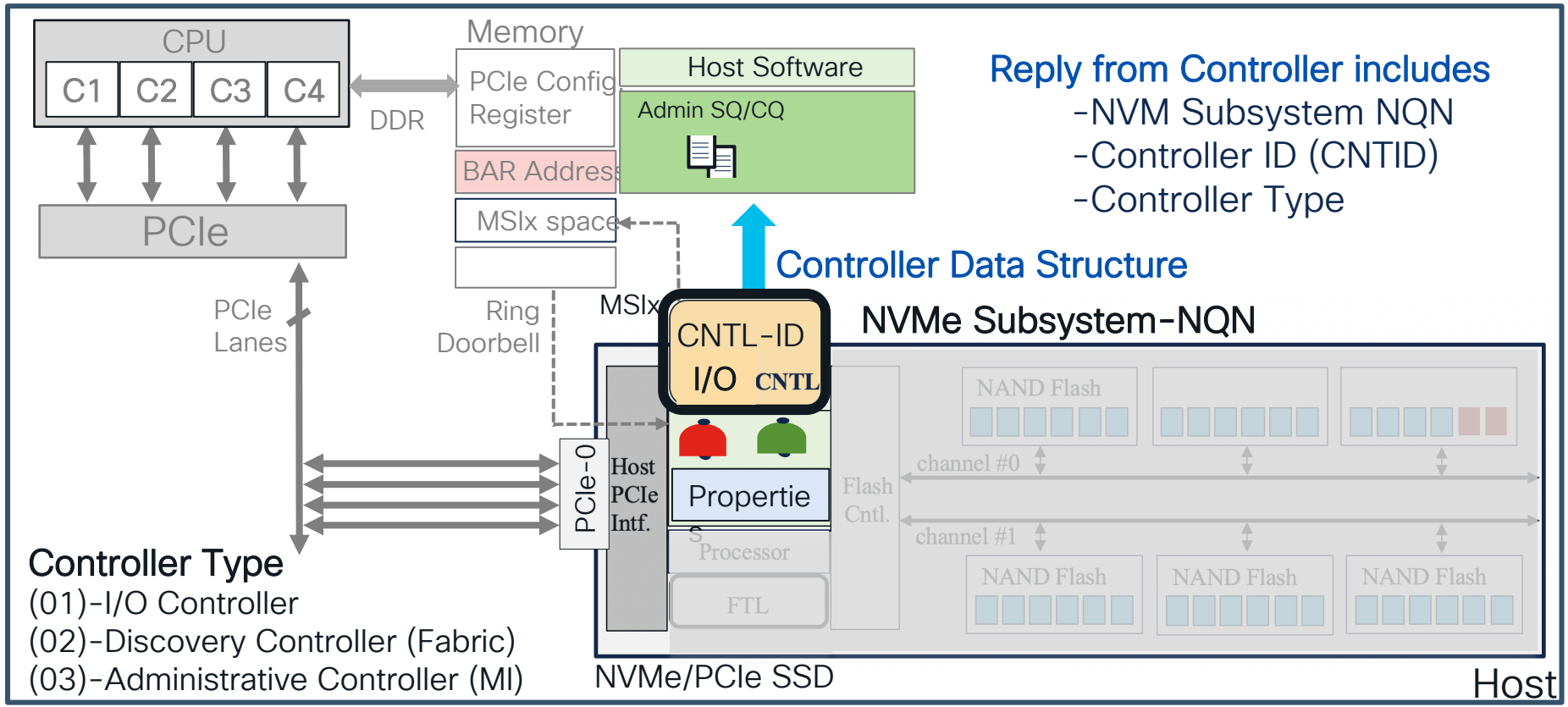


NVMe Subsystem consists of
- -NVMe Controllers, Ports, Queues
- -NSIDs, Name Spaces (LBA)

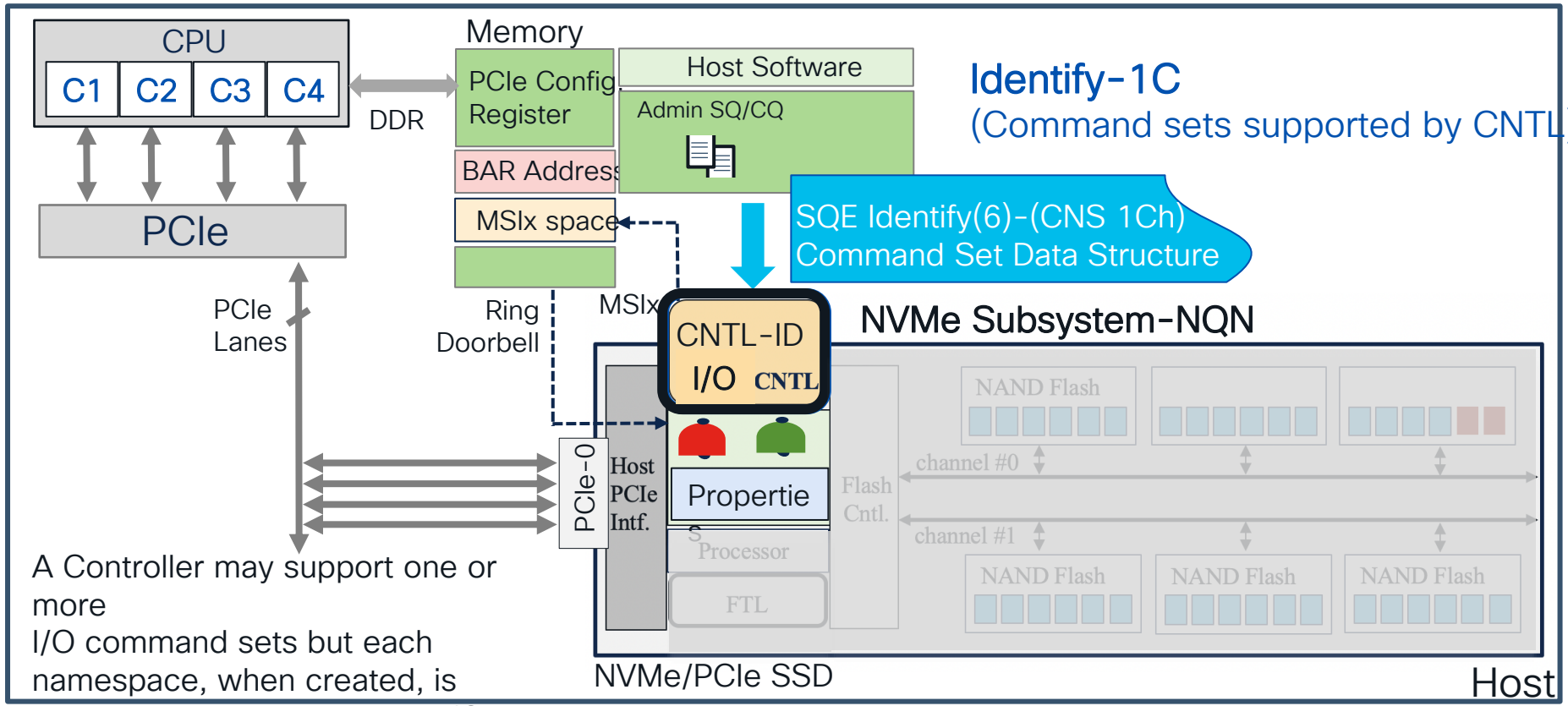
NVMe-PCIe (Identify-01) Host issues series of "Identify" commands to get NVMe Subsystem details



NVMe-PCIe (Identify Reply-01)

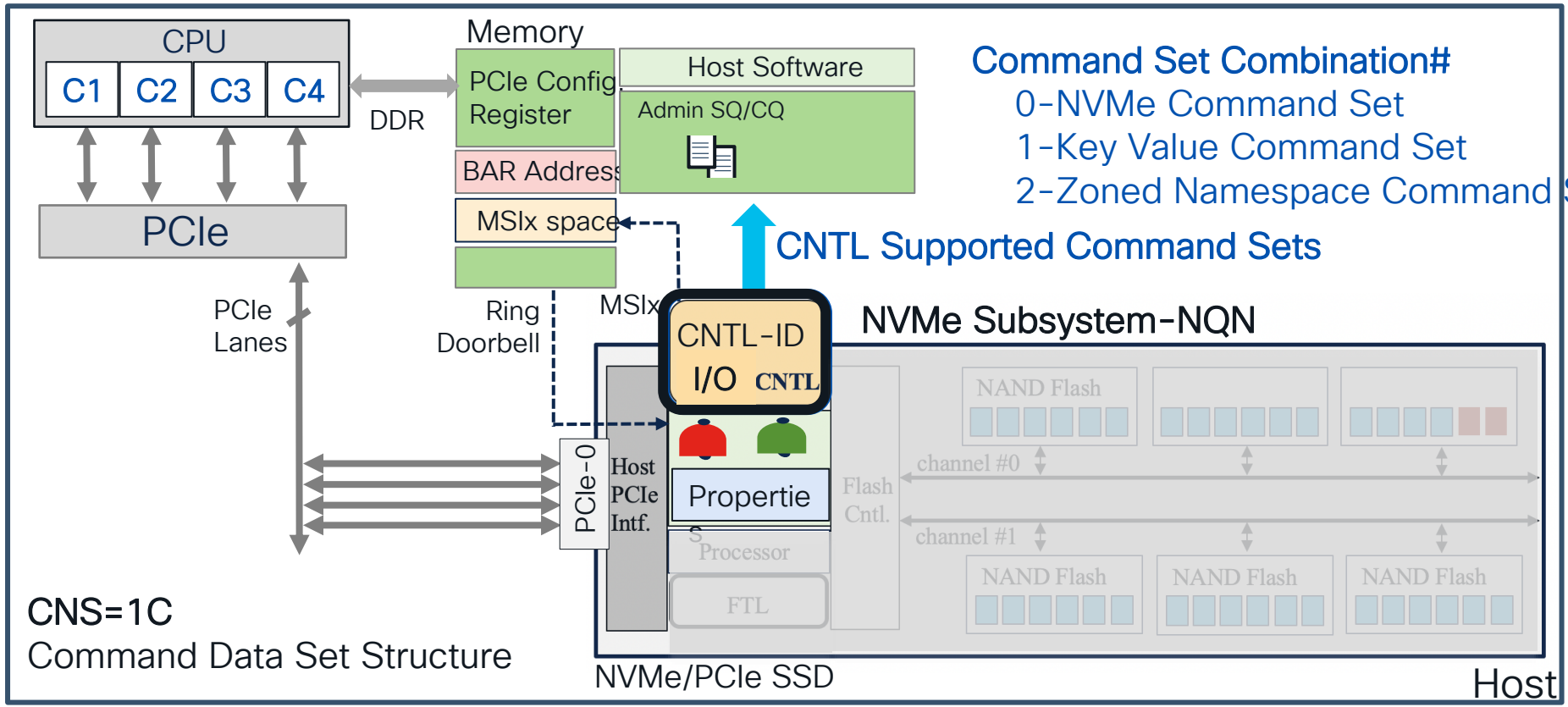


NVMe-PCIe (Identify-1C)

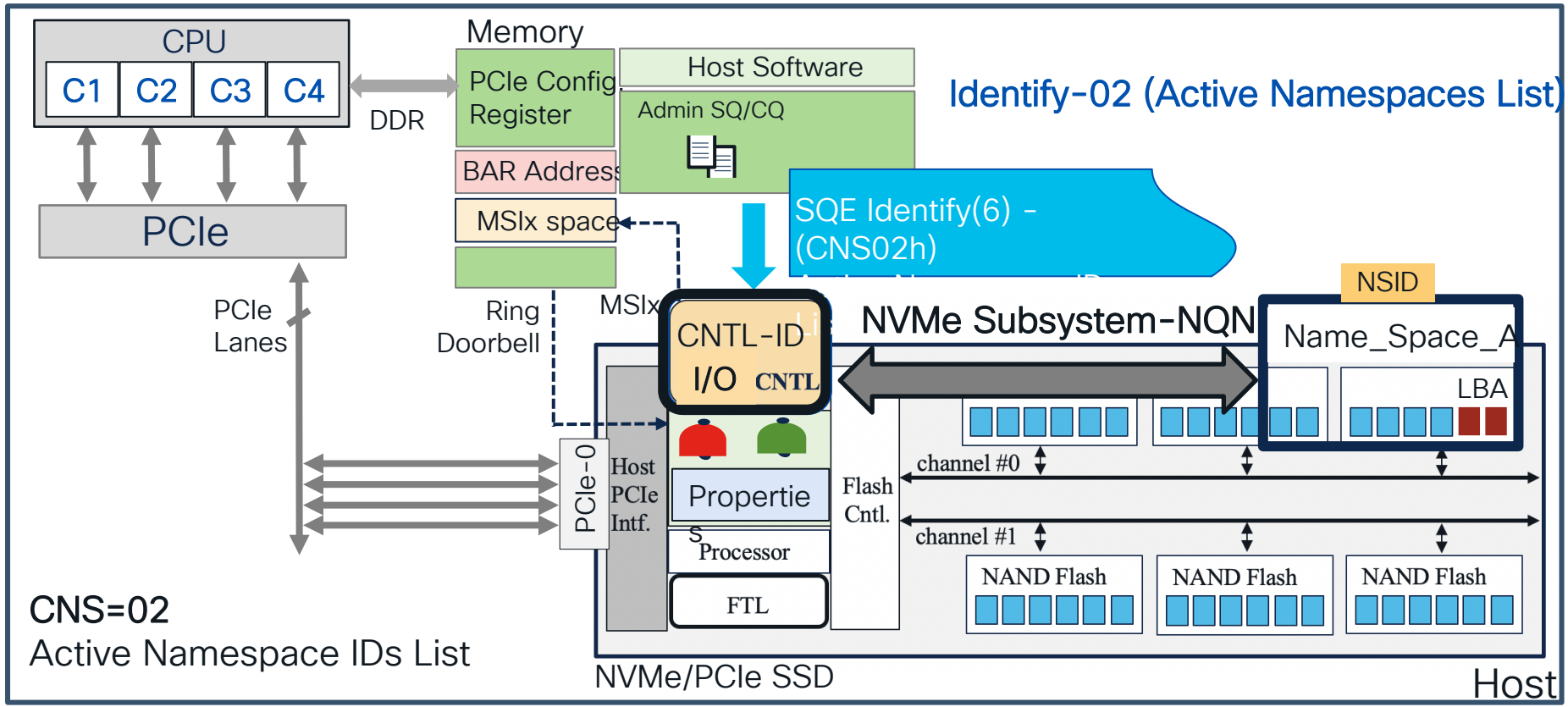


A Controller may support one or more
I/O command sets but each namespace, when created, is associated with exactly one I/O command set.

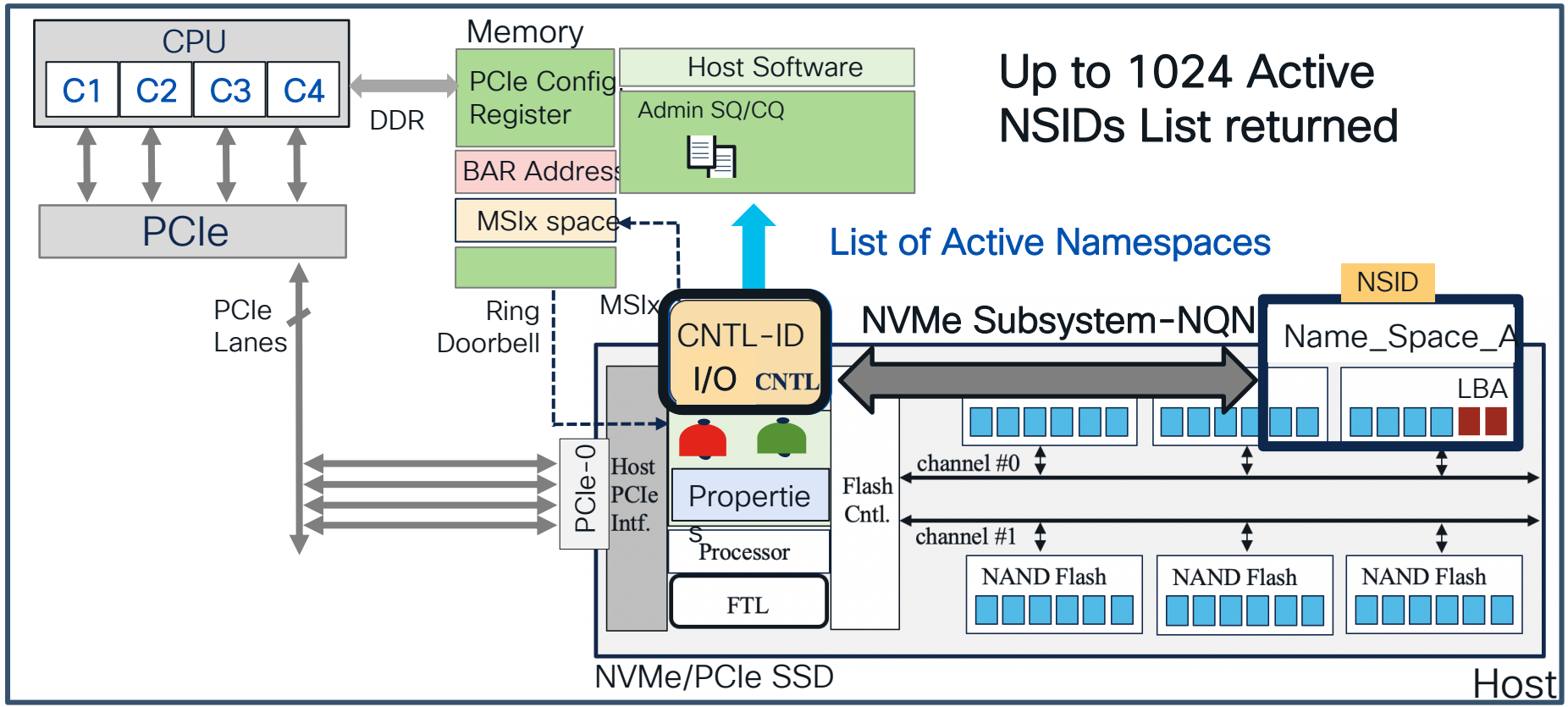
NVMe-PCIe (Identify-1C Reply)



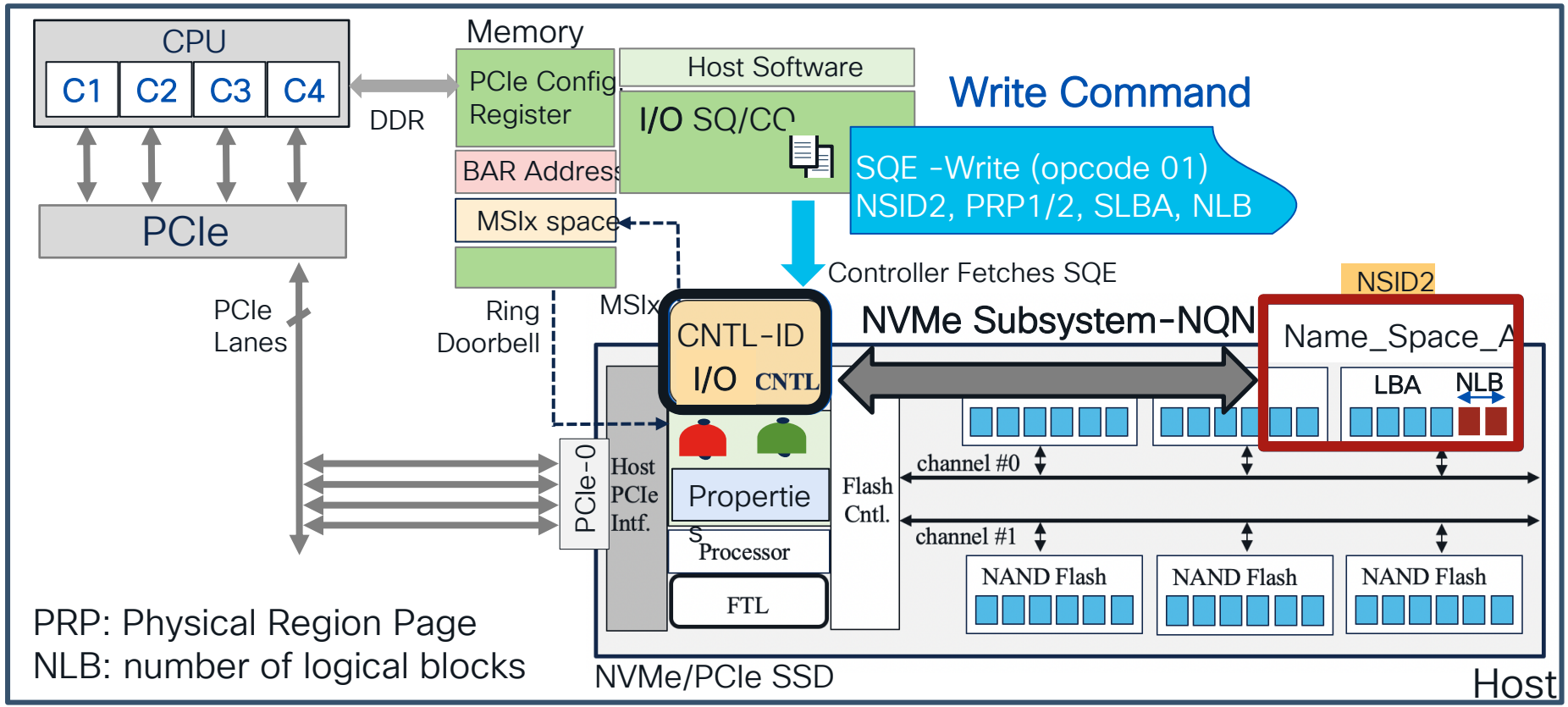
NVMe-PCIe (Identify-07)



NVMe-PCIe (Identify-07 Reply)

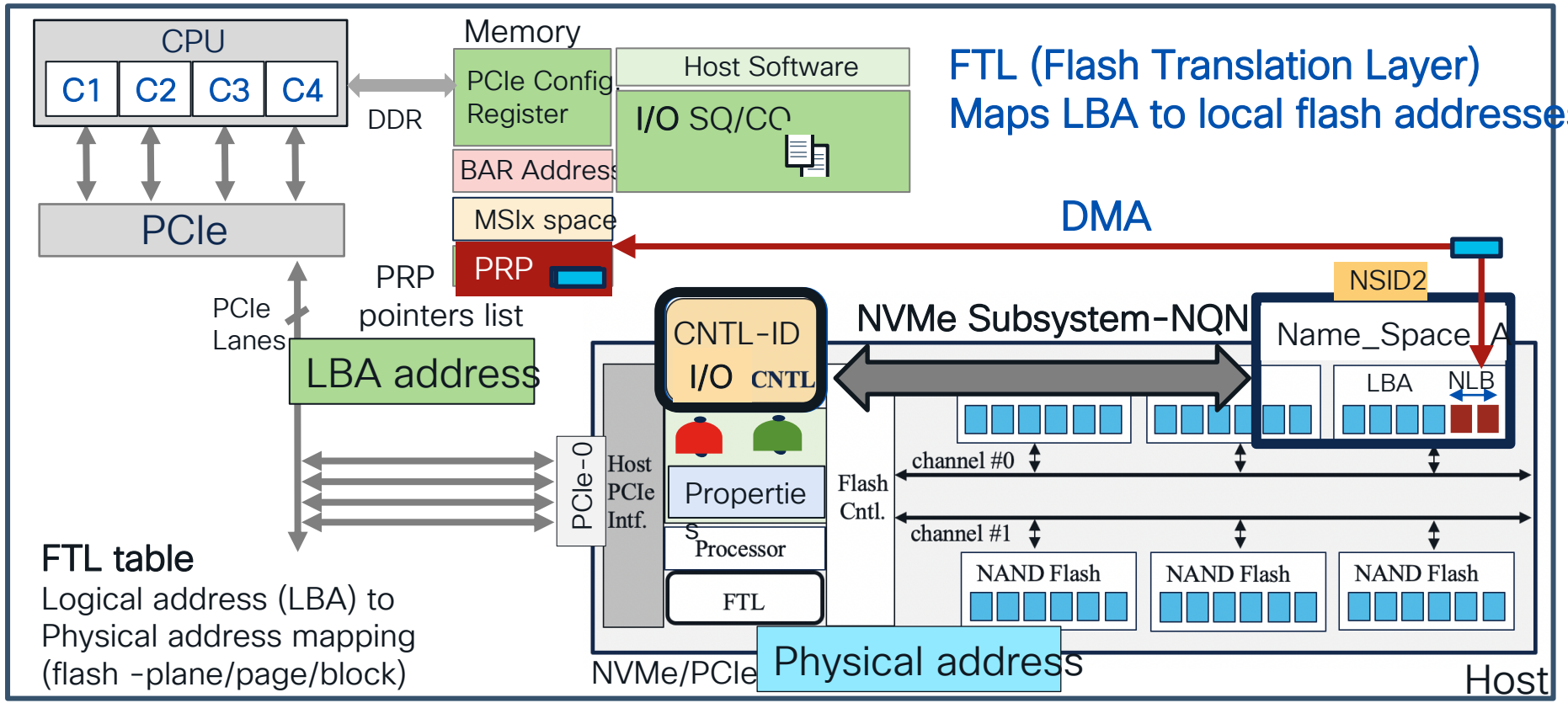


NVMe-PCIe (SQE-Write)

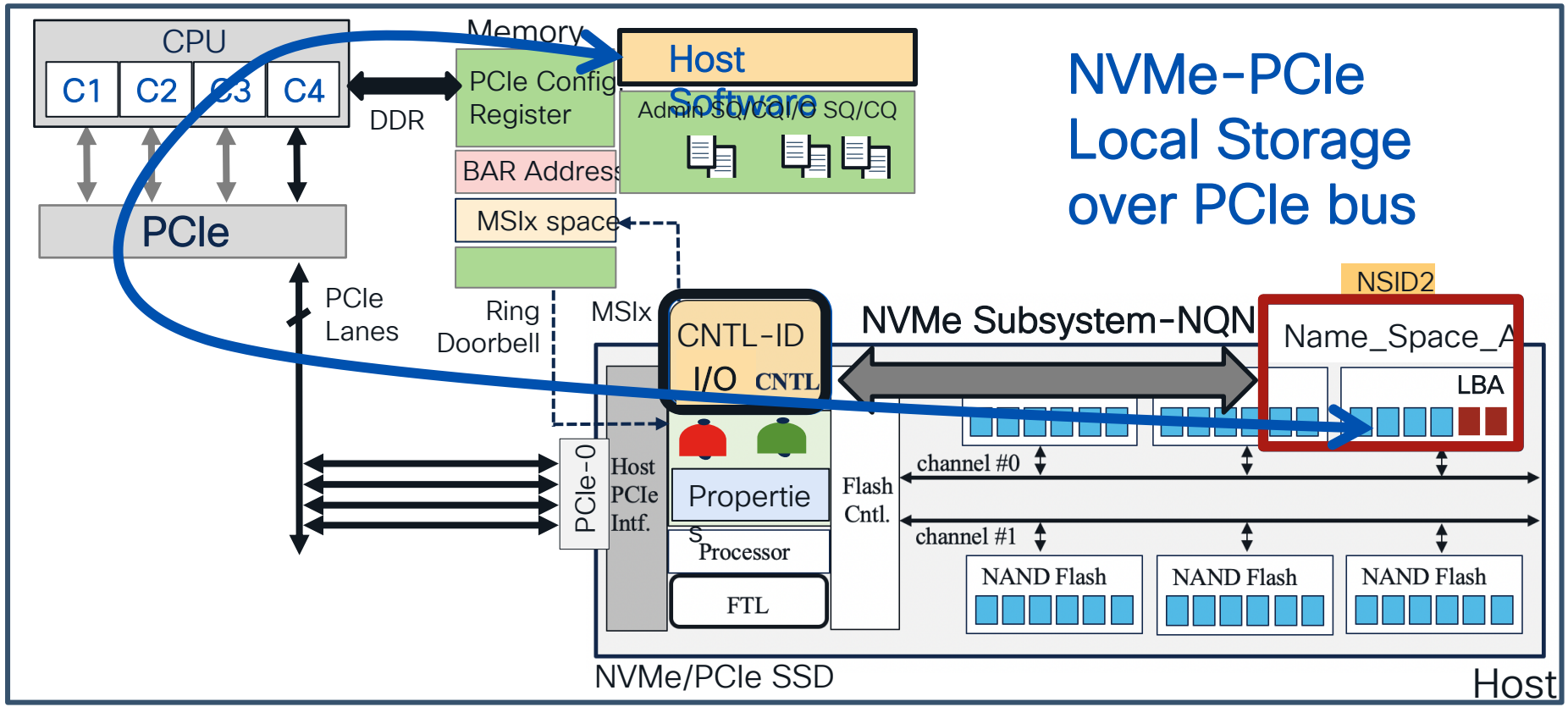


NVMe-PCIe (DMA Data)

Controller Reads Data from PRP buffers and writes it to the Flash



NVMe-PCIe



Best Practices (Do's & Don'ts)

- SSD comes in many flavors, check the type (SATA, SAS, NVMe, ZNS, KV-SSD)
- Don't use the SSD to its full capacity, leave at-least 25% free
- PCIe gen 5 SSD are now available and provide the highest performance
- For enterprise applications always use dual port NVMe-SSD (8 lanes)

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

3-NVMe Transport Options (NVMe-FC)

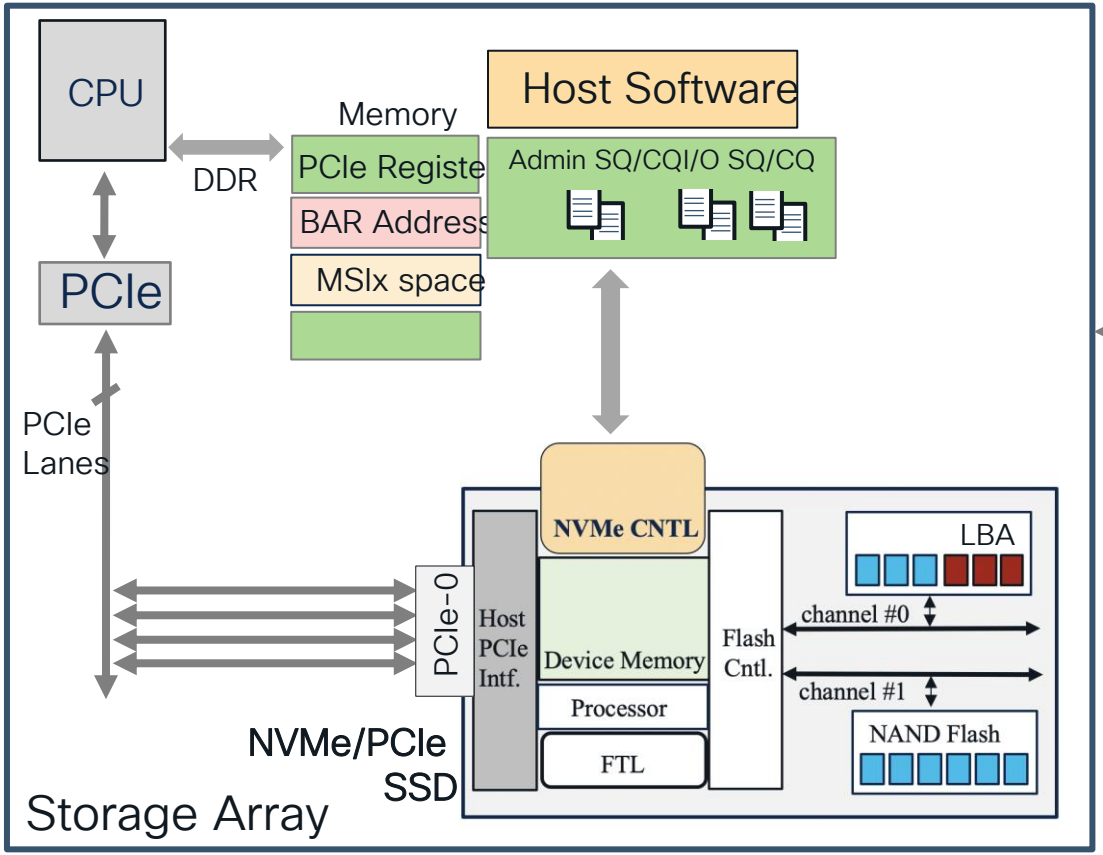
4-NVMe Datacenter Design

5-Additional Information

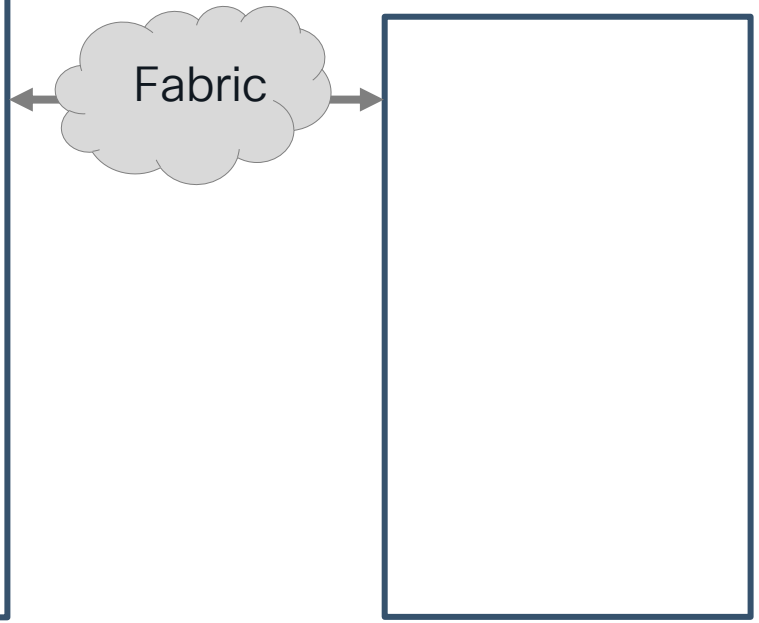
- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

NVMe

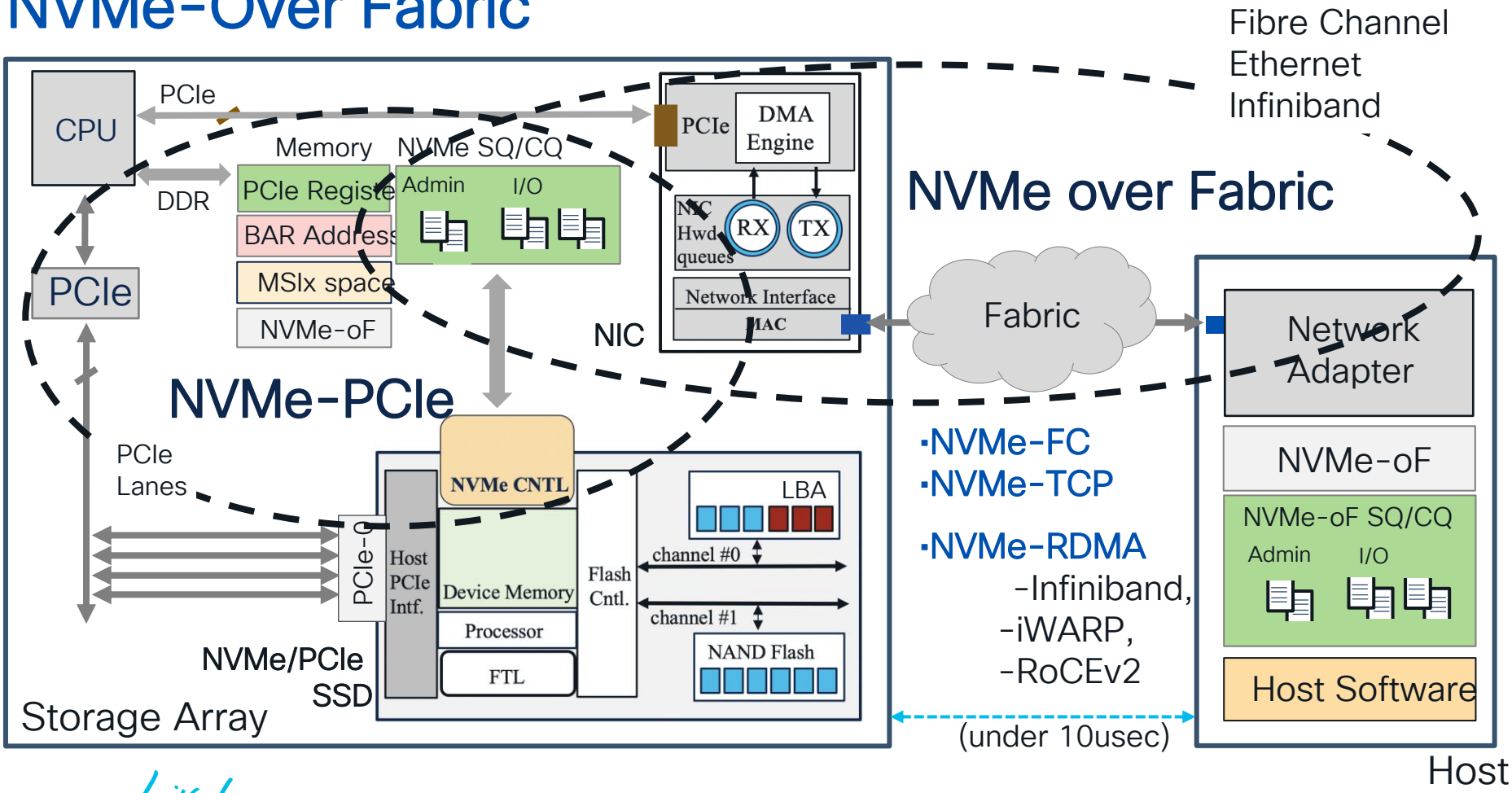
What if the Host is remote?



Who will create the NVMe queues?
What kind of network will be supported?
How the commands will be transported?
Who will ring the Doorbells or send MSI



NVMe-Over Fabric



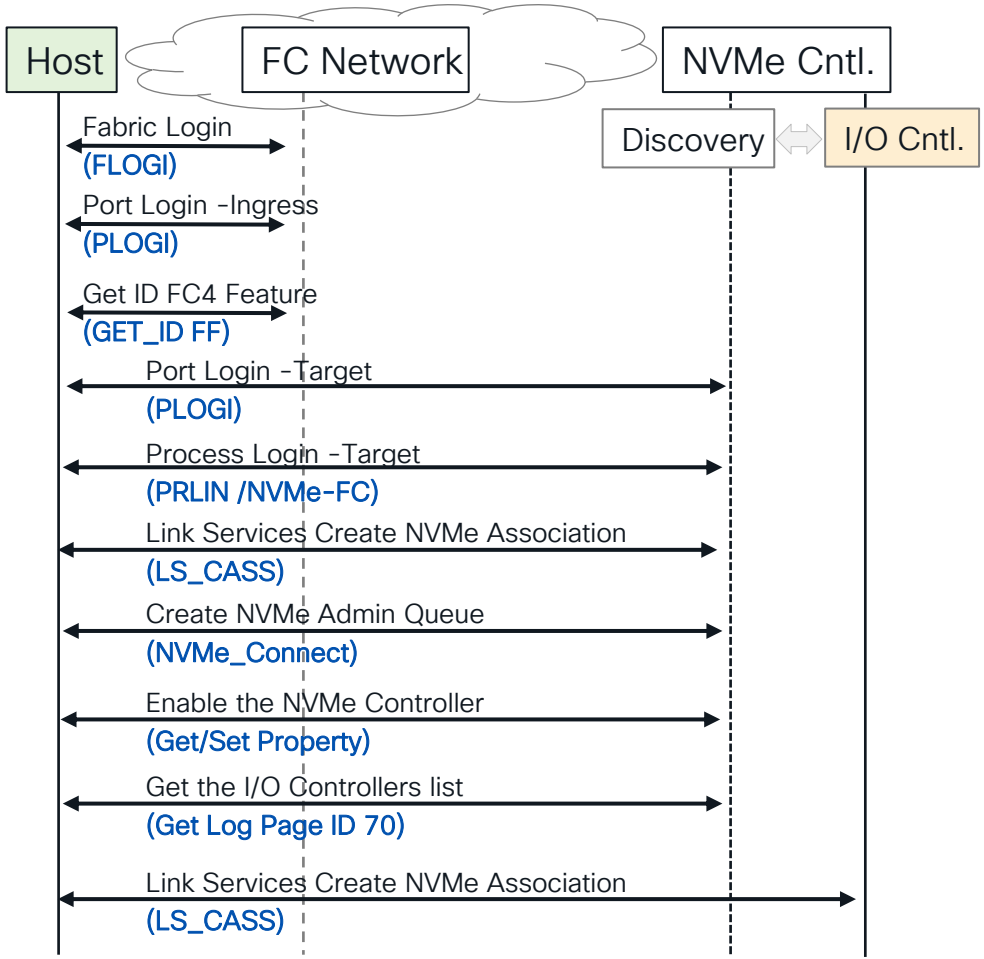
NVMe-FC Transport

I Connect to NVMe Cntl.

II Create NVMe Queues

III Discover Namespaces

IV Start NVMe I/O operation



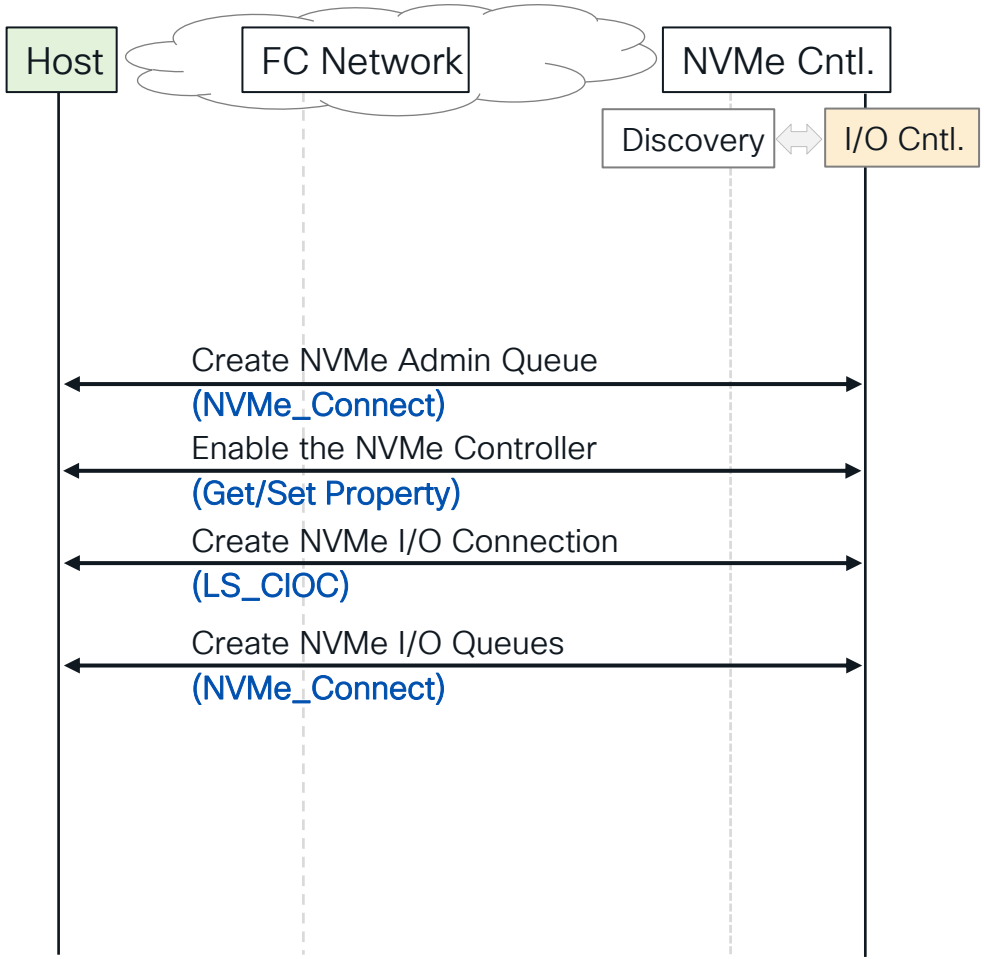
NVMe-FC Transport

I Connect to NVMe Cntl.

II Create NVMe Queues

III Discover Namespaces

IV Start NVMe I/O operation



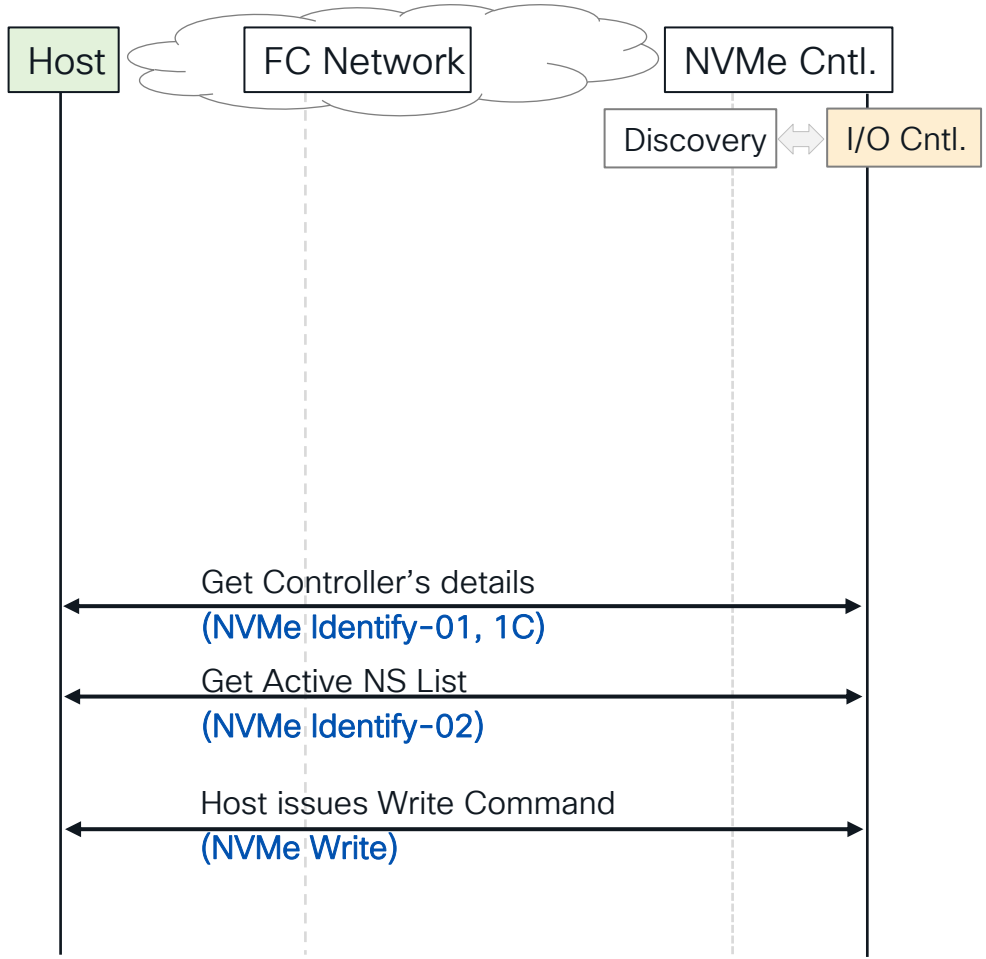
NVMe-FC Transport

I Connect to NVMe Cntl.

II Create NVMe Queues

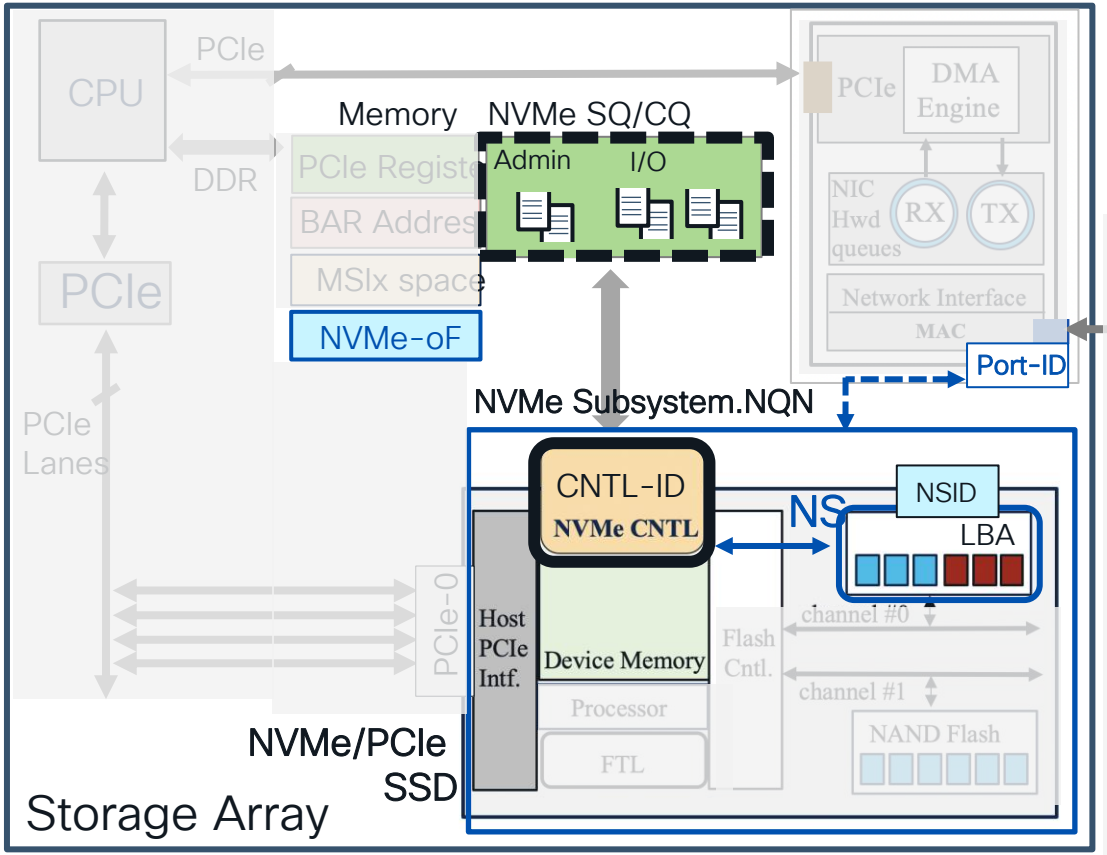
III Discover Namespaces

IV Start NVMe I/O operation



NVMe-oF (NVMe Subsystem)

skip



-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

Fabric

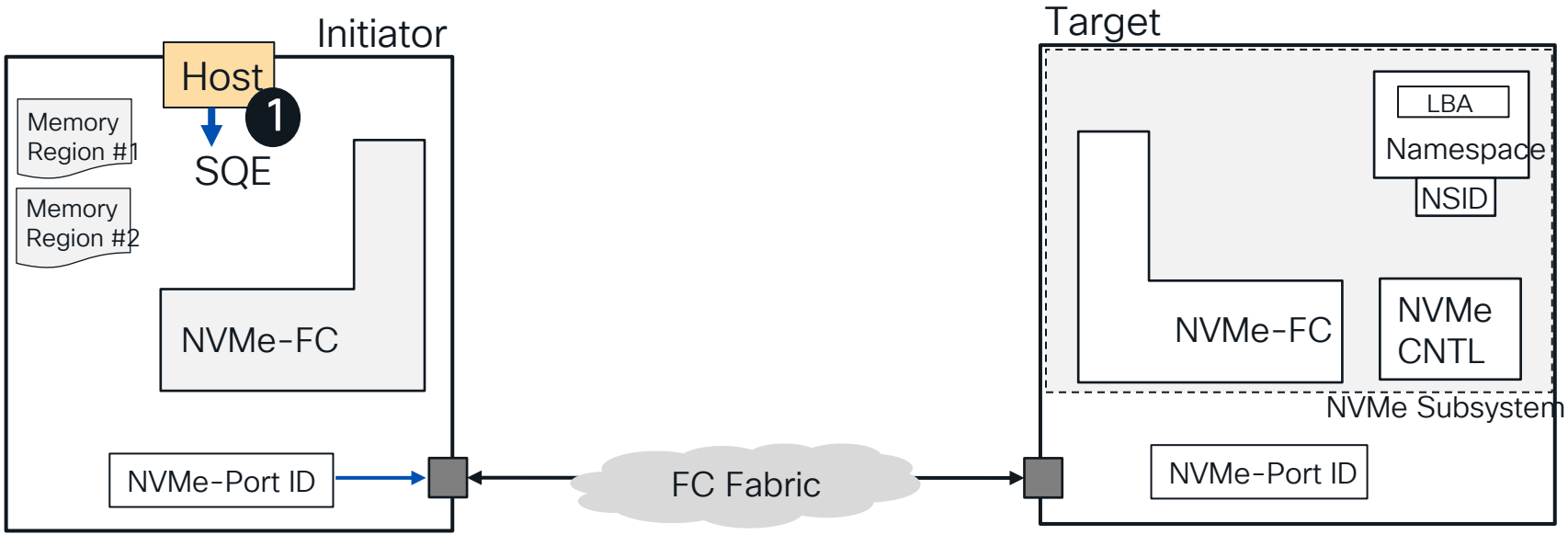
Host Bus Adapter

NVMe-oF

Host Software

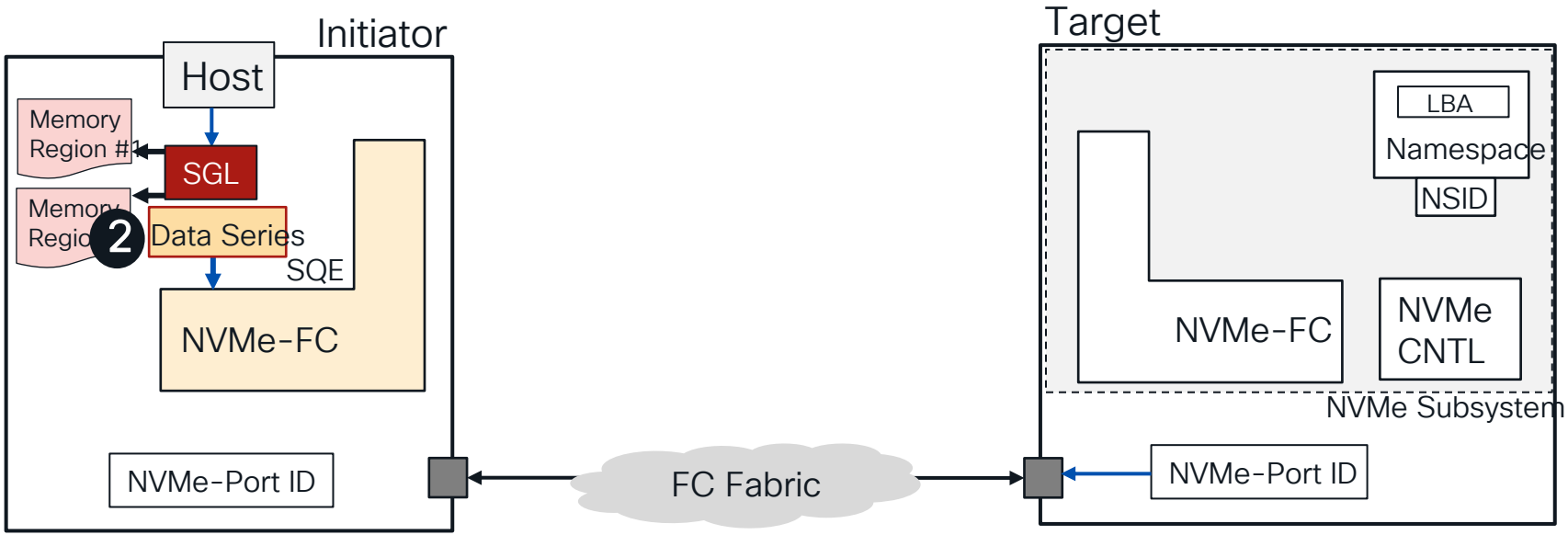
Host

NVMe-oF (FC Mapping Abstractions)



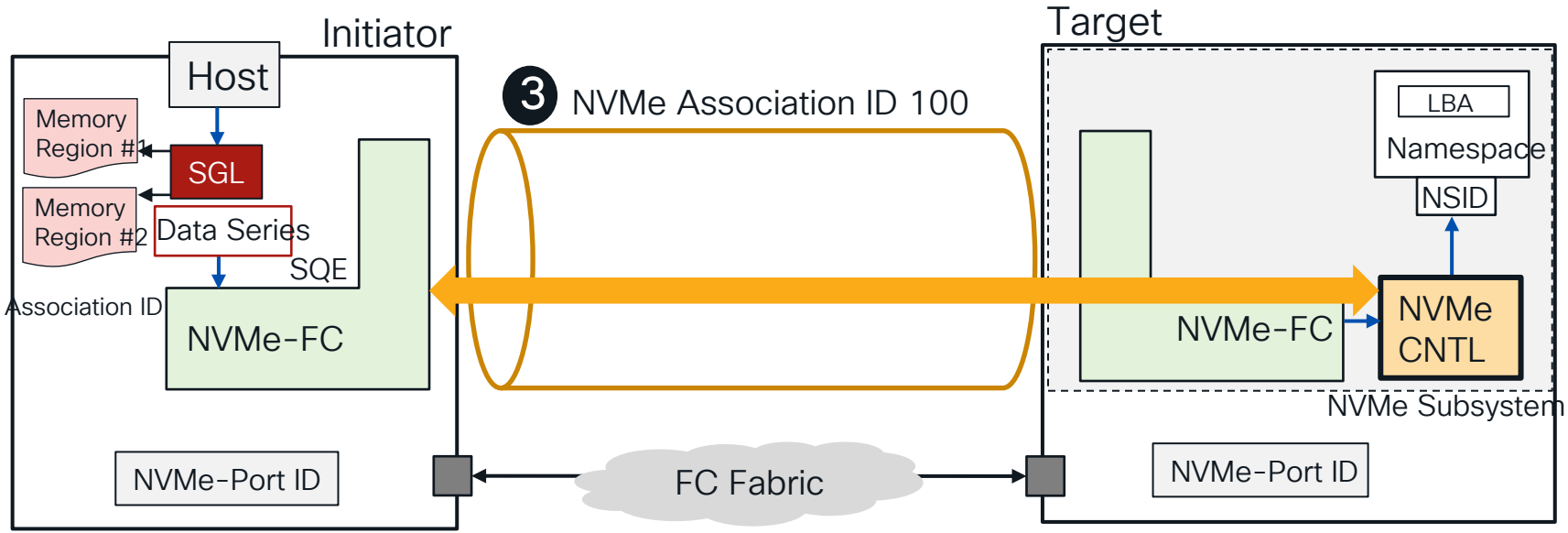
1 NVMe Host Submits a NVMe_Write command as SQE (Submission Queue Entry

NVMe-oF (FC Mapping Abstractions)



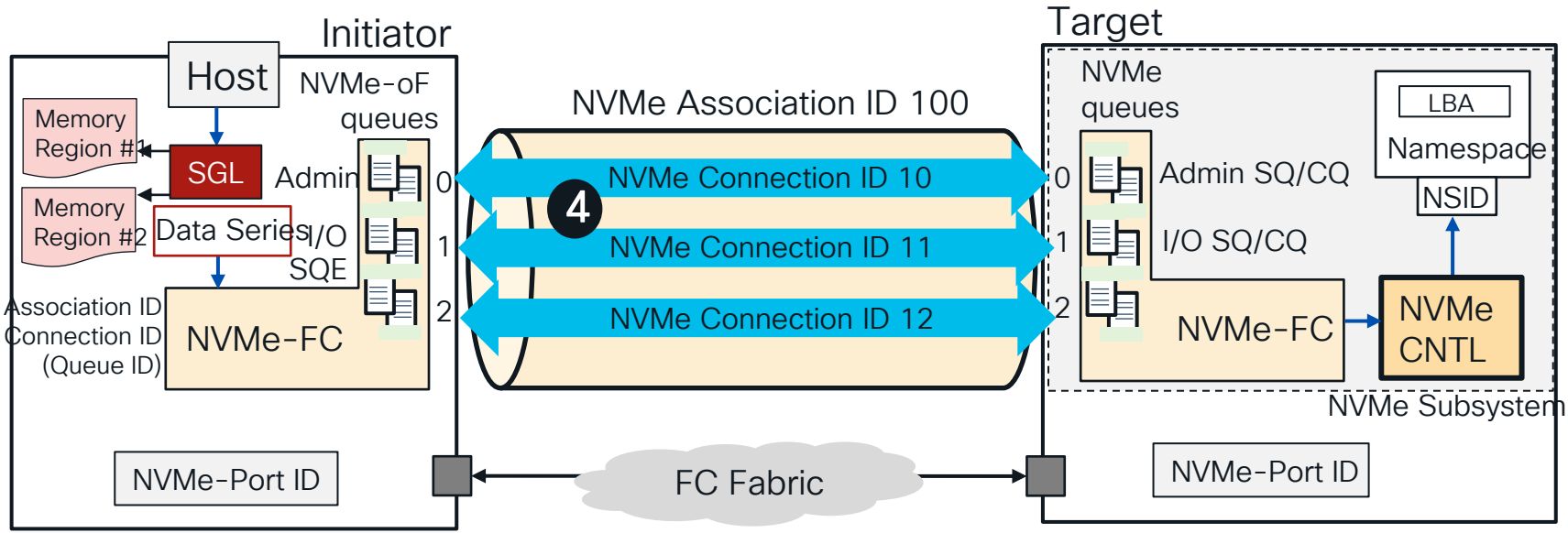
2 Data pointed by the Host SGL is placed in a Data Series and command is passed to NVMe-FC layer

NVMe-oF (FC Mapping Abstractions) -Association ID



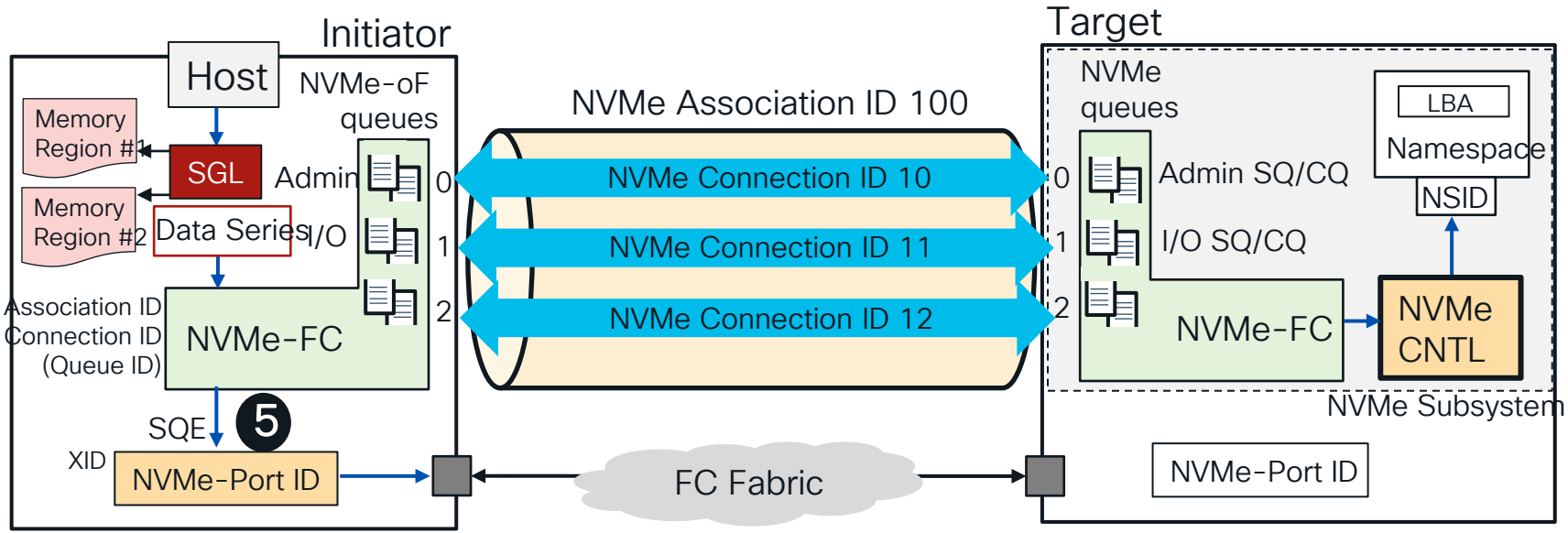
3 The Host NVMe-FC layer specifies the NVMe-FC association with the NVMe con

NVMe-oF (FC Mapping Abstractions) -Connection IDs



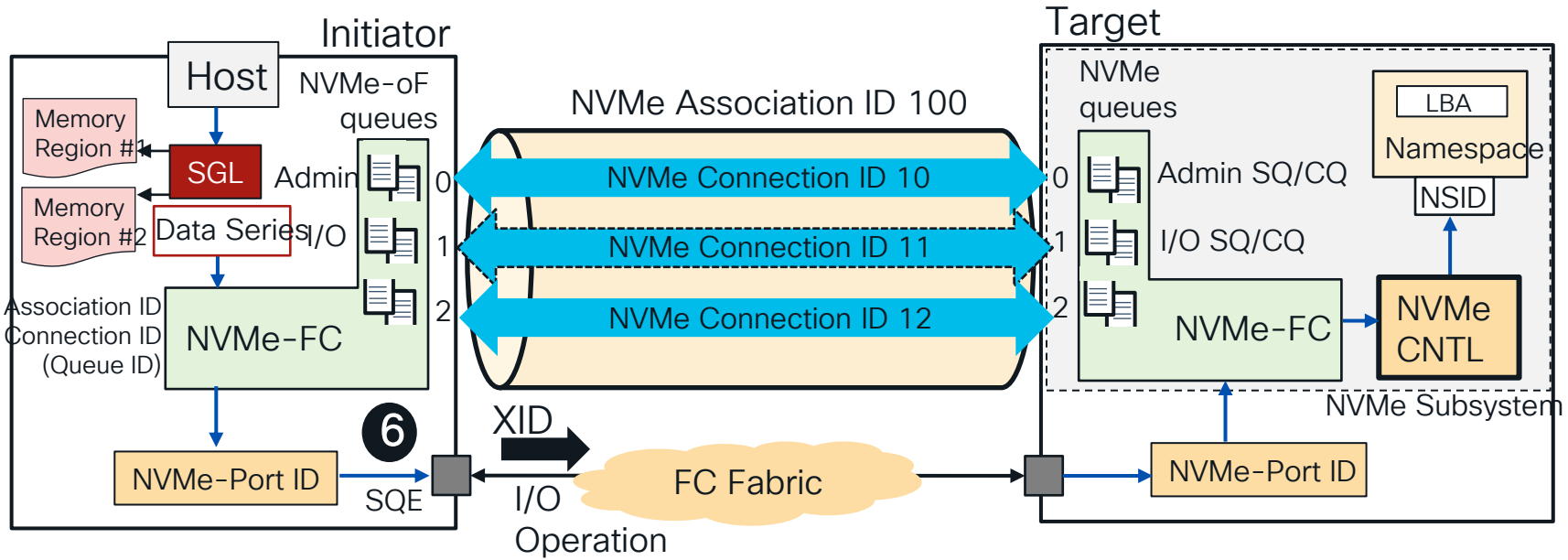
4 The Host NVMe-FC layer maintains a mapping of Host queues (NVMe-oF) to the NVMe controller's NVMe queues (SQ/CQ) via connection IDs.

NVMe-oF (FC Mapping Abstractions) -Exchange IDs



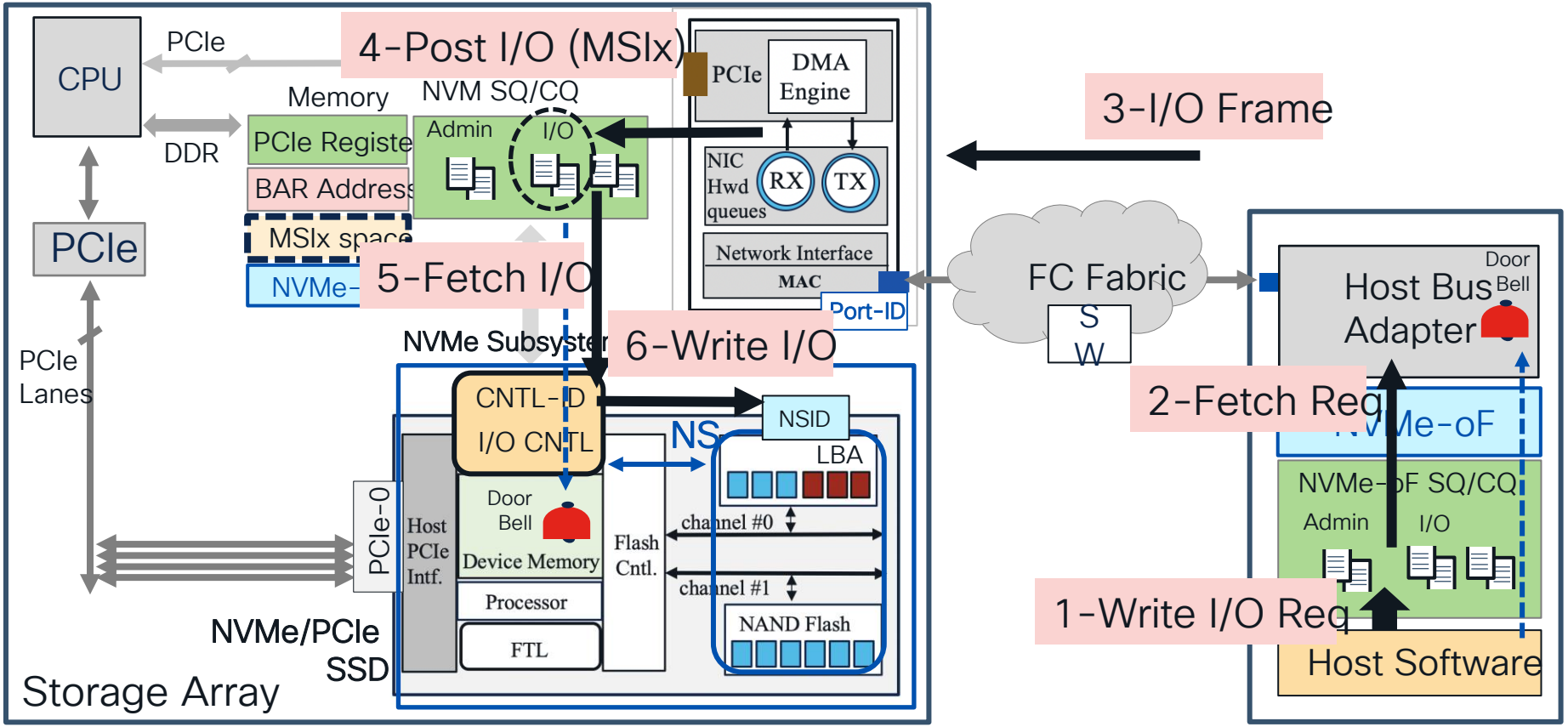
5 Upon receiving the SQE command NVMe_Port allocates XID for the NVMe-FC operation and associates the NVMe command in the SQE to the Exchange. All IUs for the NVMe-FC I/O operation are transmitted as part of this Exchange.

NVMe-FC (Association ID, Connection ID, Exchange ID, Queue ID)



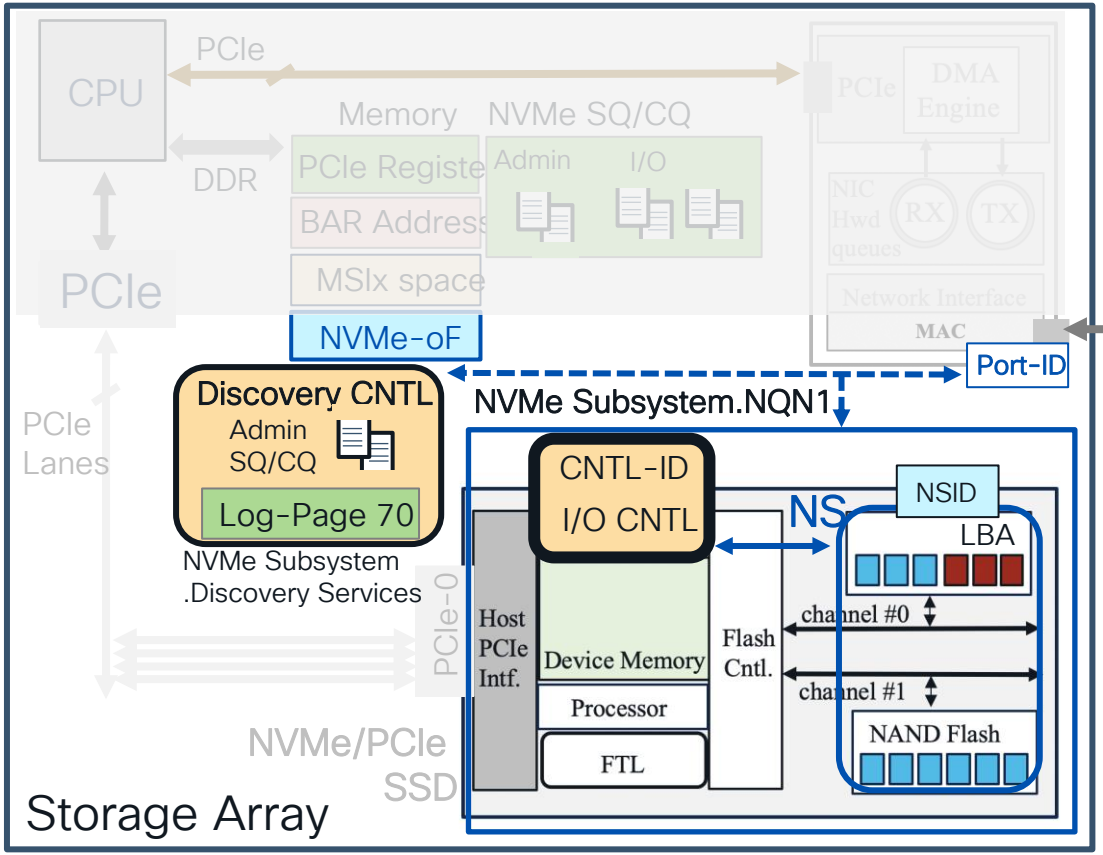
6 The initiator NVMe_Port transmits the NVMe_CMND IU payload to start the NVMe-FC I/O operation.

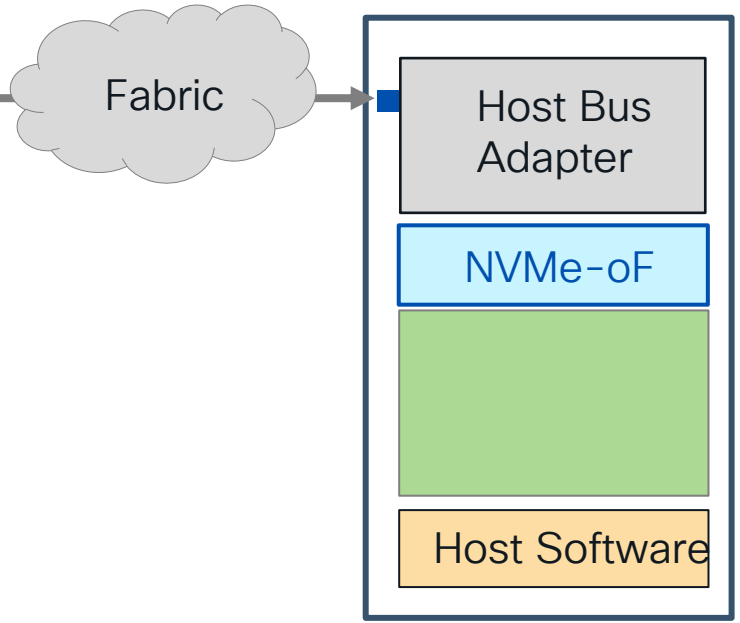
NVMe-oF (HBA/MSIx Interrupts)



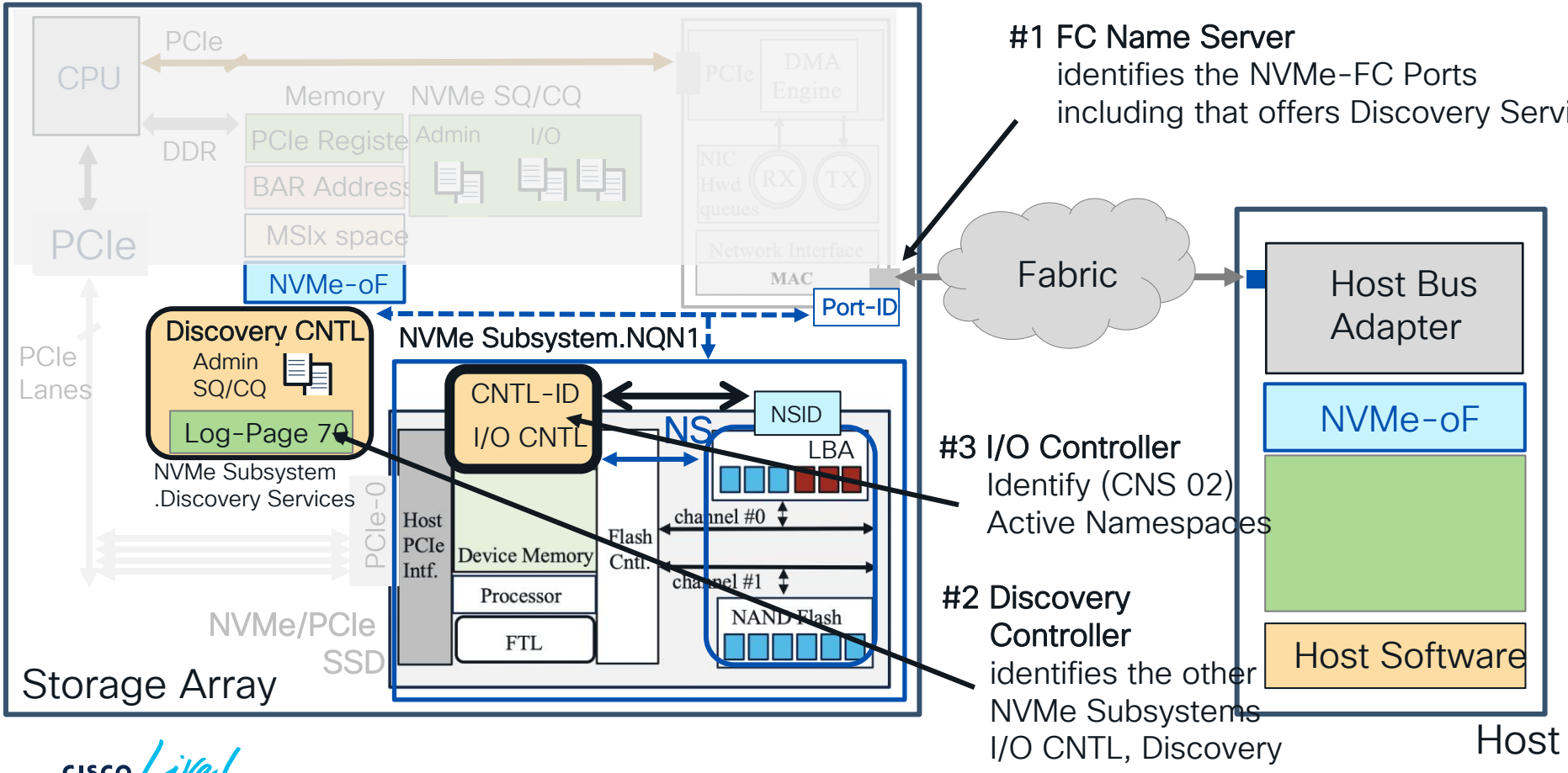
NVMe-oF (Discovery Services Subsystem)

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





NVMe-oF (Discovery Services Subsystem)

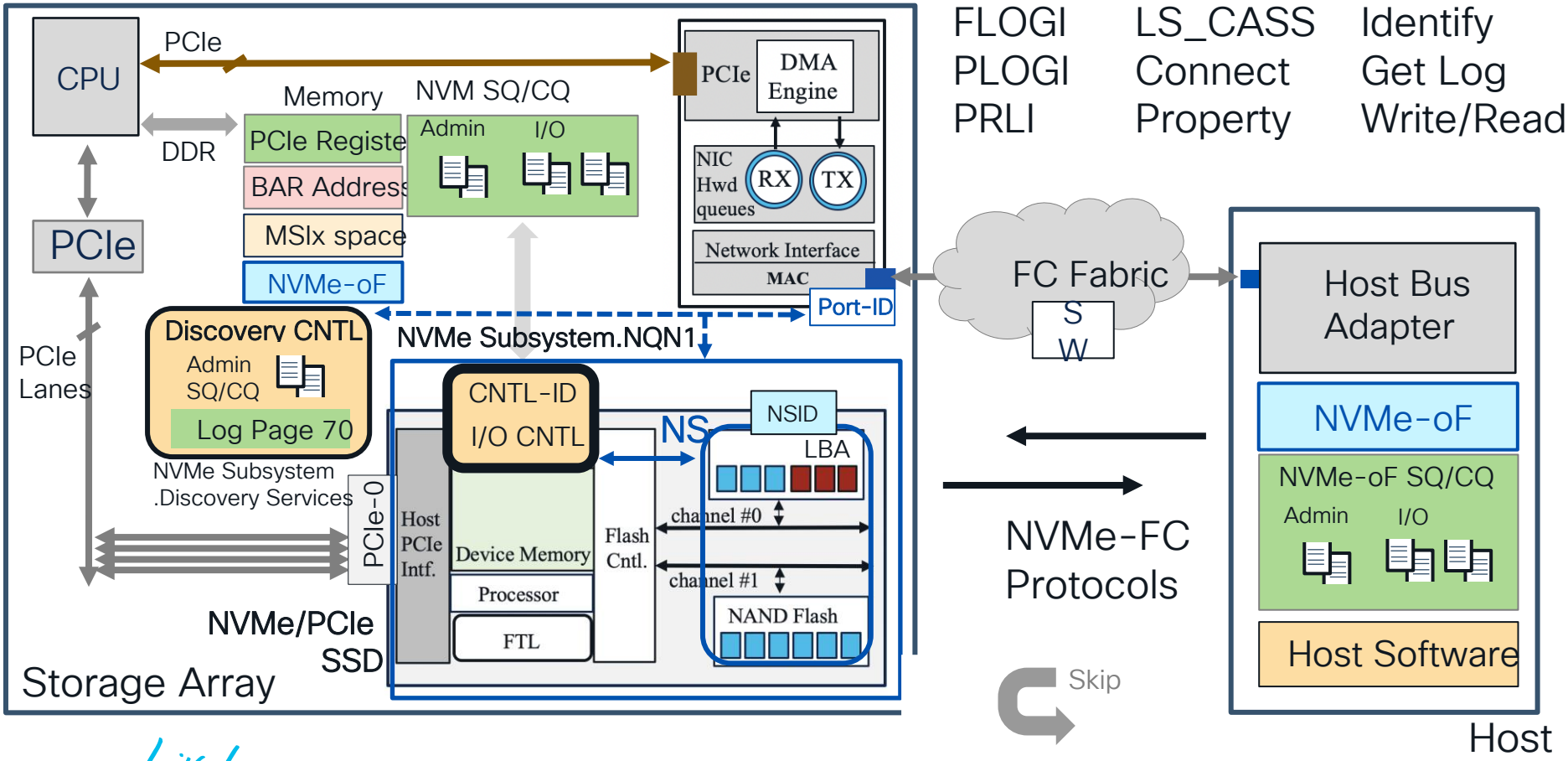


#1 FC Name Server
identifies the NVMe-FC Ports including that offers Discovery Servic

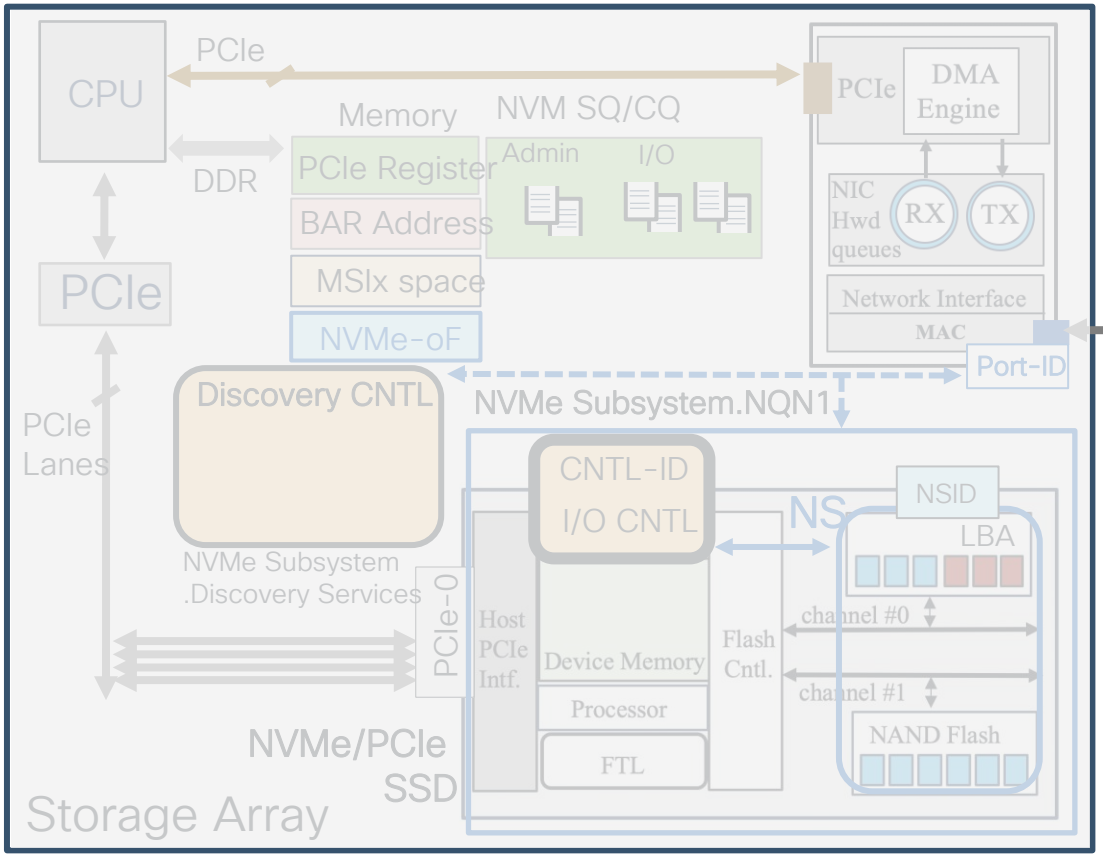
#3 I/O Controller
Identify (CNS 02)
Active Namespaces

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

NVMe-FC Protocol Flows



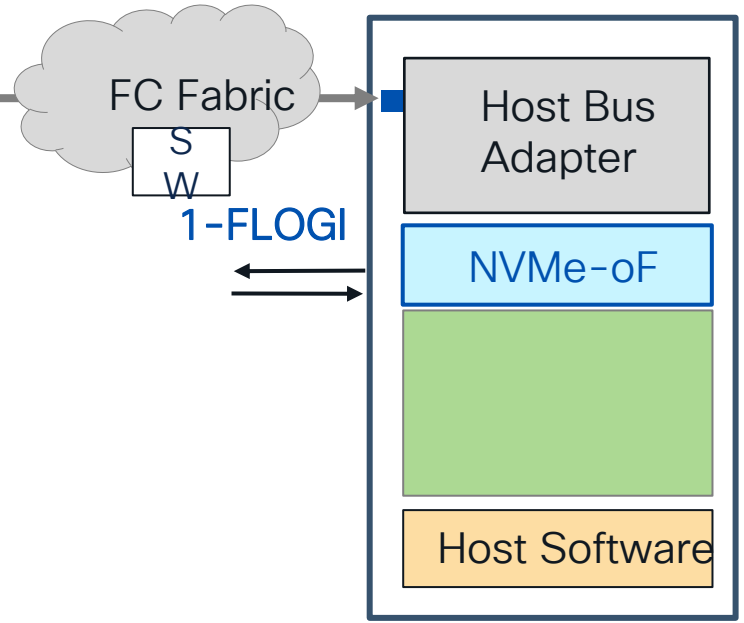
NVMe-FC Protocol Flows (FLOGI)



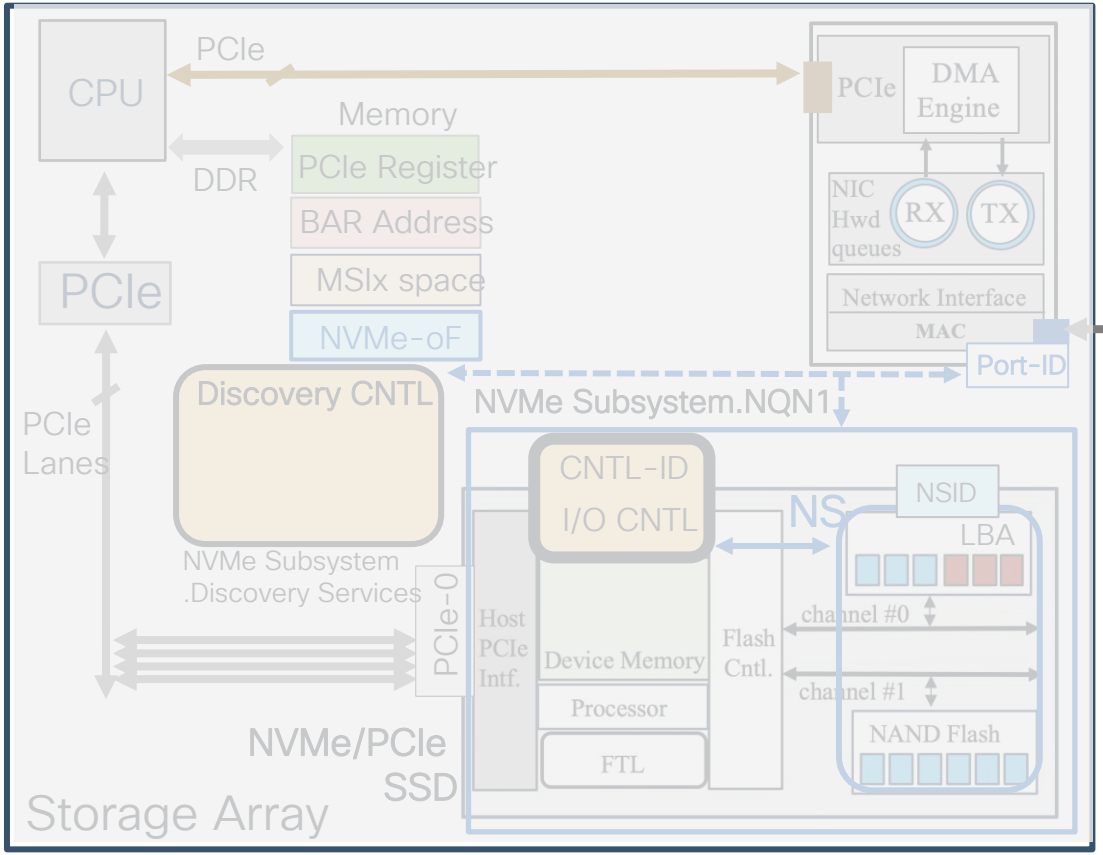
Flogi (Fabric Login)

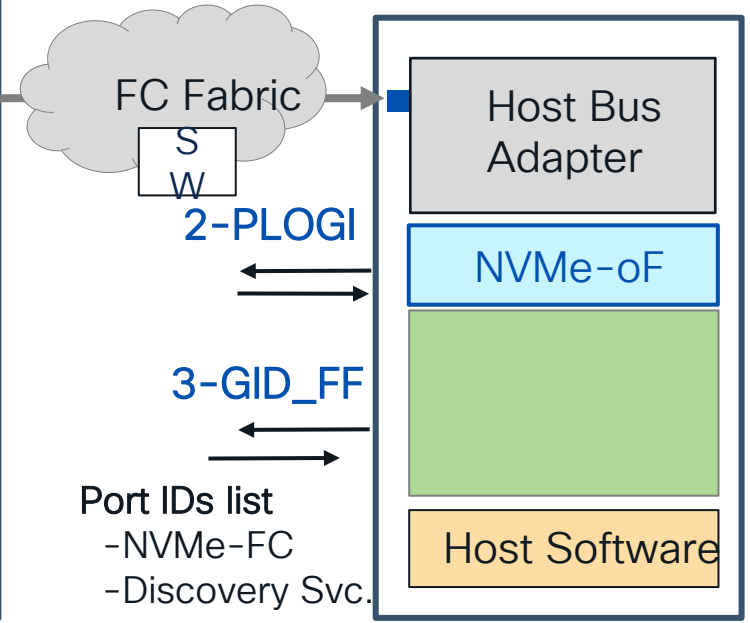
FCID is assigned

B2B are initialized



NVMe-FC Protocol Flows (PLOGI)



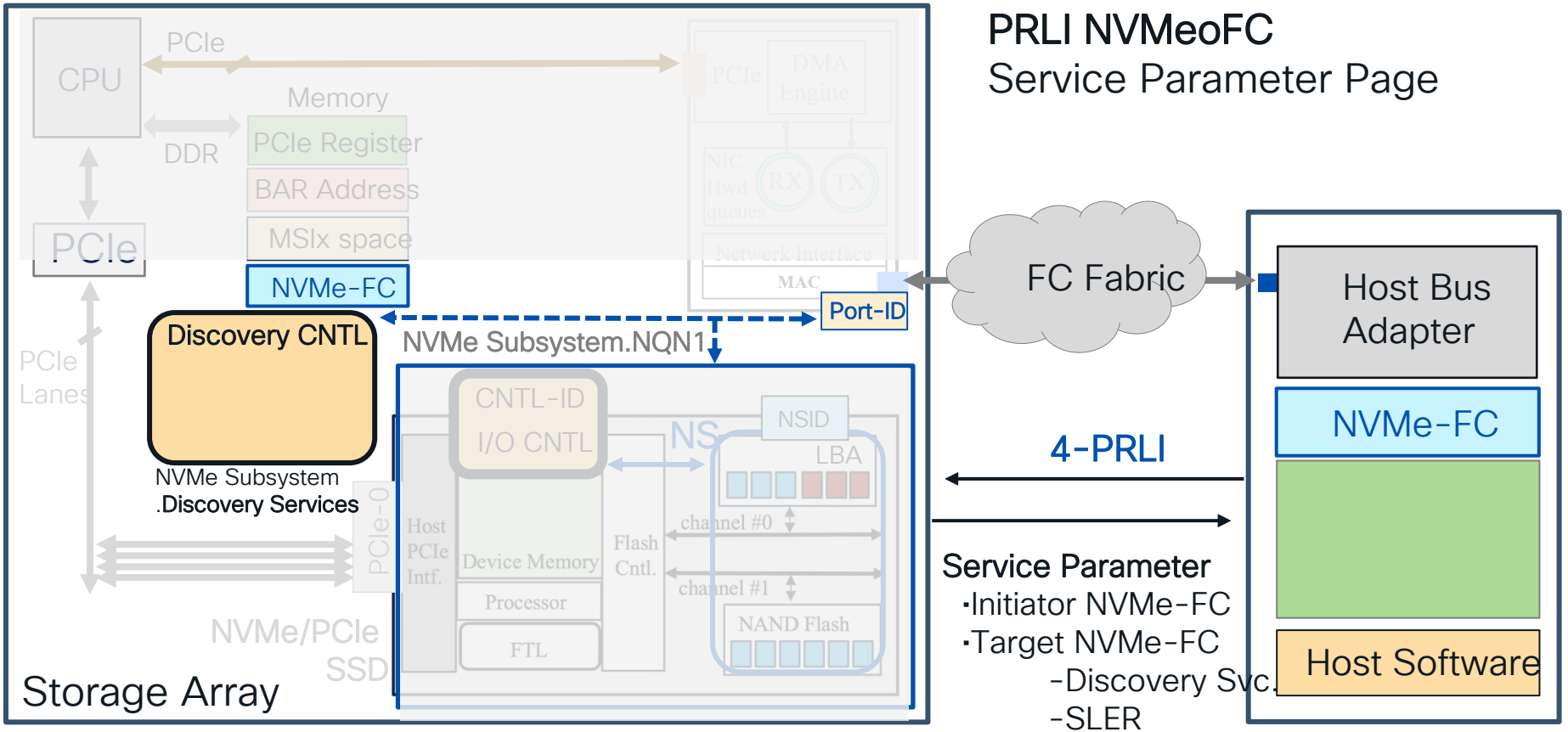


Port Logins
- Name Server Login -Registration
- Fabric Controller - SCN

Get ID_FF (FC4 Features Support)
- -Type 28/NVMeoFC
- -Feature 04/Discovery Services

Port IDs list
- -NVMe-FC
- -Discovery Svc.

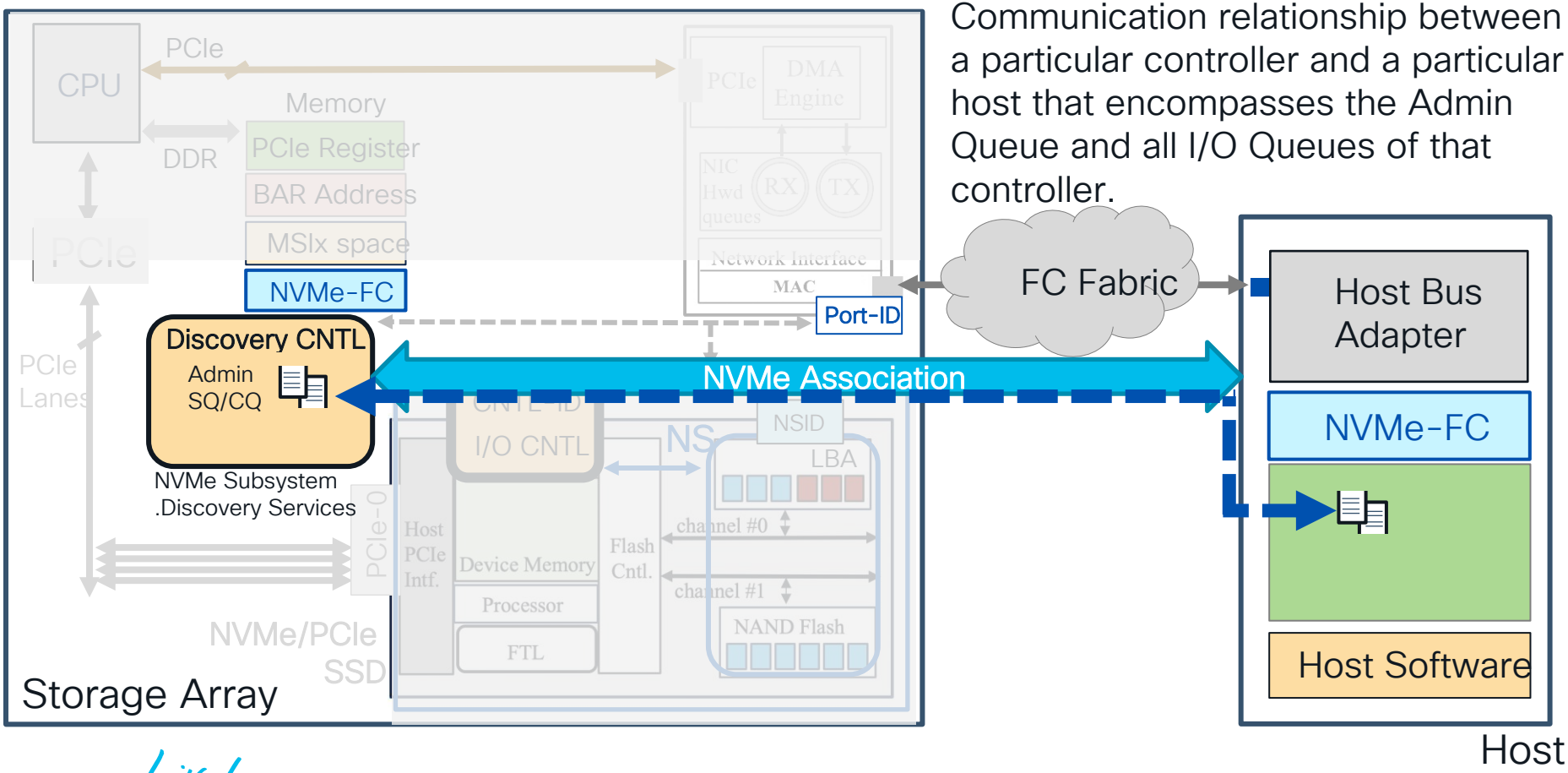
NVMe-FC Protocol Flows (PRLI)



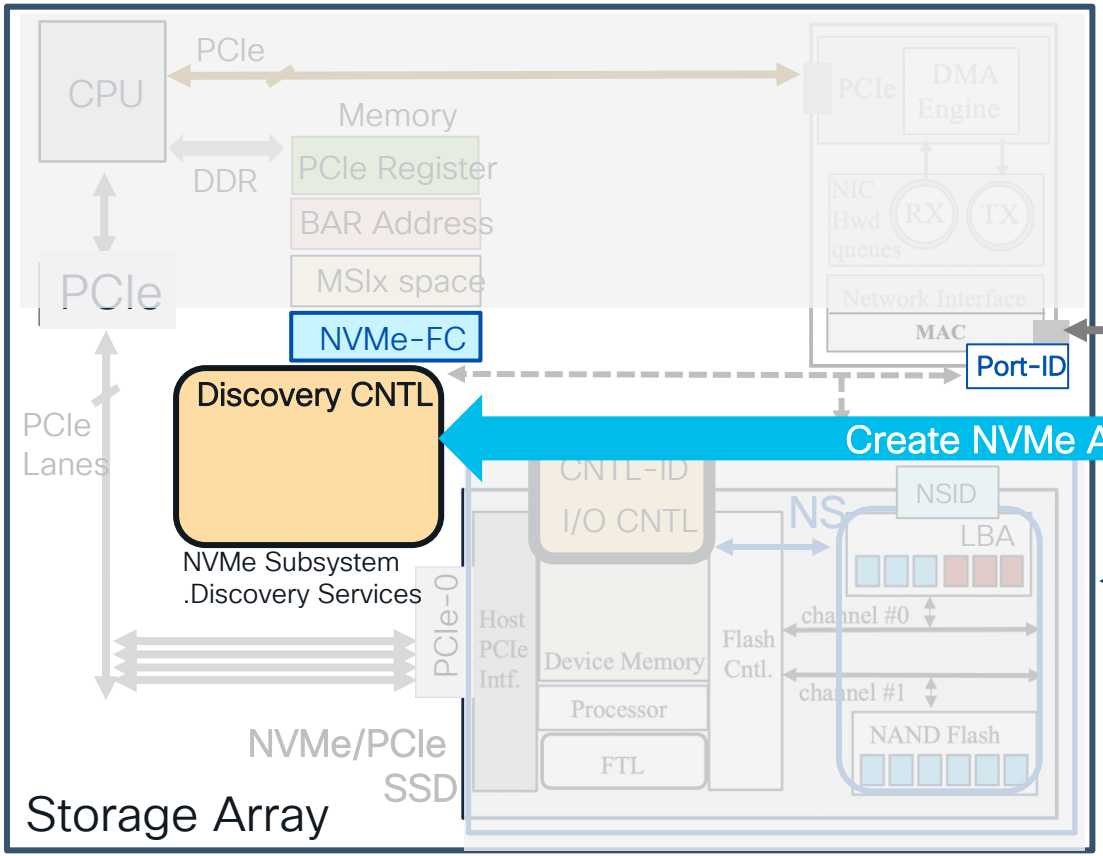
PRLI NVMeoFC
Service Parameter Page

NVMe-FC Protocol Flows (NVMe Association)

Communication relationship between a particular controller and a particular host that encompasses the Admin Queue and all I/O Queues of that controller.



NVMe-FC Protocol Flows (LS _CASS)



Host sends **"Create Association"** NVMe_LS to the Discovery Service subsystem WKA NQN *nqn.2014-08.org.nvmexpress.discovery*

FC Fabric

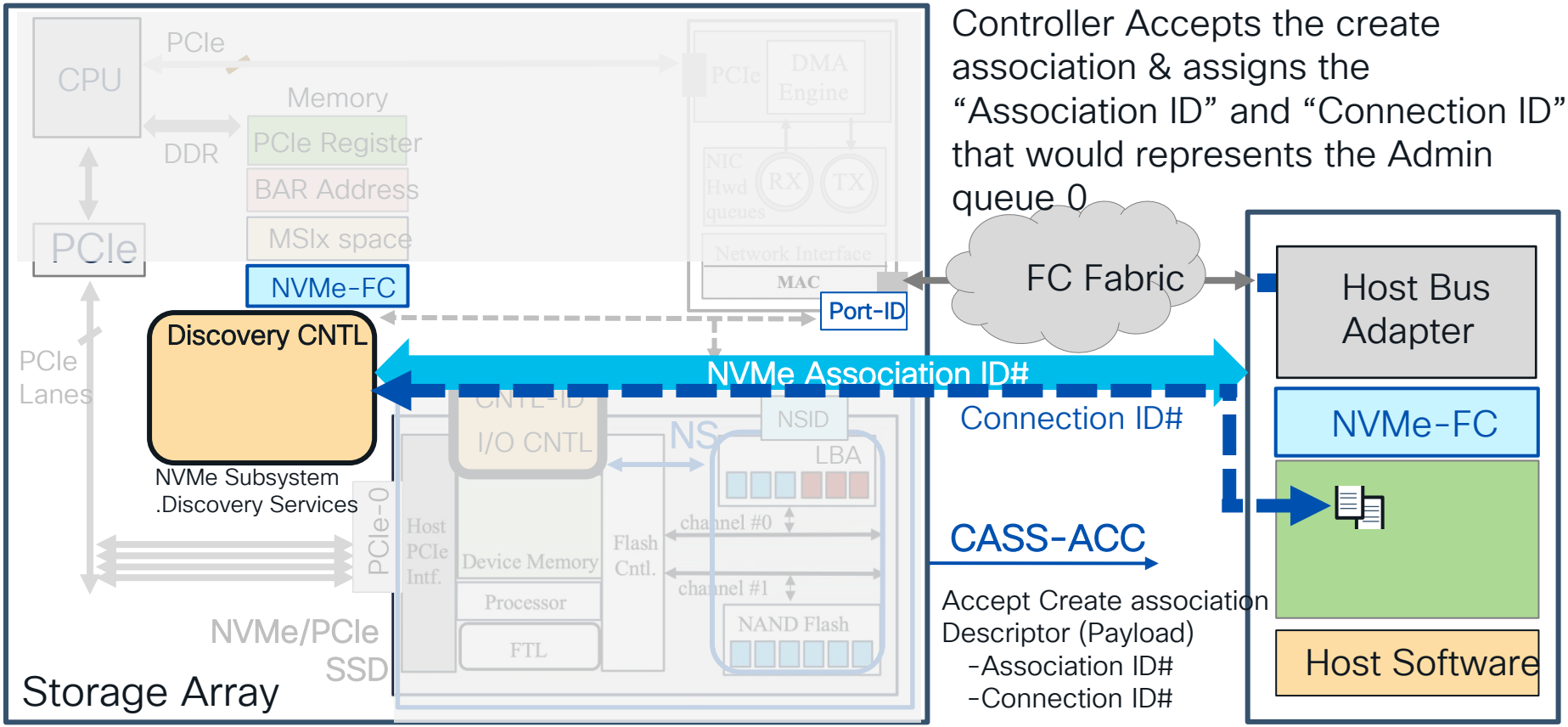
Create NVMe Association

LS_CASS

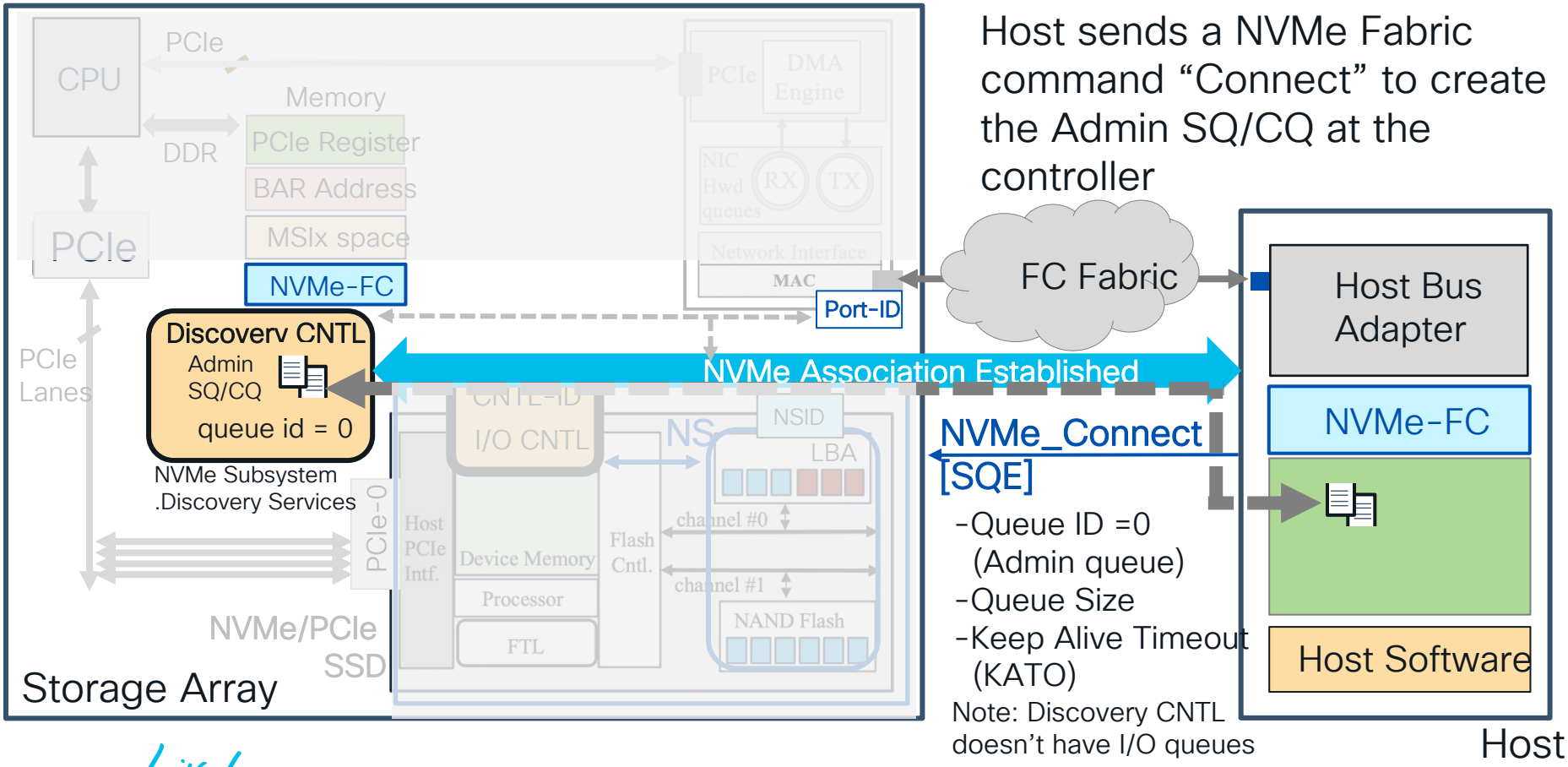
Create association
Descriptor (Payload)
- -CNTLID FFFF (dynamic controller)
- -SQ Size (ADMIN)
- -Host.NQN
- -NVMe Subsystem-WKA

Host

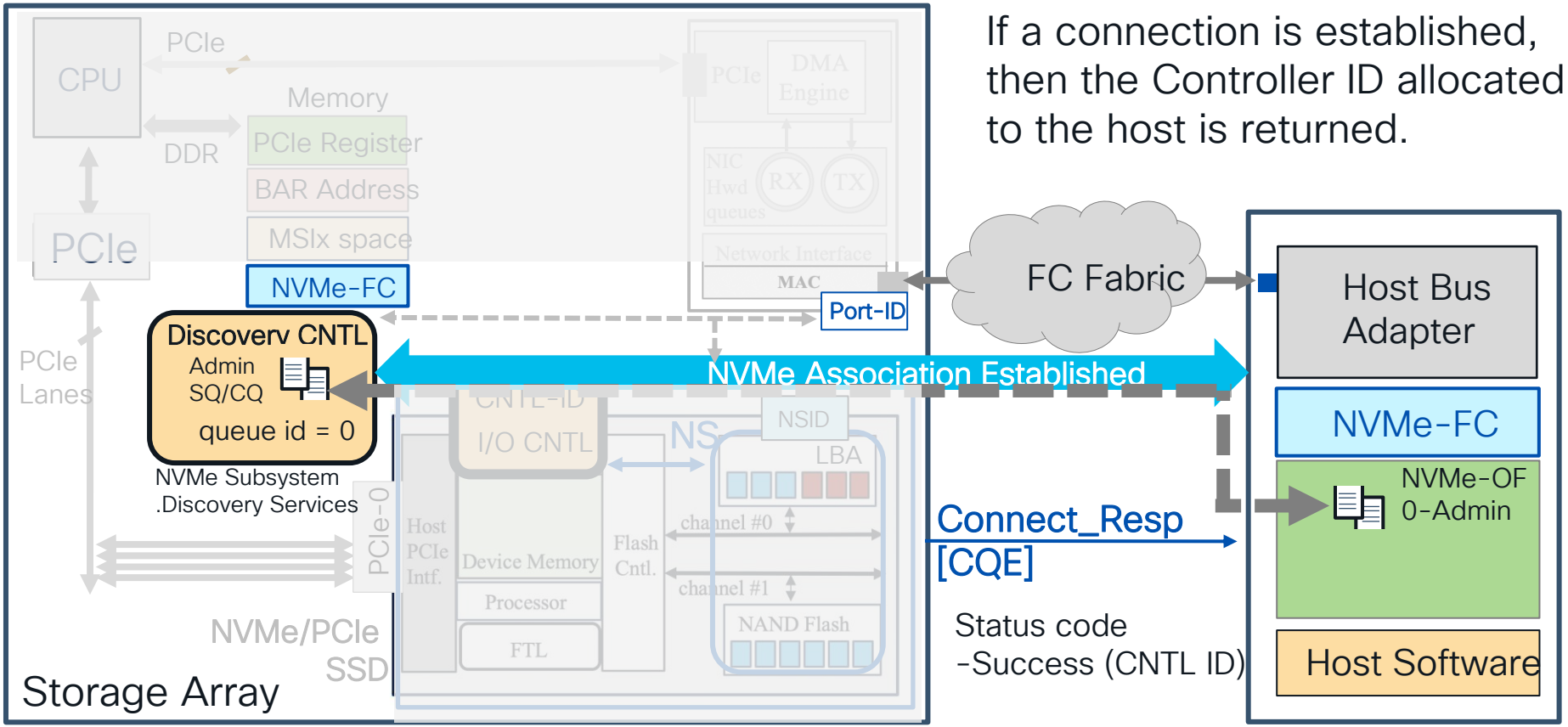
NVMe-FC Protocol Flows (LS CASS_ACC)



NVMe-FC Protocol Flows (Connect Command SQE)

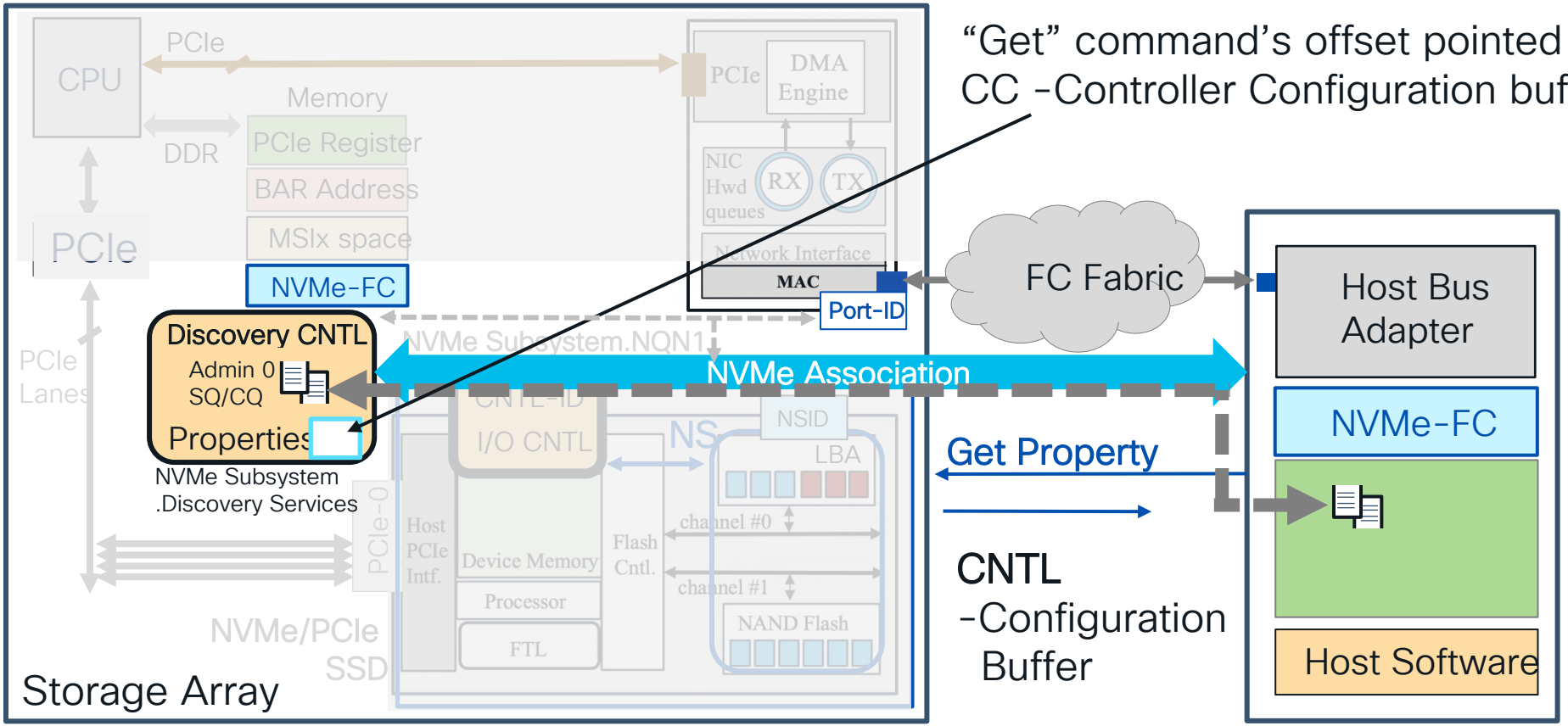


NVMe-FC Protocol Flows (Connect Response CQE)

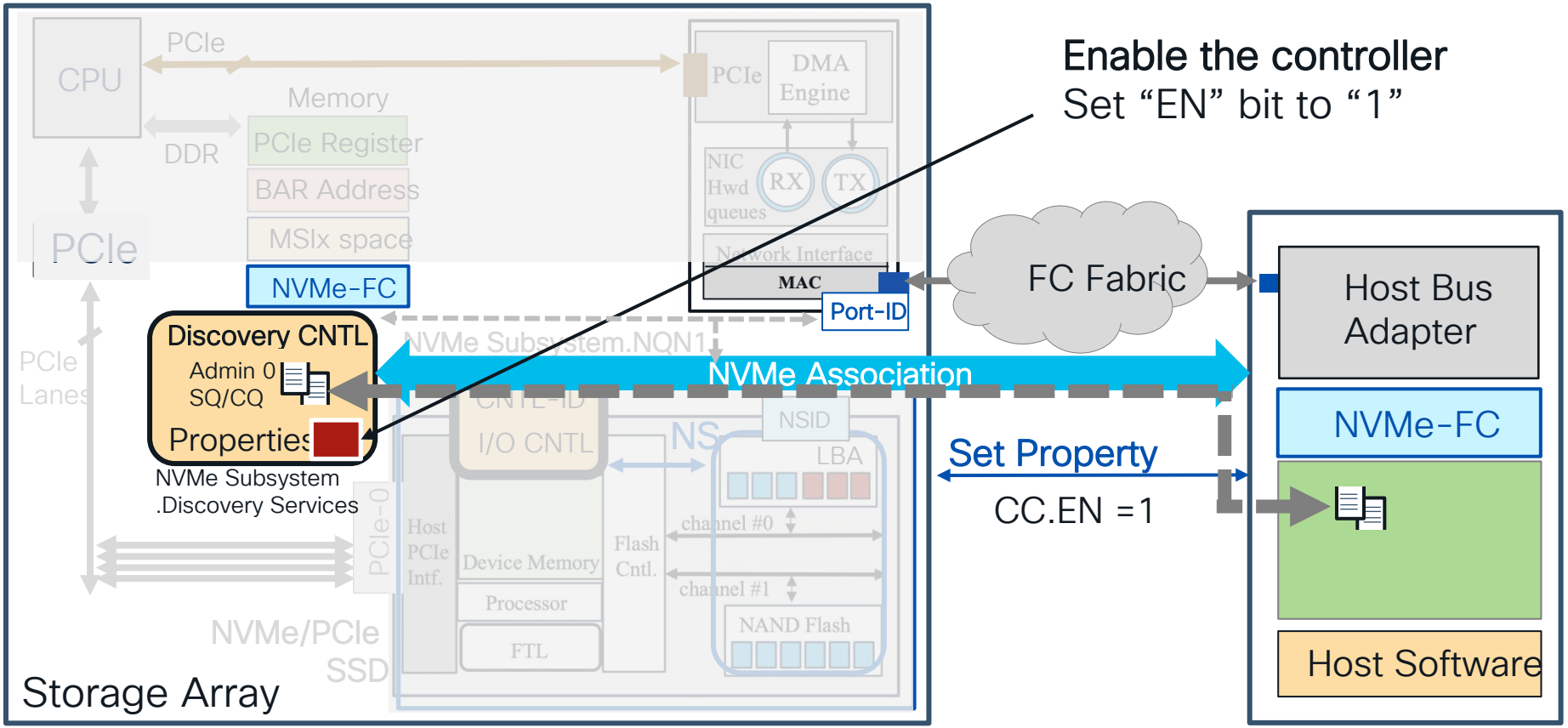


If a connection is established, then the Controller ID allocated to the host is returned.

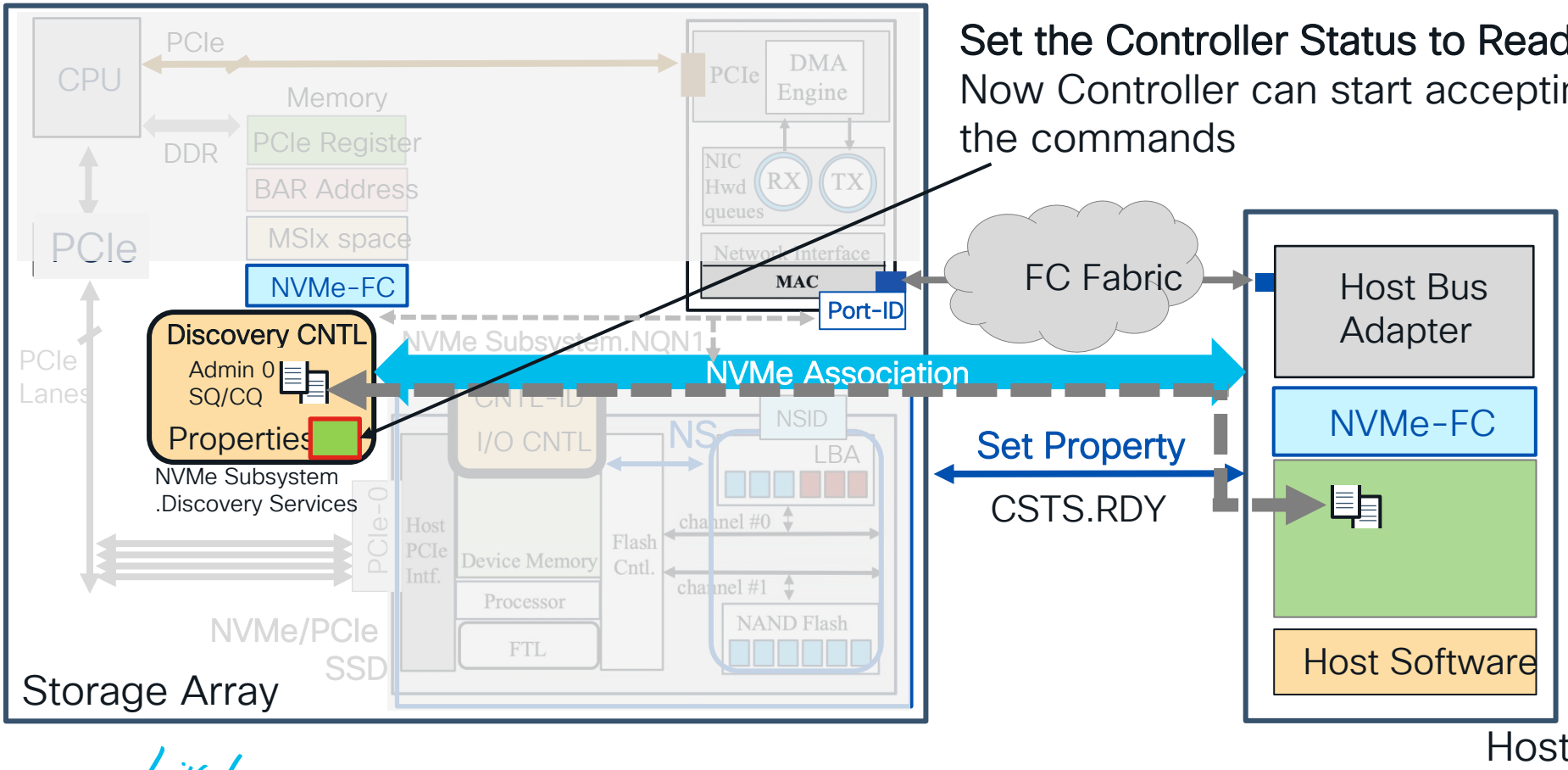
NVMe-FC Protocol Flows (Get Property CC)



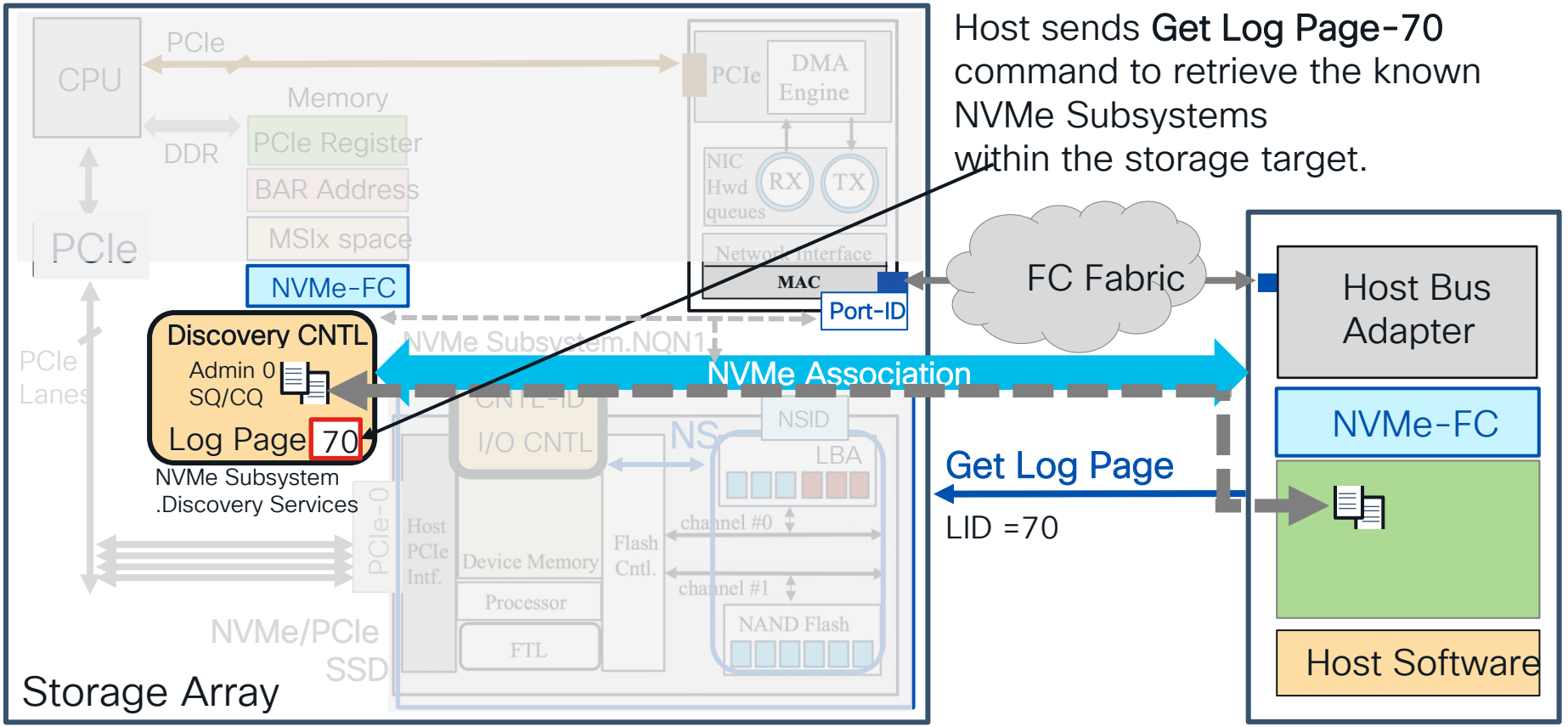
NVMe-FC Protocol Flows (Set Property CC.EN)



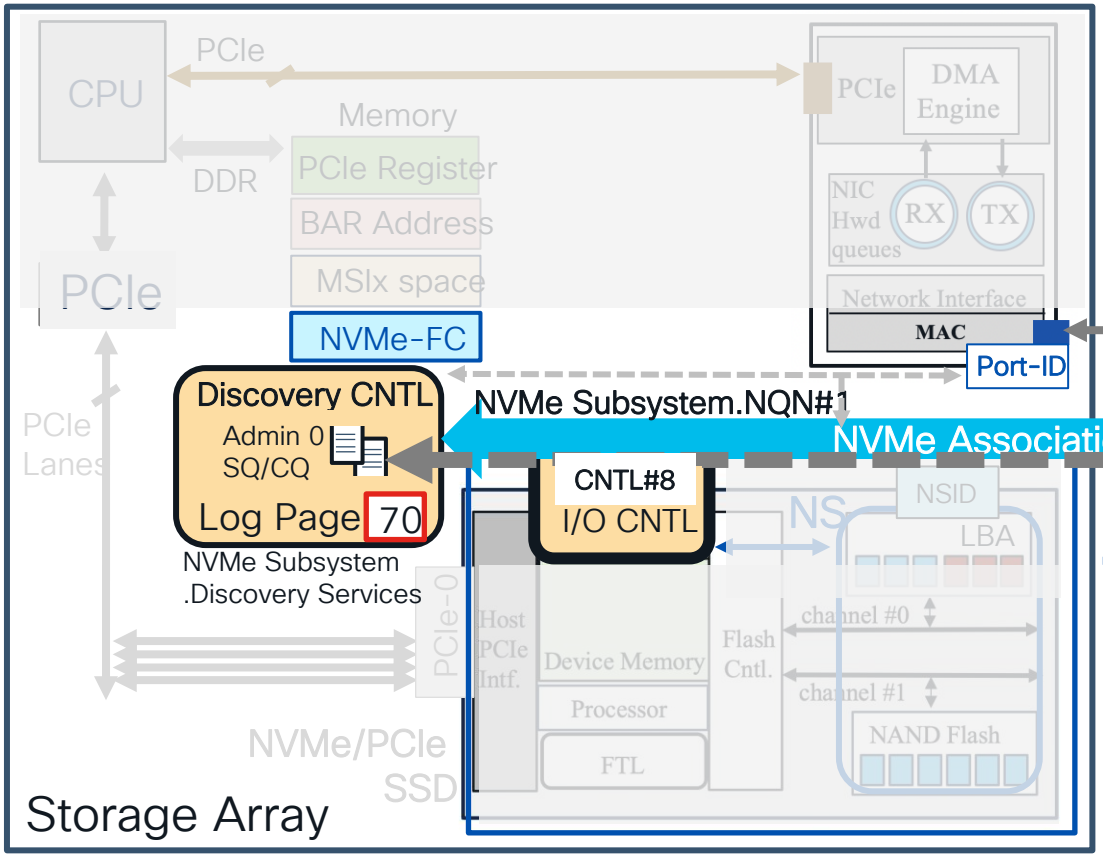
NVMe-FC Protocol Flows (Set Property CSTS.RDY)



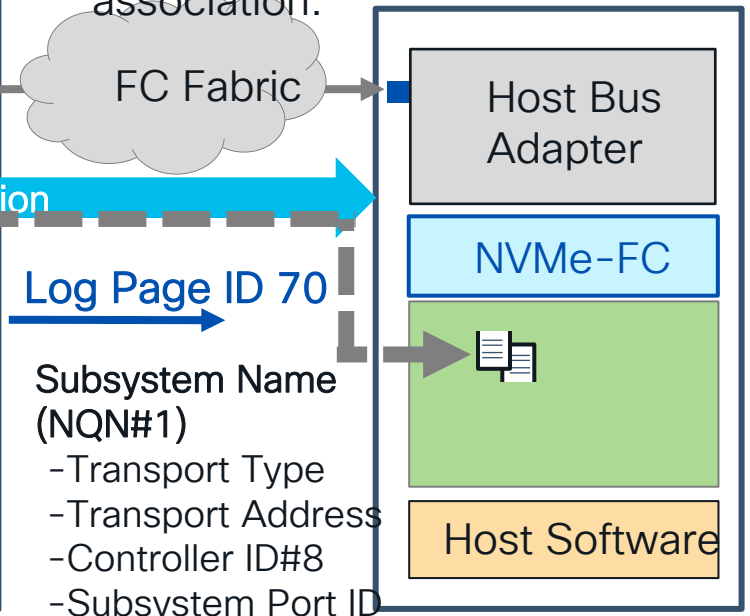
NVMe-FC Protocol Flows (Get Log Page)



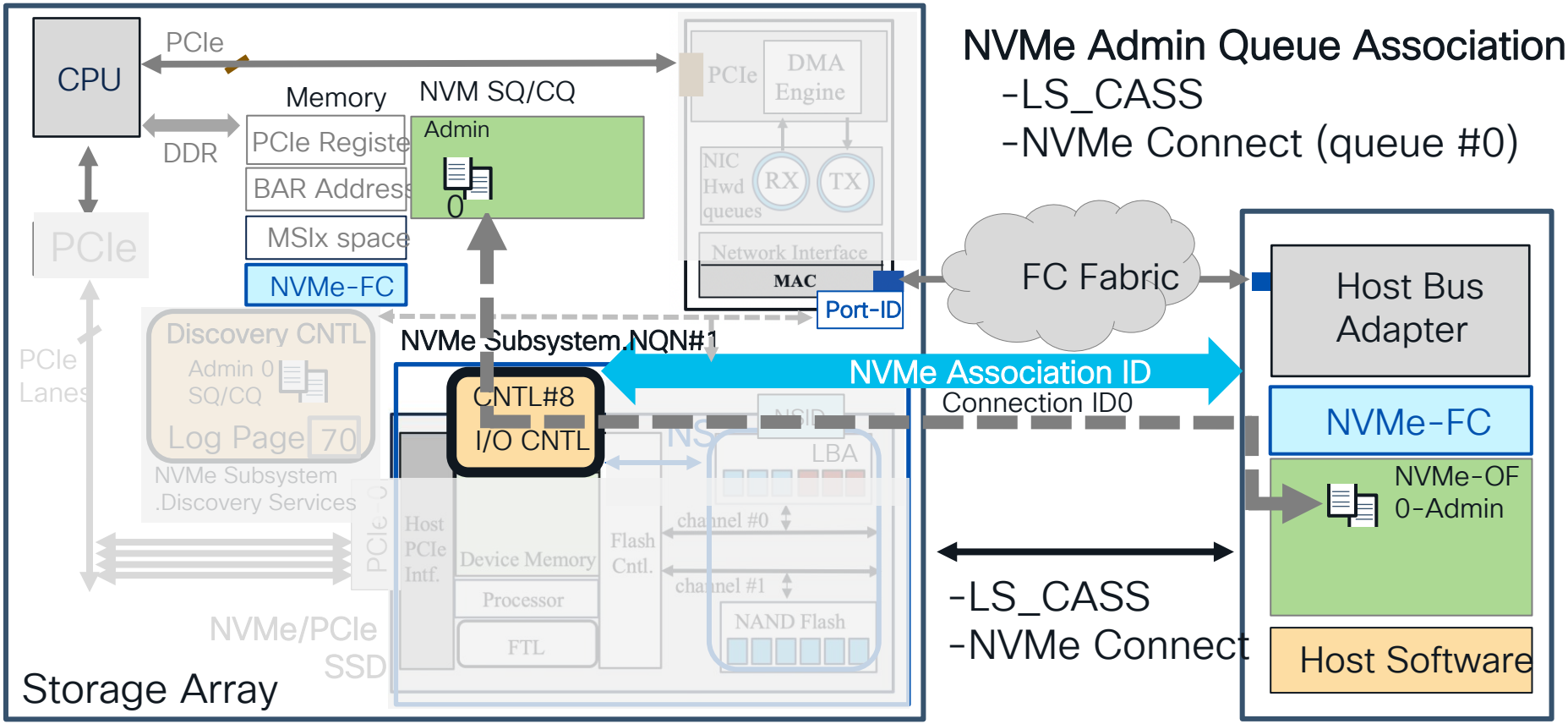
NVMe-FC Protocol Flows (Get Log Page)



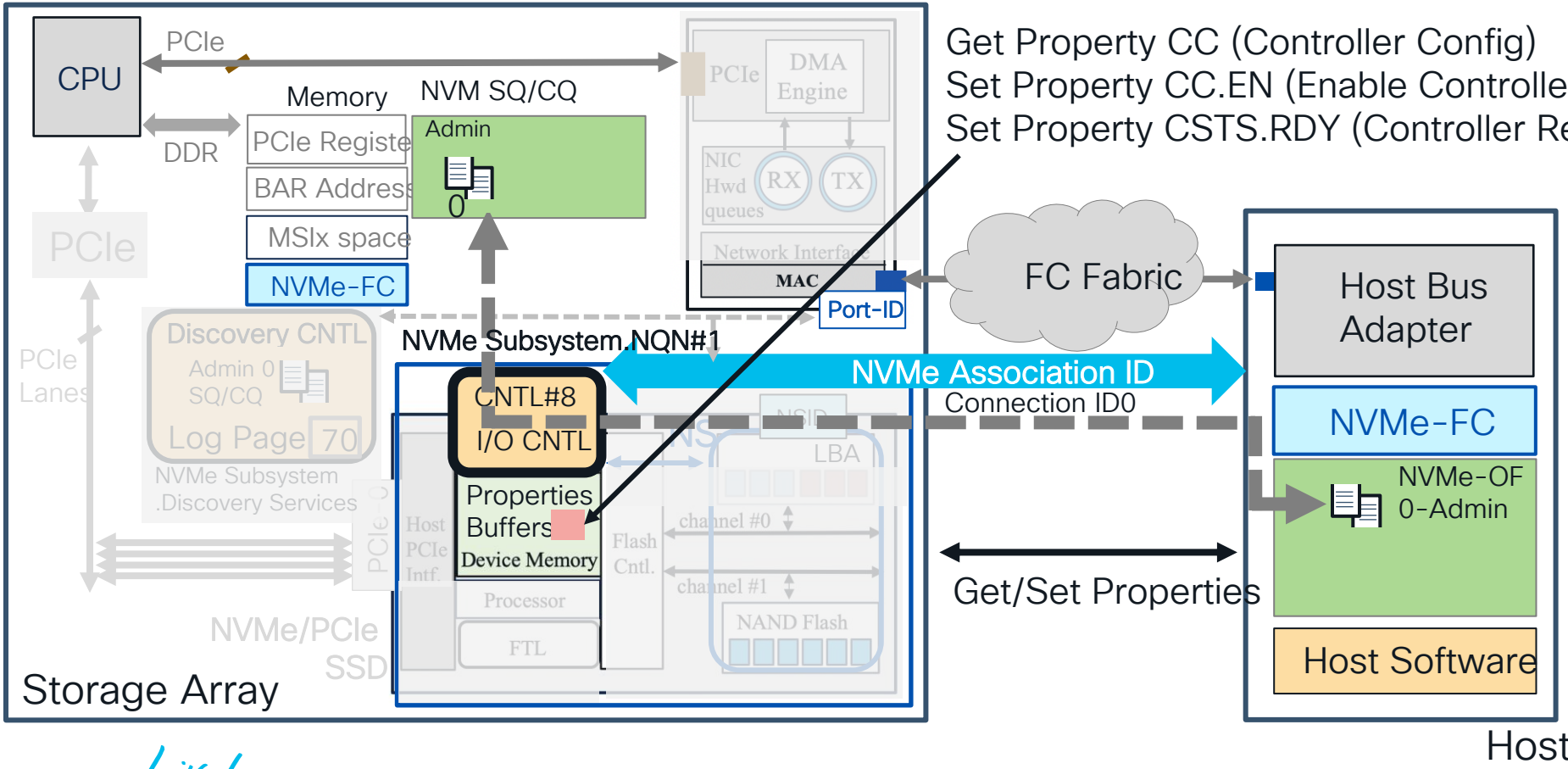
The Discovery Log Page provides an **inventory of NVM subsystems** with which a host may attempt to form an association.



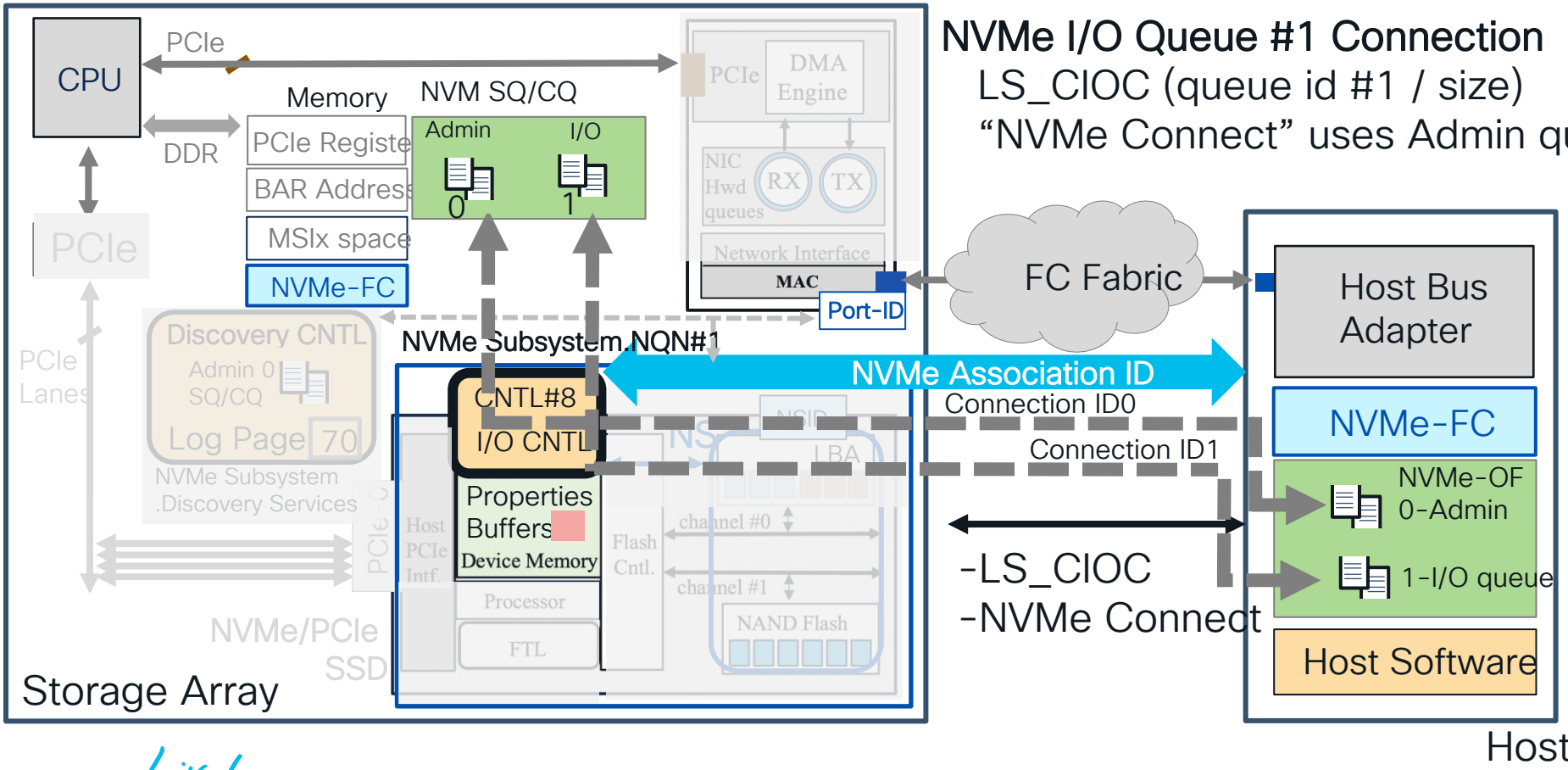
NVMe-FC Protocol Flows (Create Association with I/O CNTL)



NVMe-FC Protocol Flows (I/O CNTL Ready to accept commands)

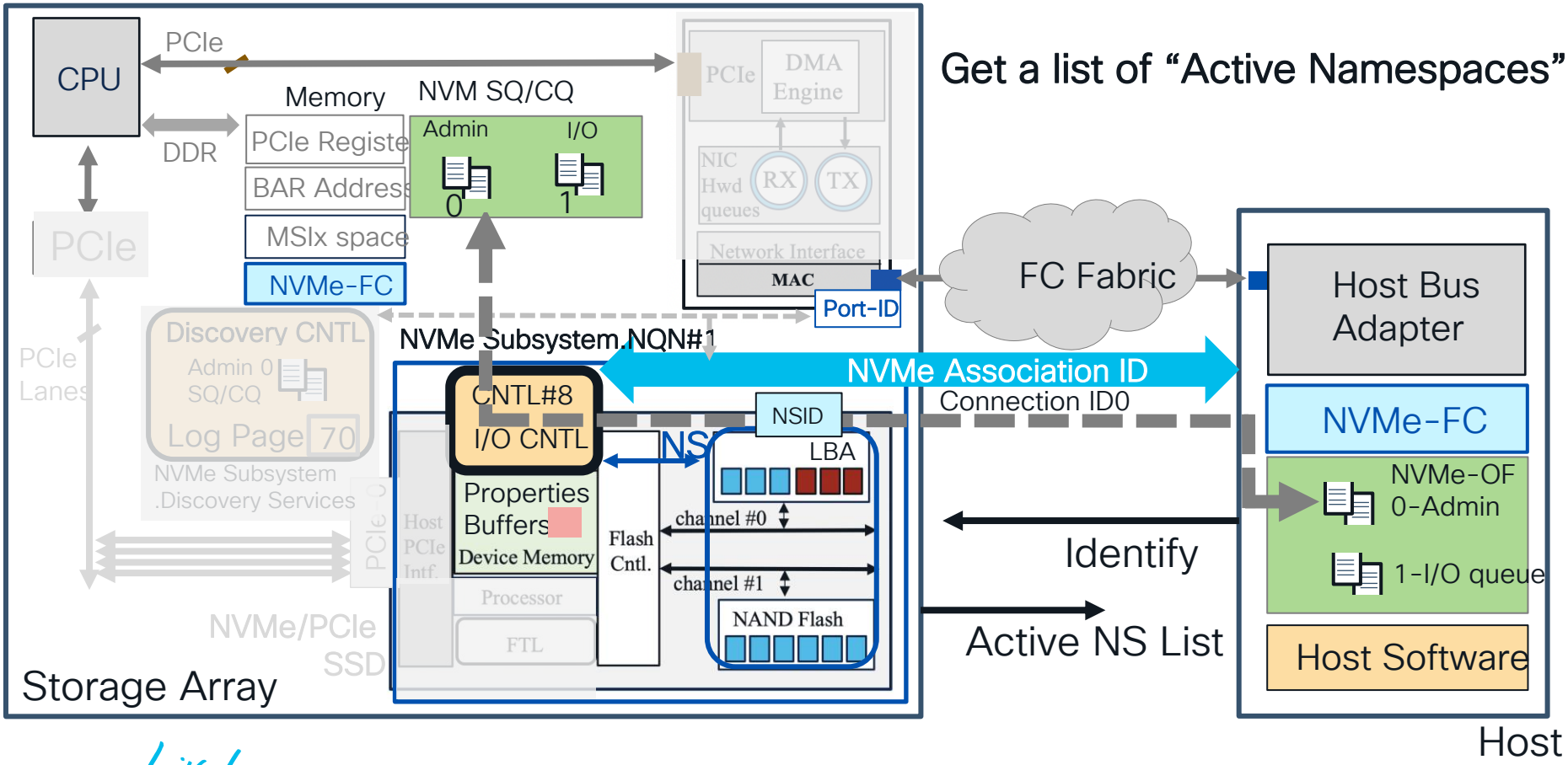


NVMe-FC Protocol Flows (Create I/O Queues)

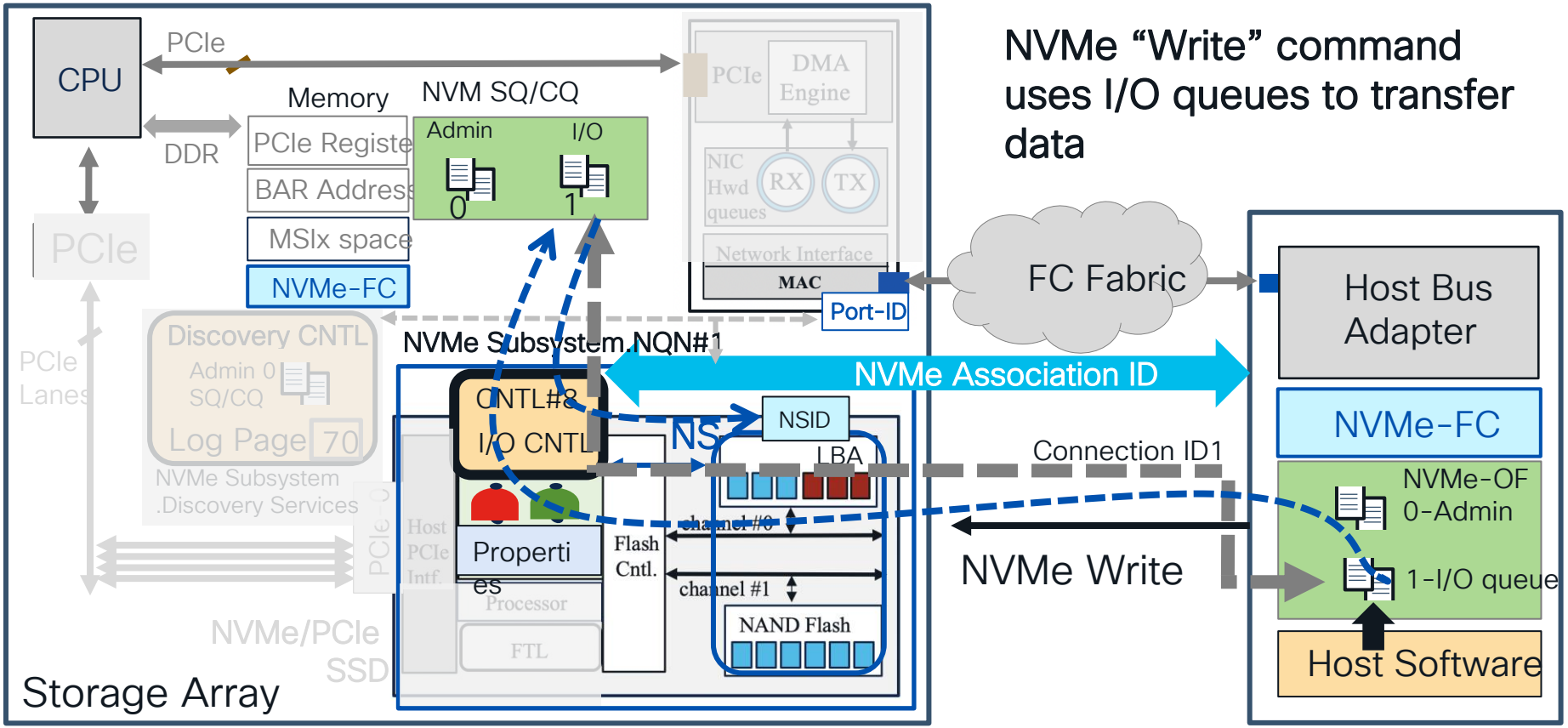


NVMe-FC Protocol Flows (NVMe Identify CNS 02)

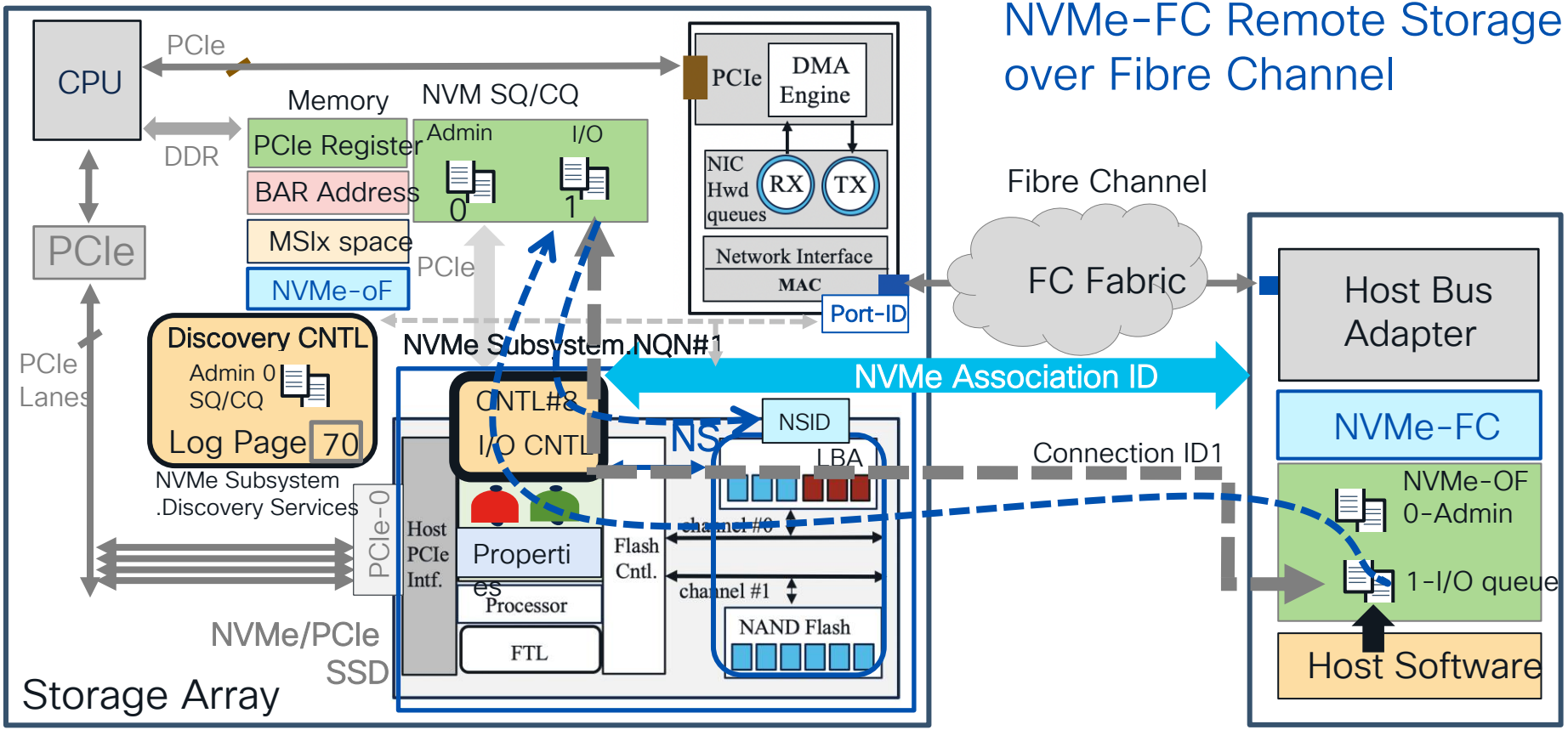
Return



NVMe-FC Protocol Flows (NVMe Write)



NVMe-FC



Best Practices (Do's & Don'ts)

- NVMe/FC protocol and SCSI/FC protocol use the same Fibre channel infrastructure
- Use different VSANs to keep the separation between NVMe and SCSI FC traffic
- NVMe/FC provides higher performance and better error recovery (SLER)
- Today the current speed of Fibre channel is 64G, 128G standard is being worked on
- Cisco MDS provides rich ASIC based NVMe/FC analytics capability with dedicated additional NPU for further analysis of NVMe frames

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

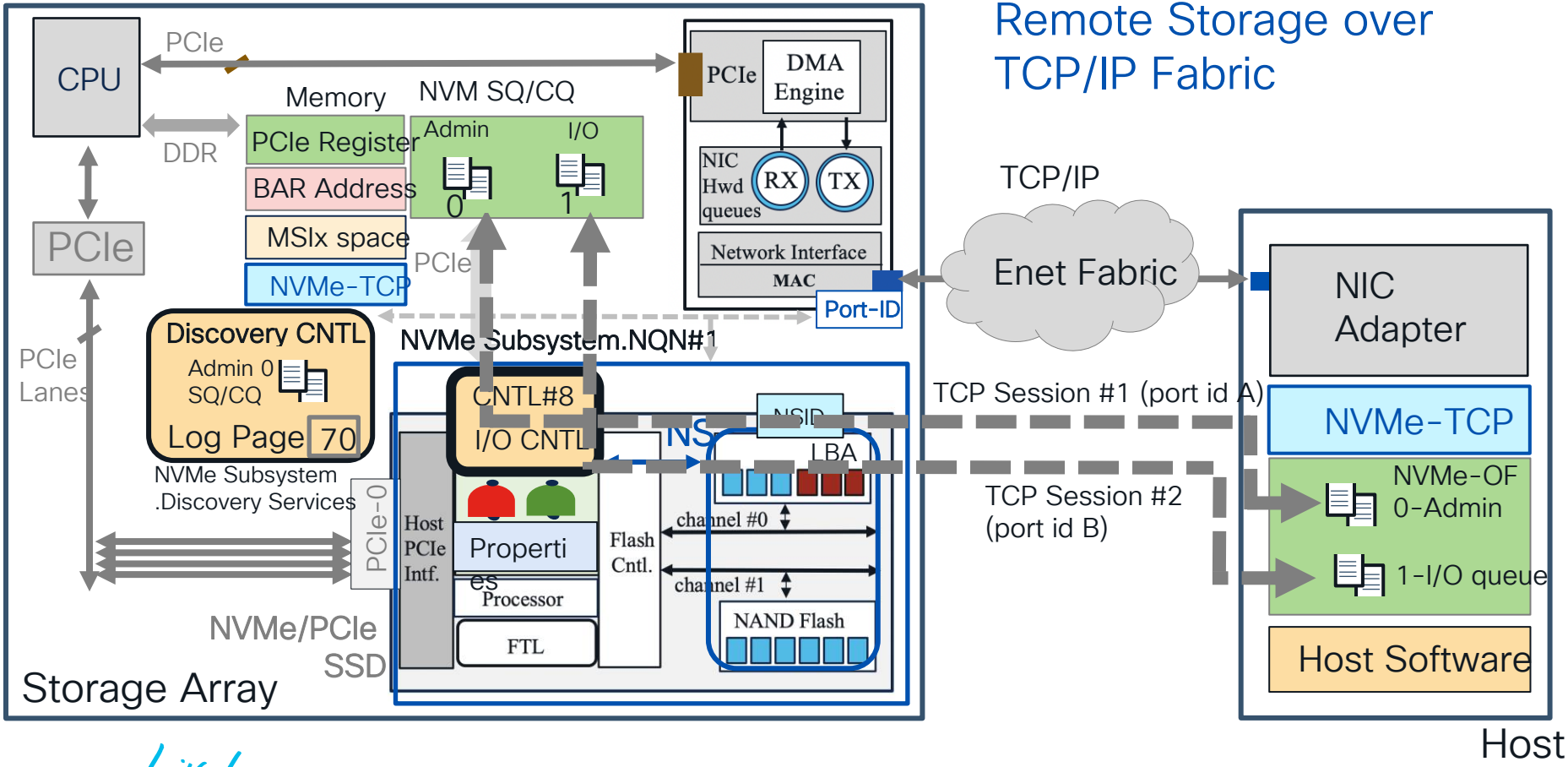
3-NVMe Transport Options (NVMe-TCP)

4-NVMe Datacenter Design

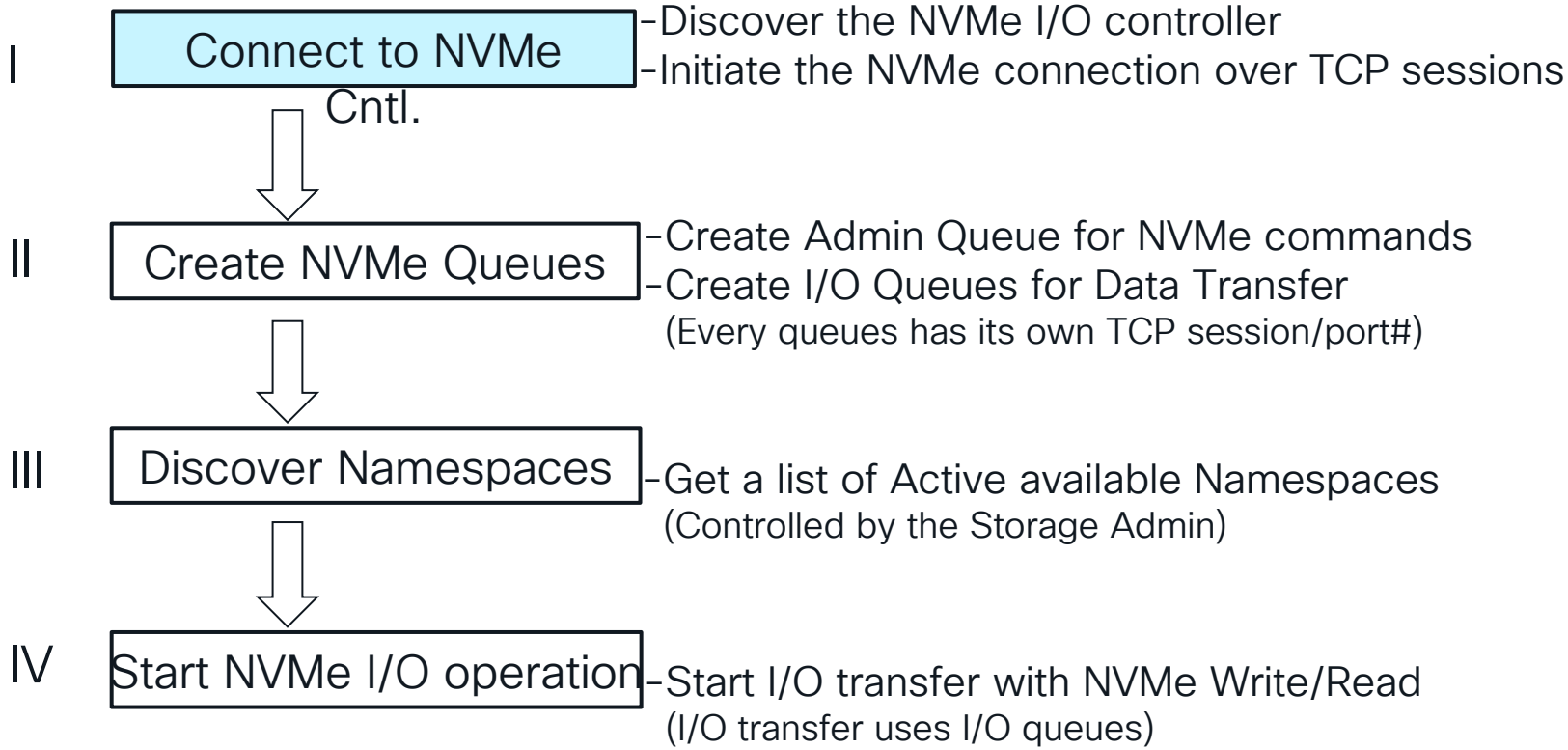
5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

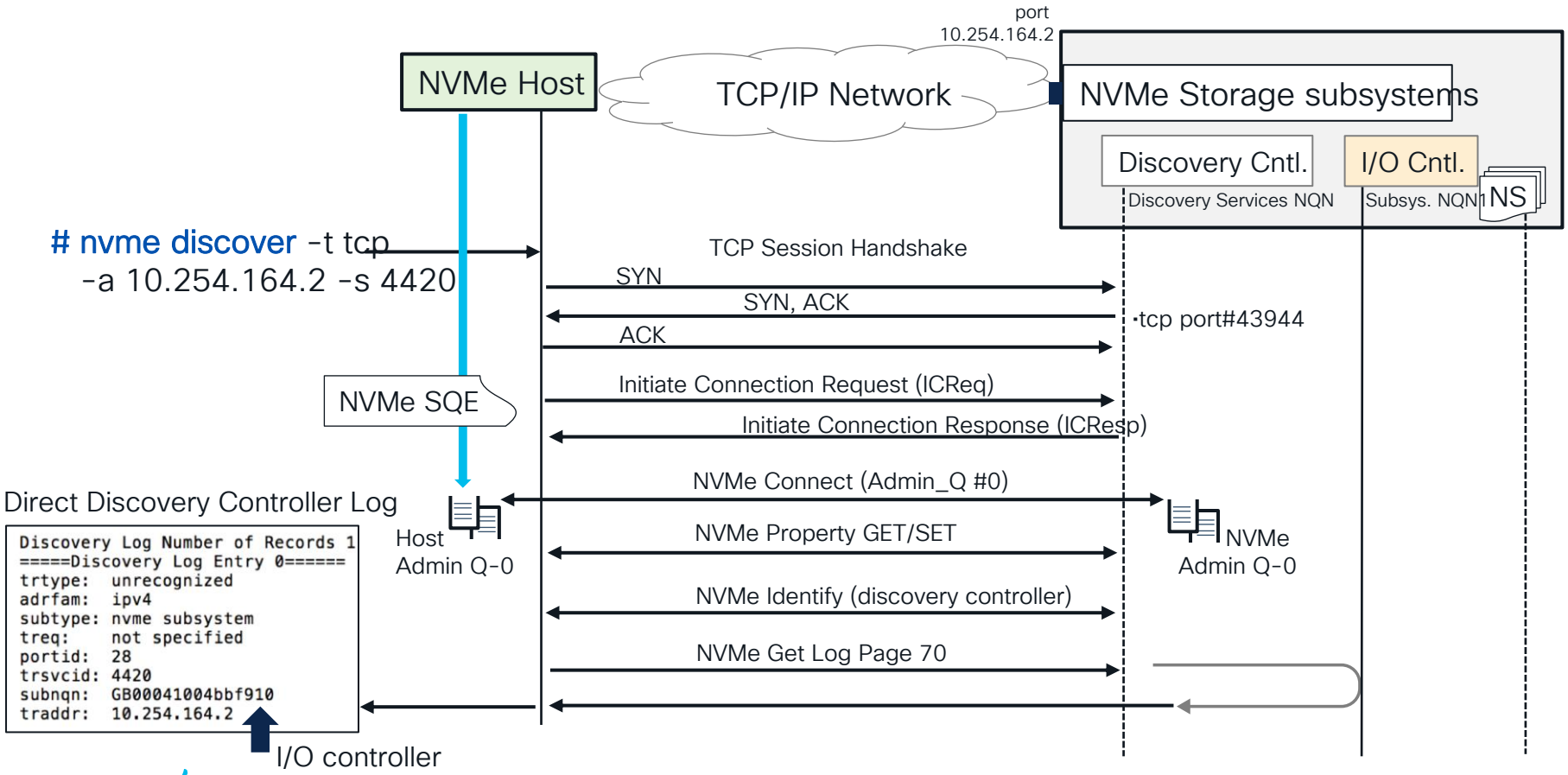
NVMe-TCP



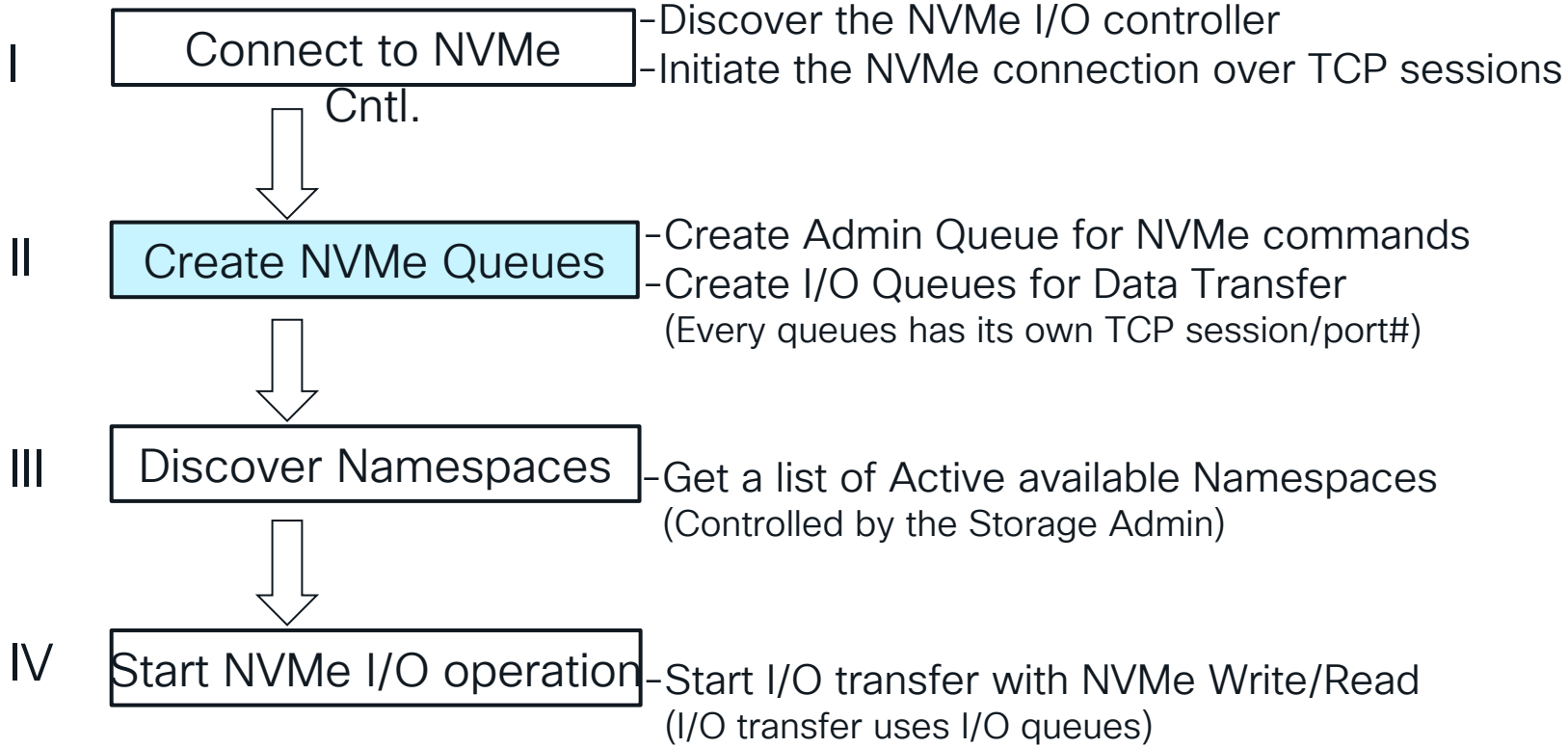
NVMe-TCP steps



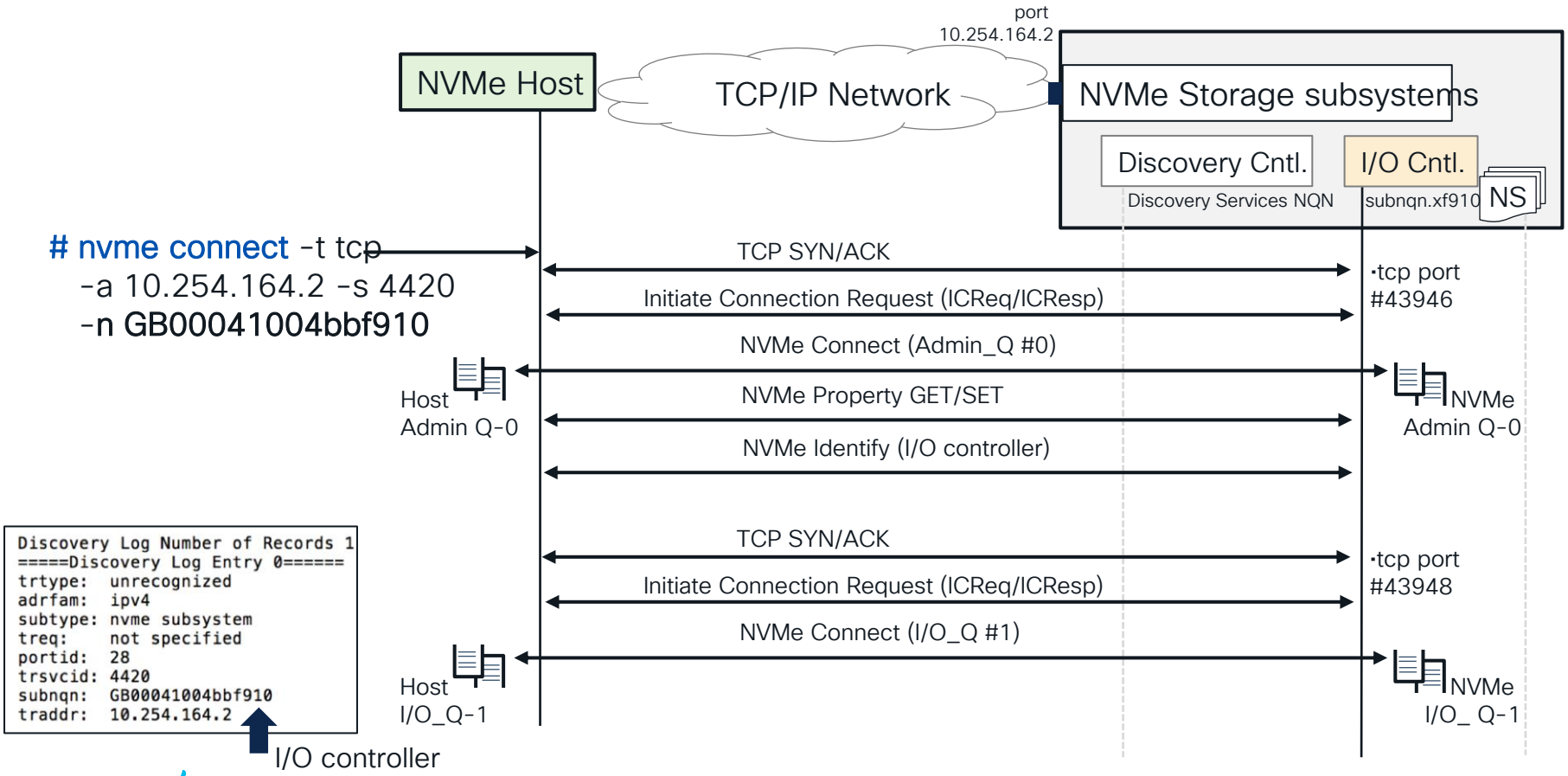
1-Discover & Connect to NVMe Controller



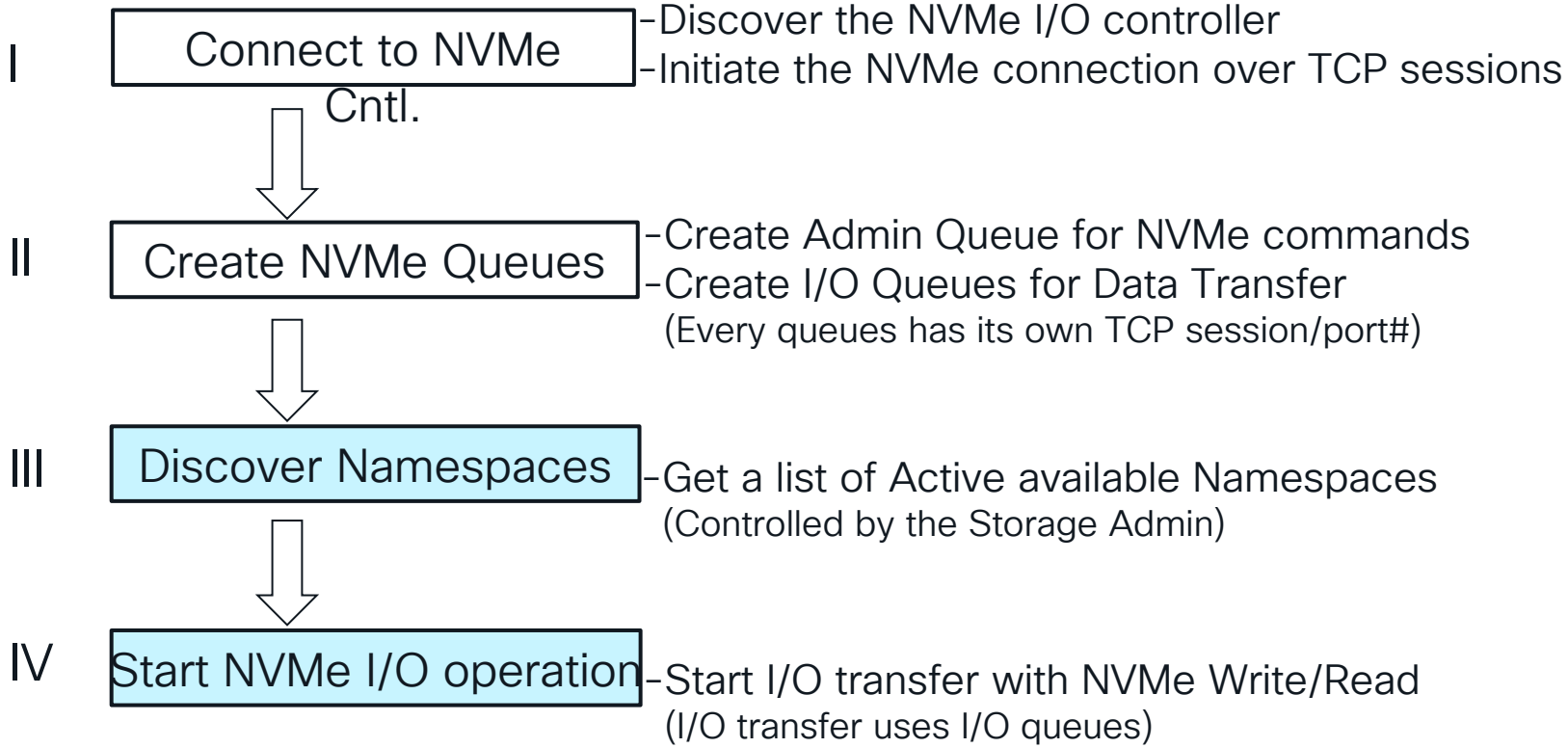
NVMe-TCP steps



2-Create NVMe Queues

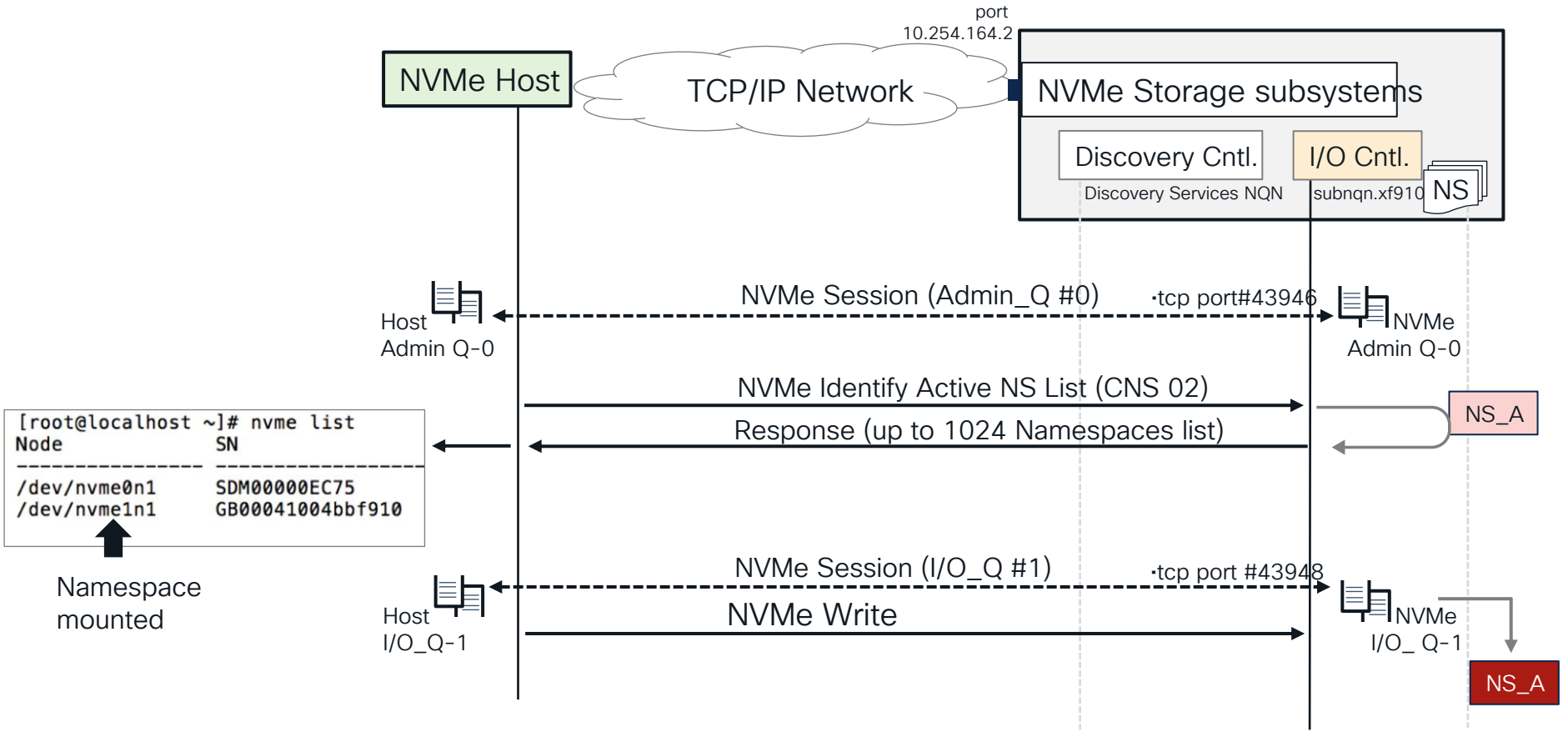


NVMe-TCP steps



3,4-Discover Namespaces, start I/O

Jump to CDC



Best Practices (Do's & Don'ts)

- Many HBA vendors offer SmartNIC that offload NVMe/TCP and other functions
- For NVMe/TCP traffic management over lossy ethernet, use QoS/ECN
- Do not use "deep buffers" for NVMe/TCP as it would hold the packet longer in the buffers that would violate the max. 10us additional fabric delay best practices guidance. Cisco Nexus 9k offer "smart buffers" that minimize the buffer delays.
- NVMe/TCP uses "keepalive" (mice flows) to determine if the connection is still alive, Cisco Nexus 9k offers automatic separation of Elephant/Mice flow and prioritization of mice flows (keepalives) helps NVMe/TCP traffic through congested networks
- Cisco ACI/APIC can auto configure the NVMe/TCP/RoCEv2 across the entire fabric

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

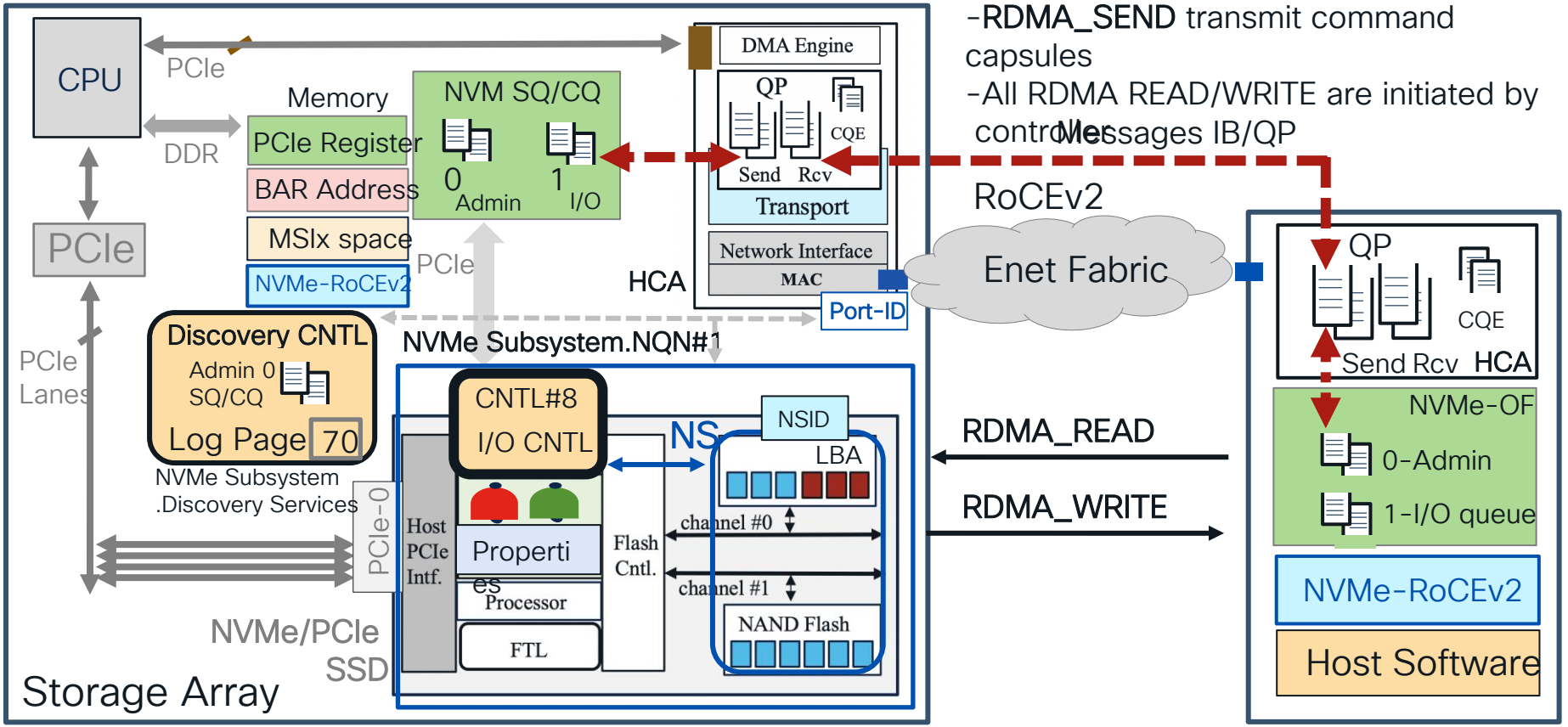
3-NVMe Transport Options (NVMe-RoCEv2)

4-NVMe Datacenter Design

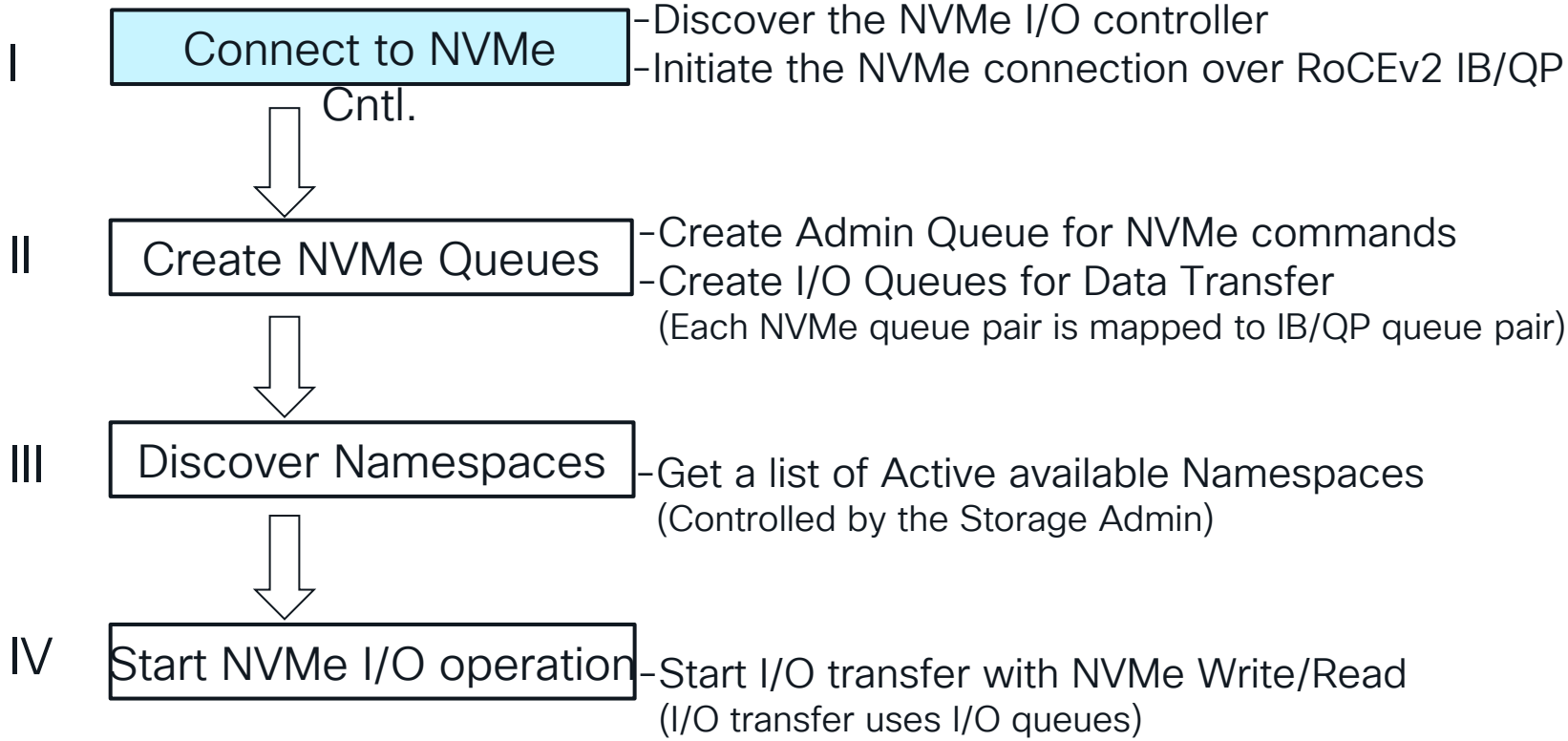
5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

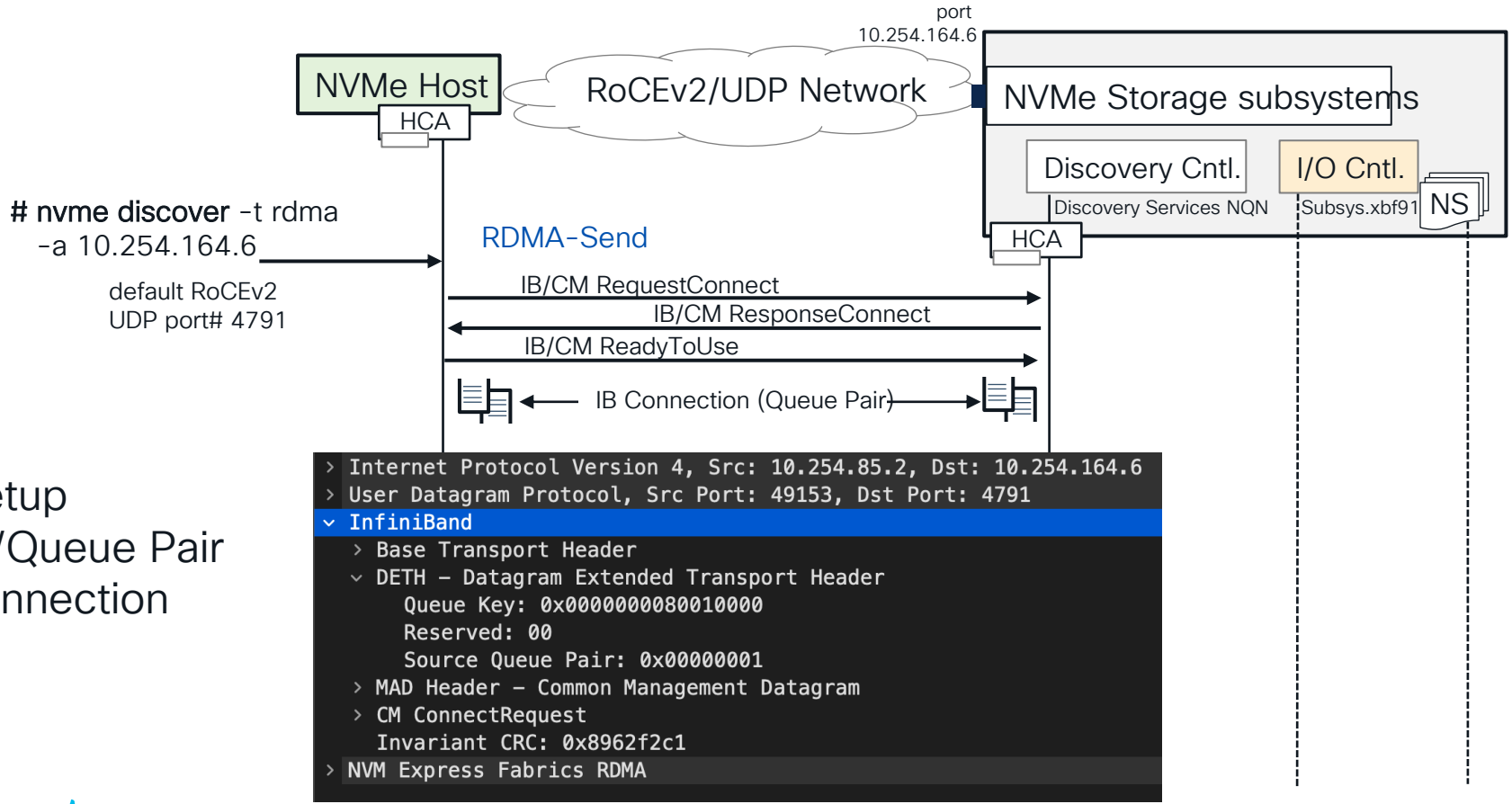
RoCEv2_Queues maps to NVMe_Queues



NVMe-RoCEv2 steps

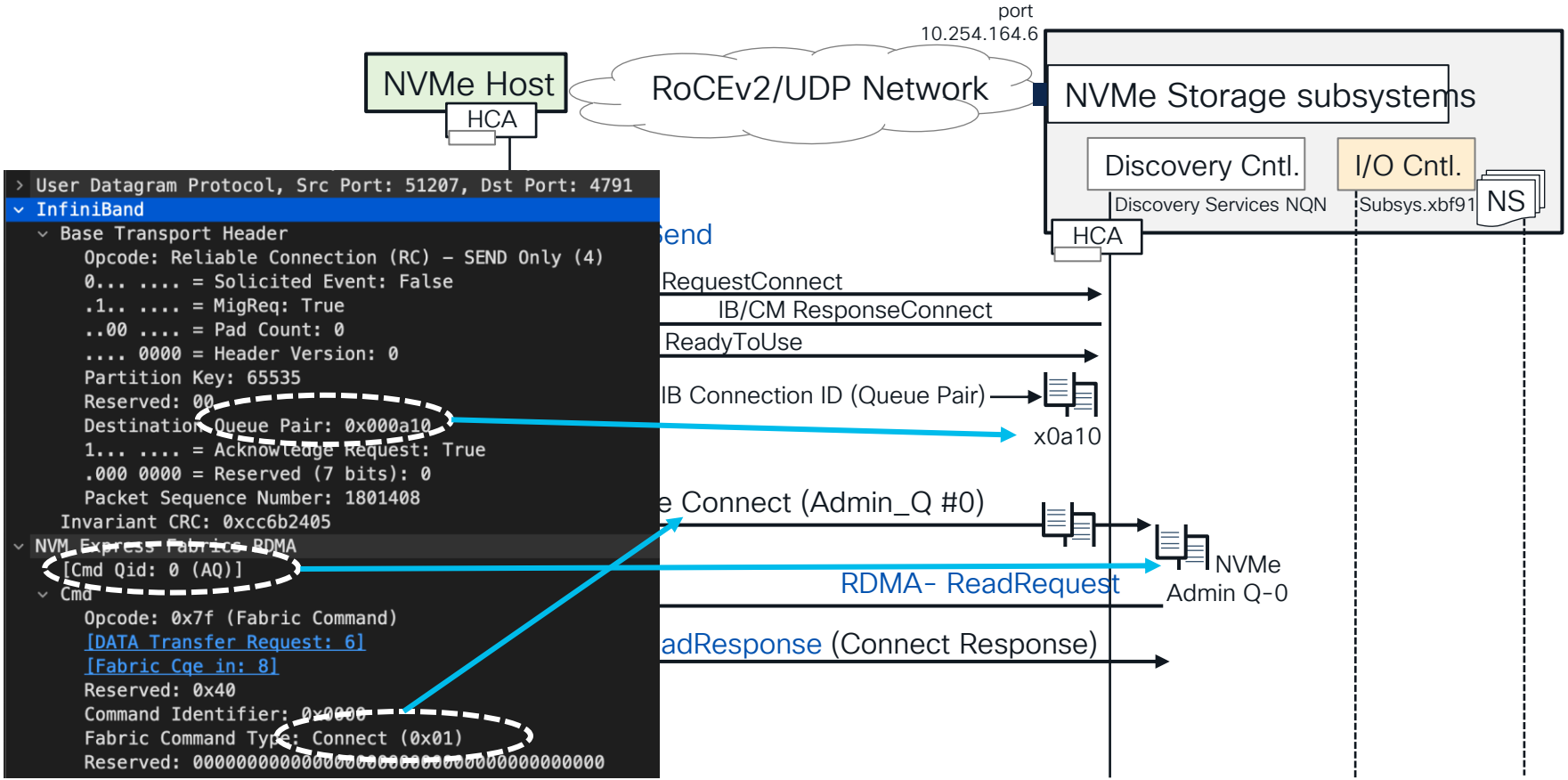


NVMe-RoCEv2 (Start IB/queue pair for Discovery Controller)

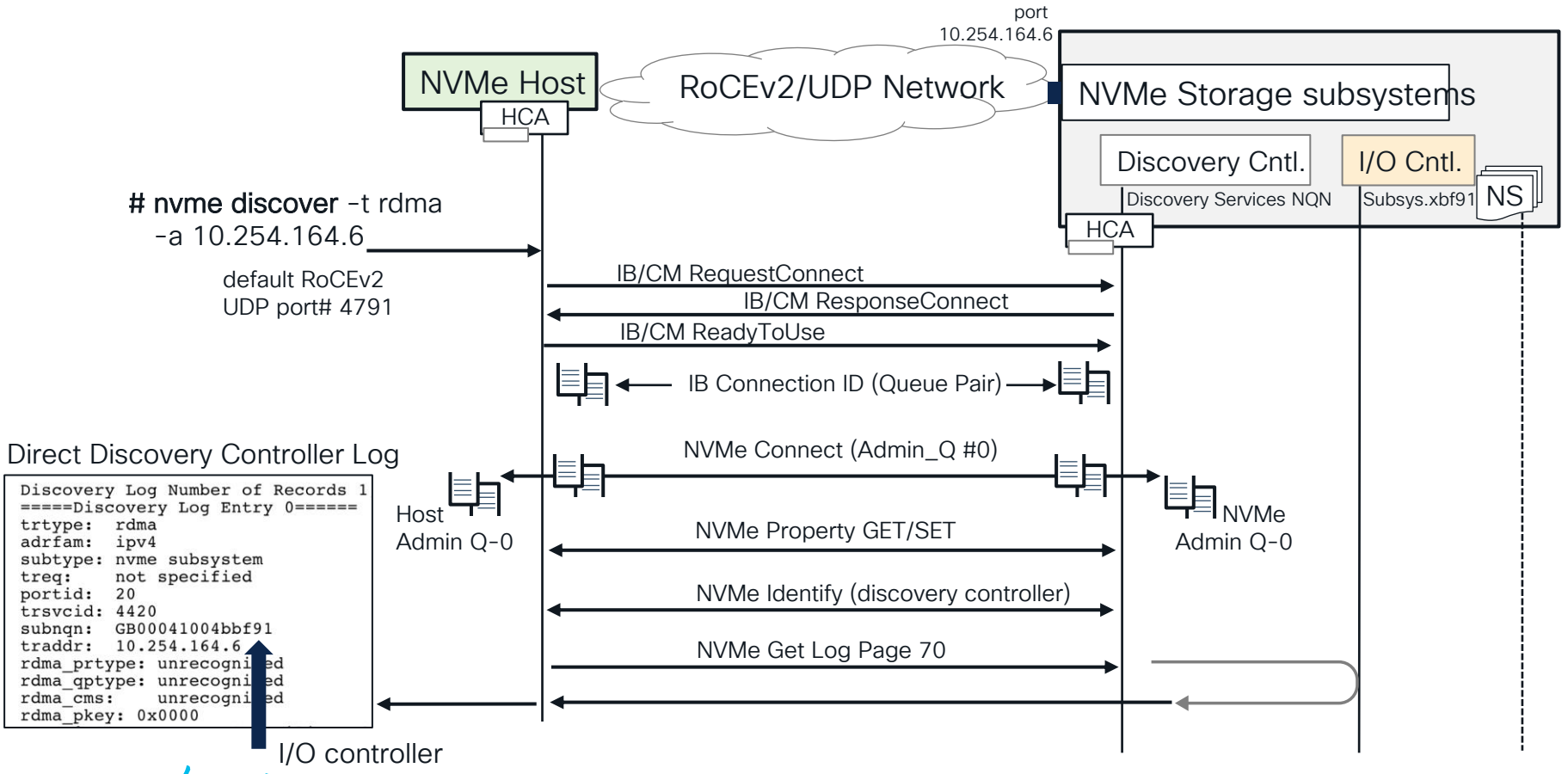


Setup
IB/Queue Pair
connection

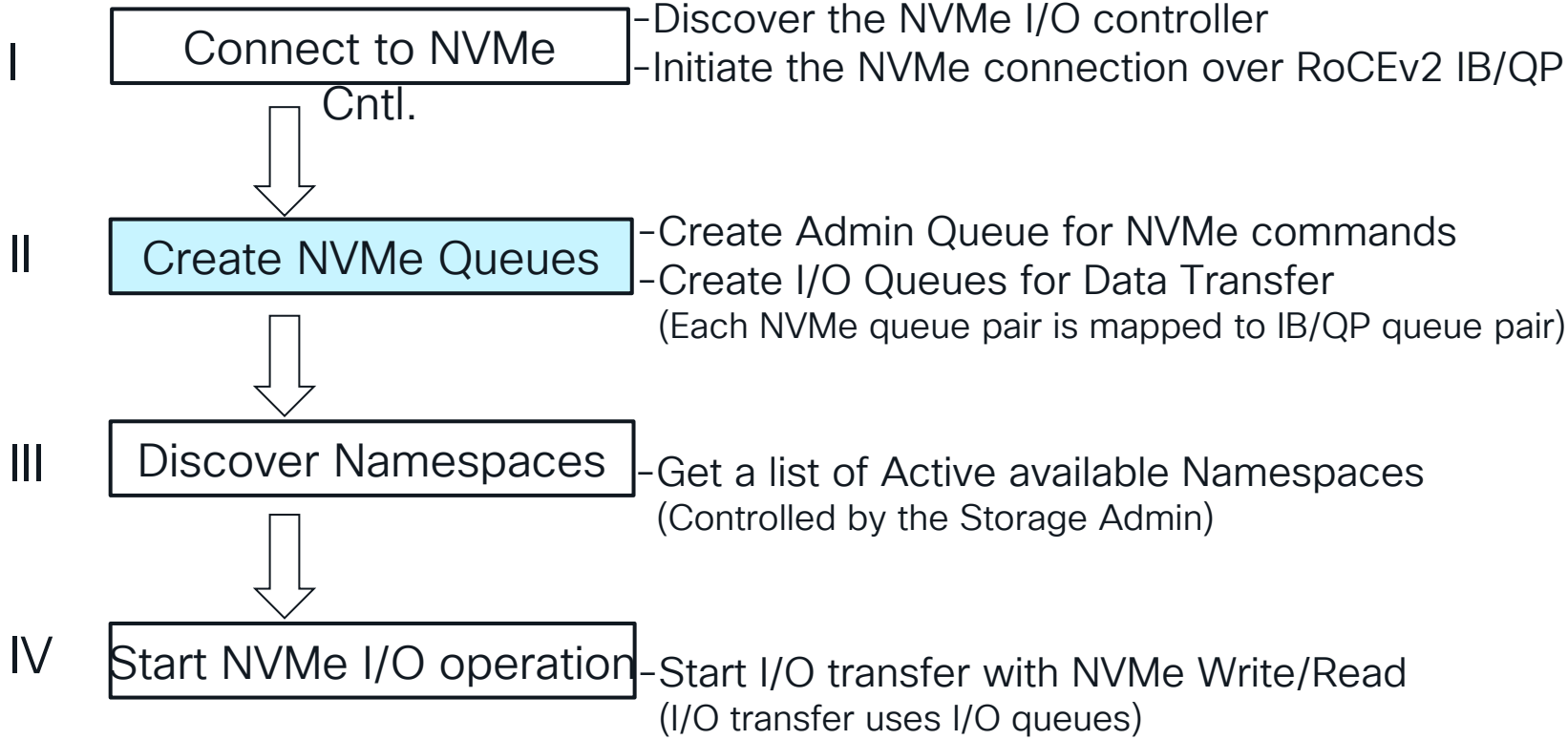
NVMe-RoCEv2 (Connect to Discovery Controller)



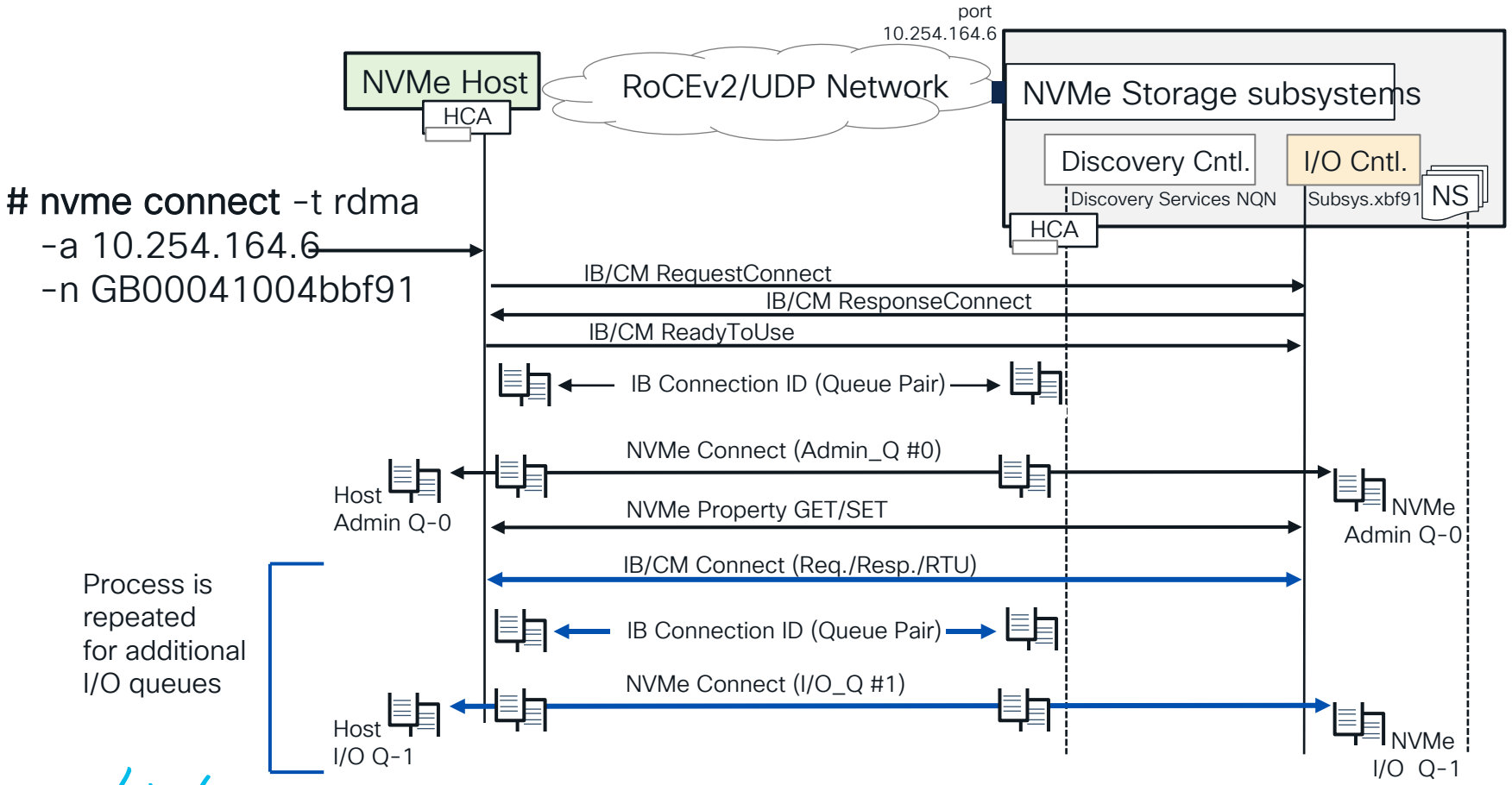
NVMe-RoCEv2 (Get I/O controller details)



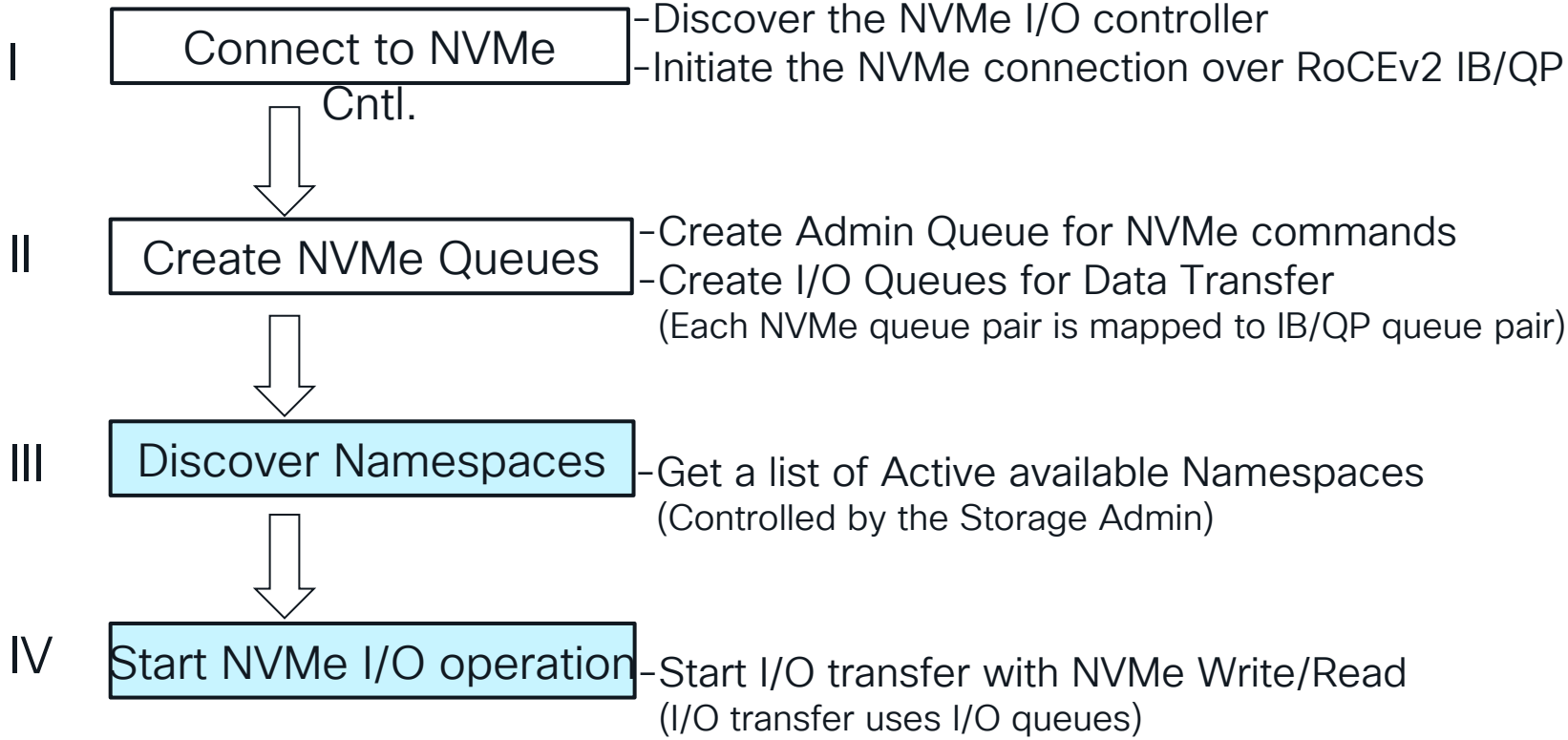
NVMe-RoCEv2 steps



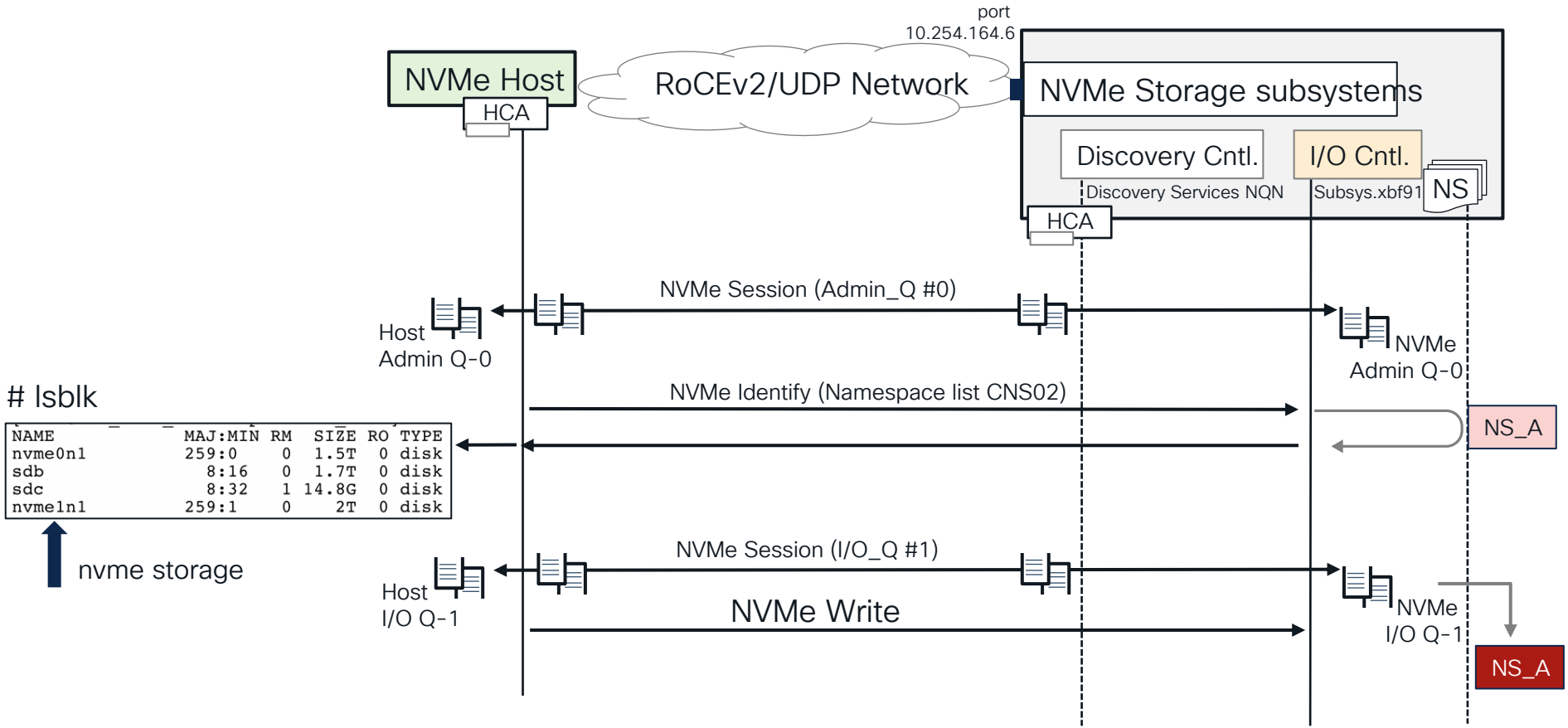
NVMe-RoCEv2 (Create NVMe I/O queues)



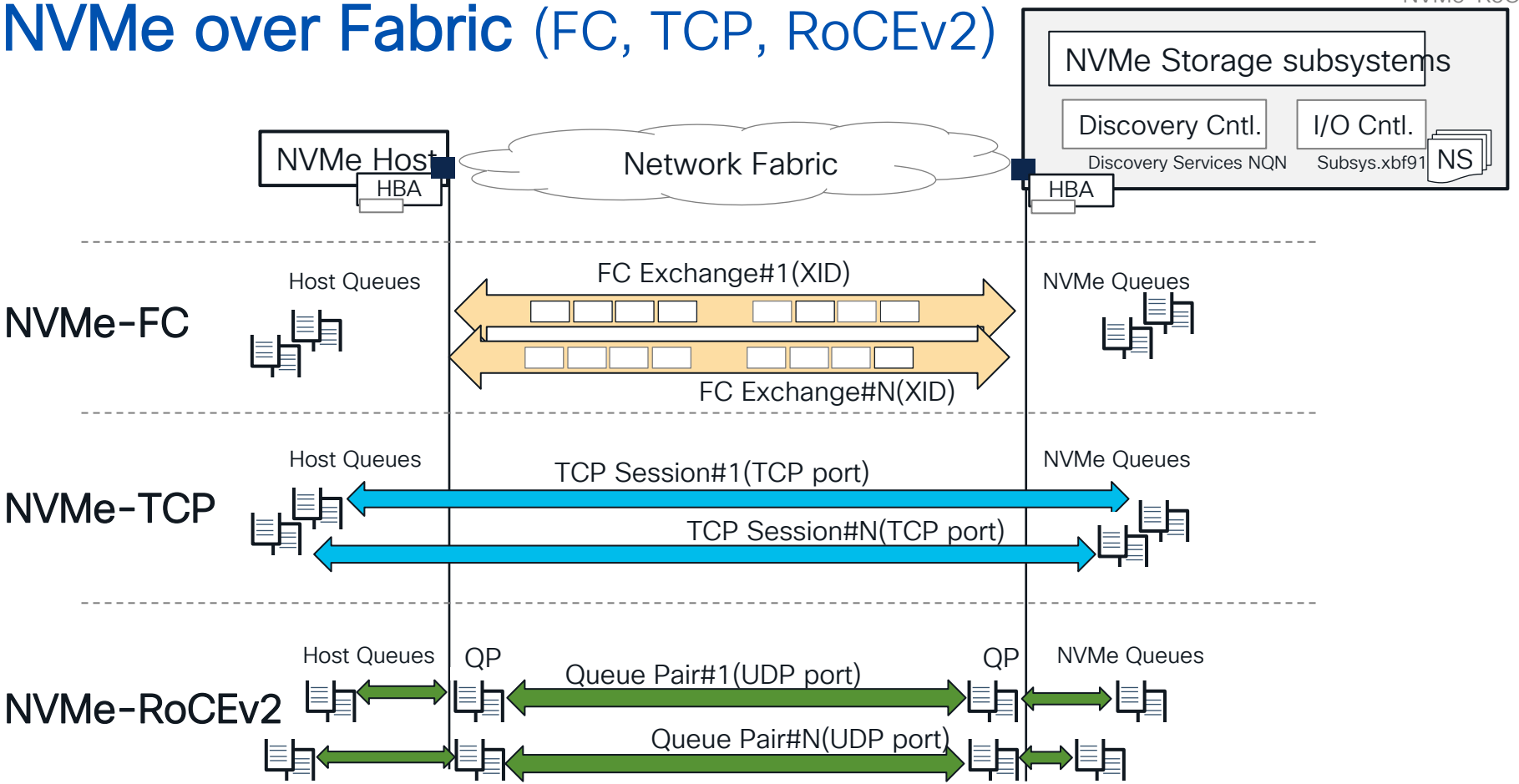
NVMe-RoCEv2 steps



NVMe-RoCEv2 (Namespace and Write)



NVMe over Fabric (FC, TCP, RoCEv2)



Best Practices (Do's & Don'ts)

- Usually NVMe/RoCEv2 is deployed within a Rack using loss-less ethernet
- Traffic engineering is managed via DSCP, PFC, ECN, DCQCN, IB/CNP features
- Resilient RoCEv2 can be used to limited scaling of NVMe/RoCEv2 beyond Rack
- NVMe/RoCEv2 does provide the best performance among all NVMe-oF options
- Troubleshooting NVMe/RoCEv2 requires the knowledge of Infiniband TH protocol
- NVMe/RoCEv2 cannot be used for long distances (NVMe/TCP is better choice)

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

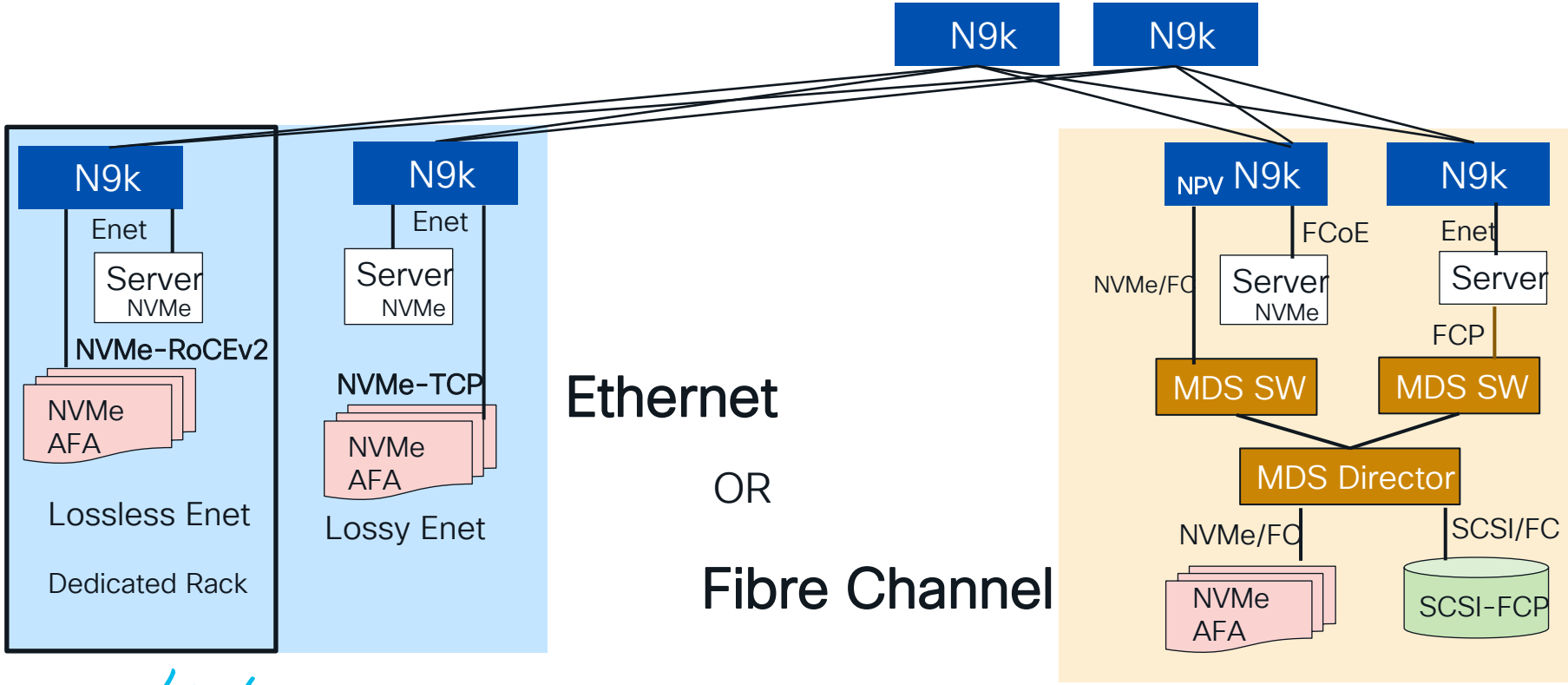
3-NVMe Transport Options (FC, TCP, RoCEv2)

4-NVMe Datacenter Design

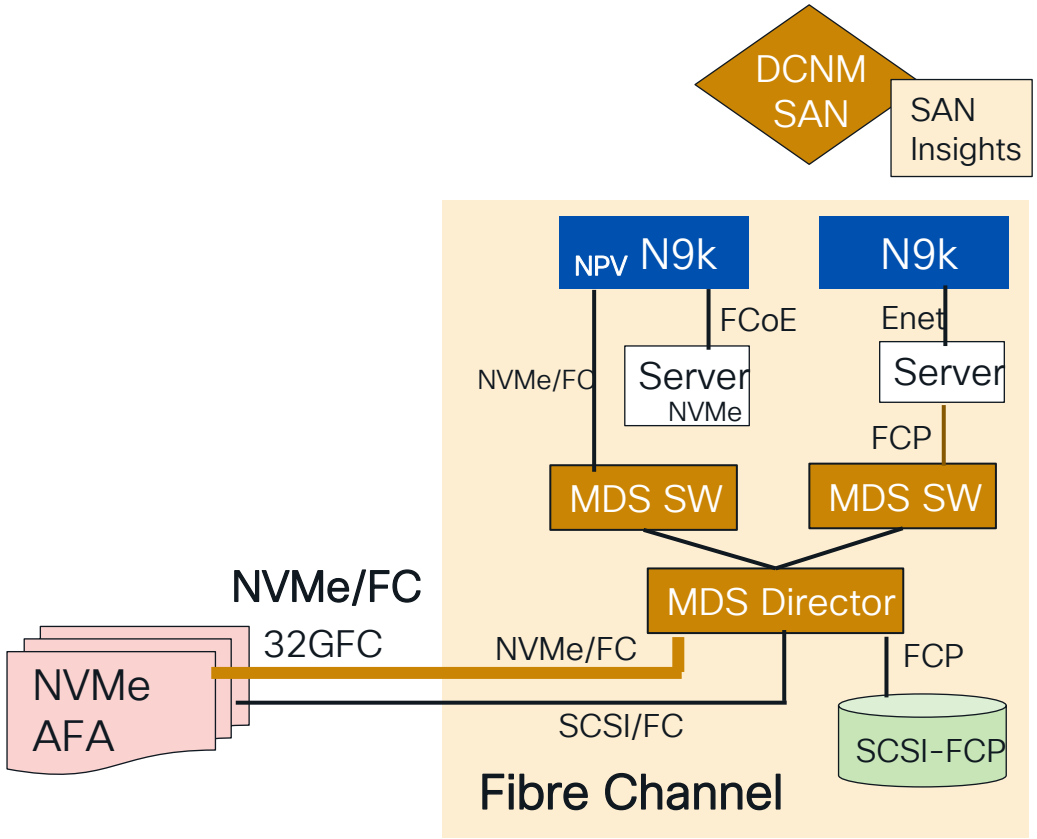
5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

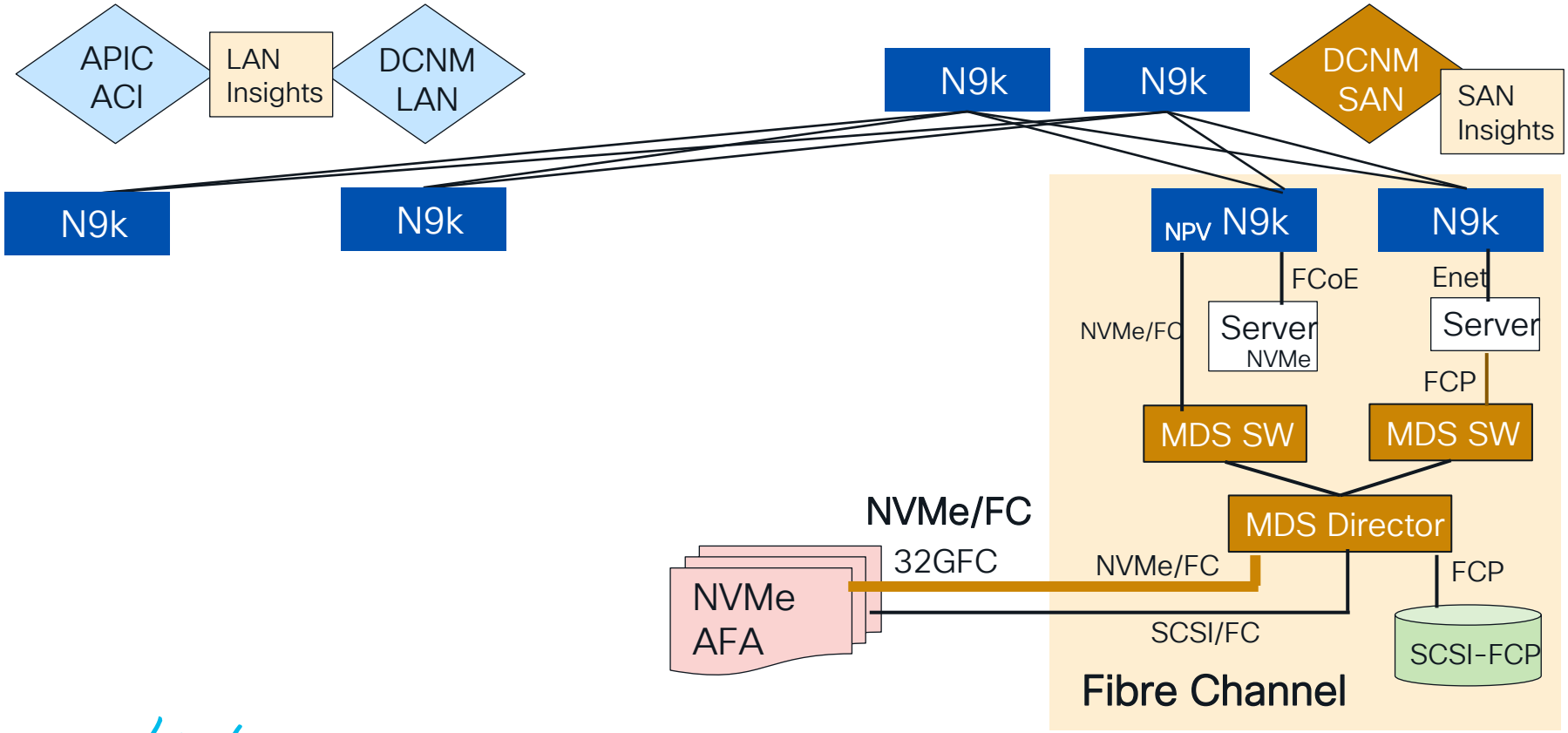
Return on Infrastructure Investments for Storage



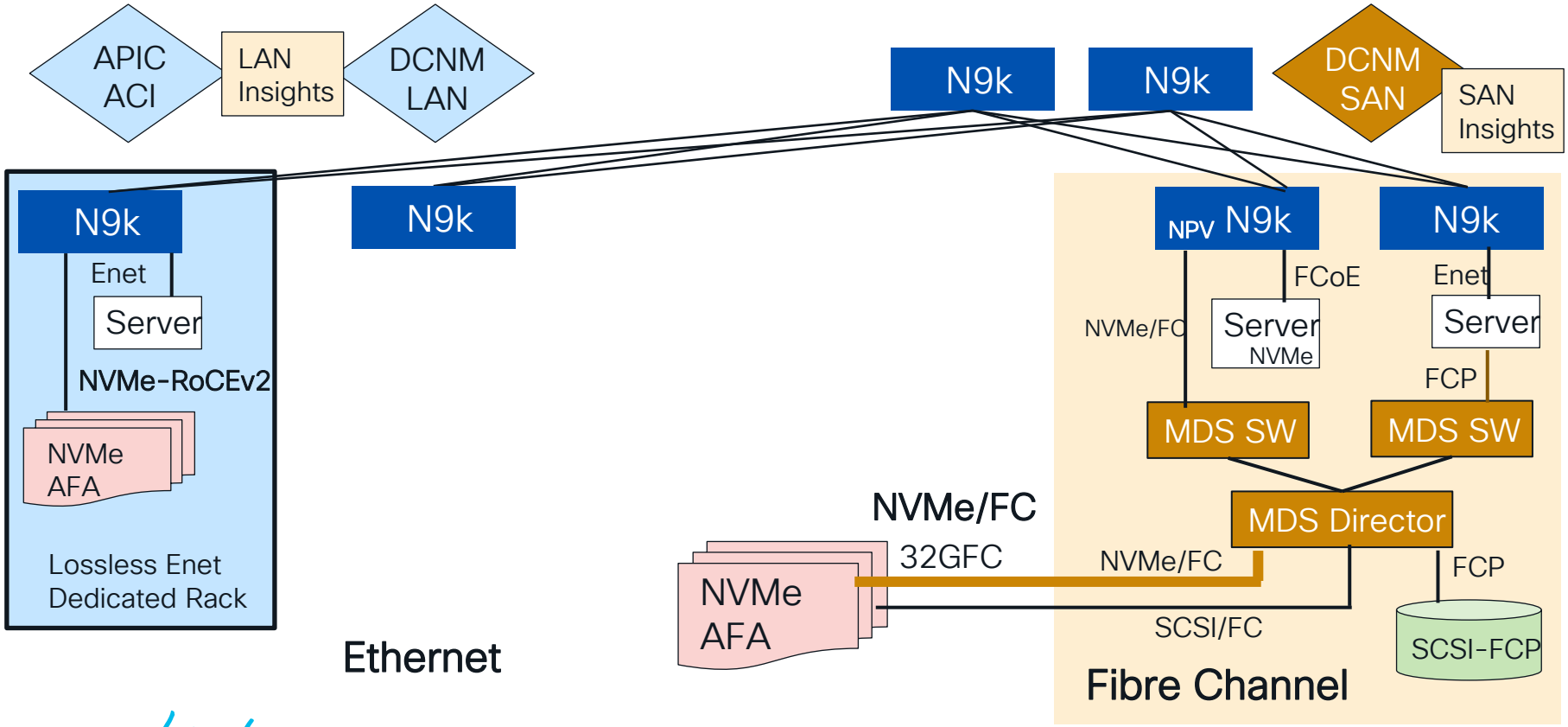
Cisco NVMe-Anywhere (Fibre Channel / Ethernet)



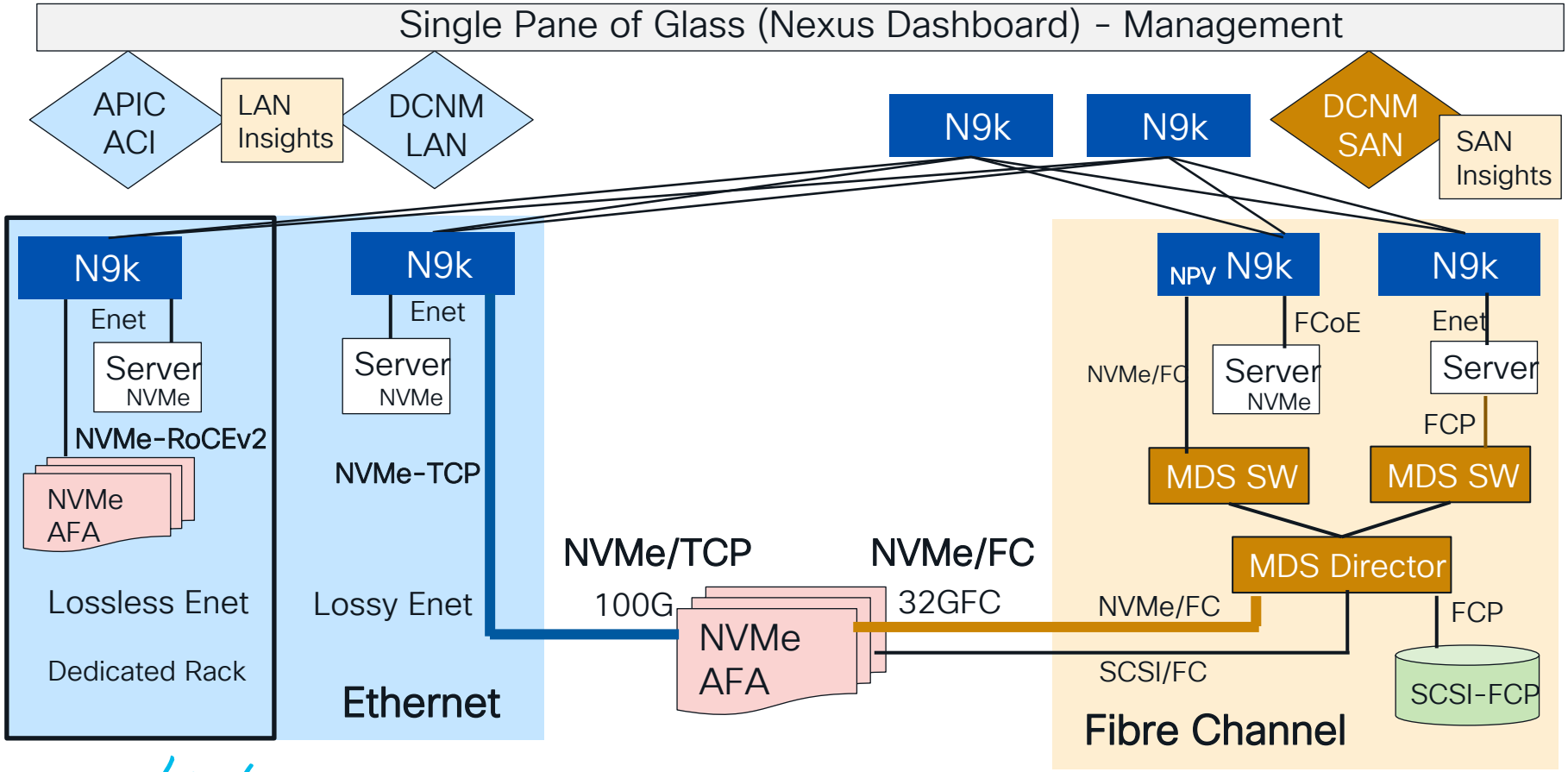
Cisco NVMe-Anywhere (Fibre Channel / Ethernet)



Cisco NVMe-Anywhere (Fibre Channel / Ethernet)



Cisco NVMe-Anywhere (Fibre Channel / Ethernet)



Best Practices (Do's & Don'ts)

- Keep your mission critical applications on Fibre Channel
- On FC fabric Start migrating towards NVMe/FC (check the VMware support)
- For certain workloads use NVMe/TCP lossy on the smaller scale (without CDC)
- NVMe-RoCEv2 traffic should be confined to the Rack level (below TOR switch)
- Cisco Nexus Dashboard will be key to manage the hybrid fabric (Enet/FC) with a single pane of glass
- Use NVMe ANA feature for Multipathing, sharing of the same namespace for NVMe/TCP and NVMe/FC hosts

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

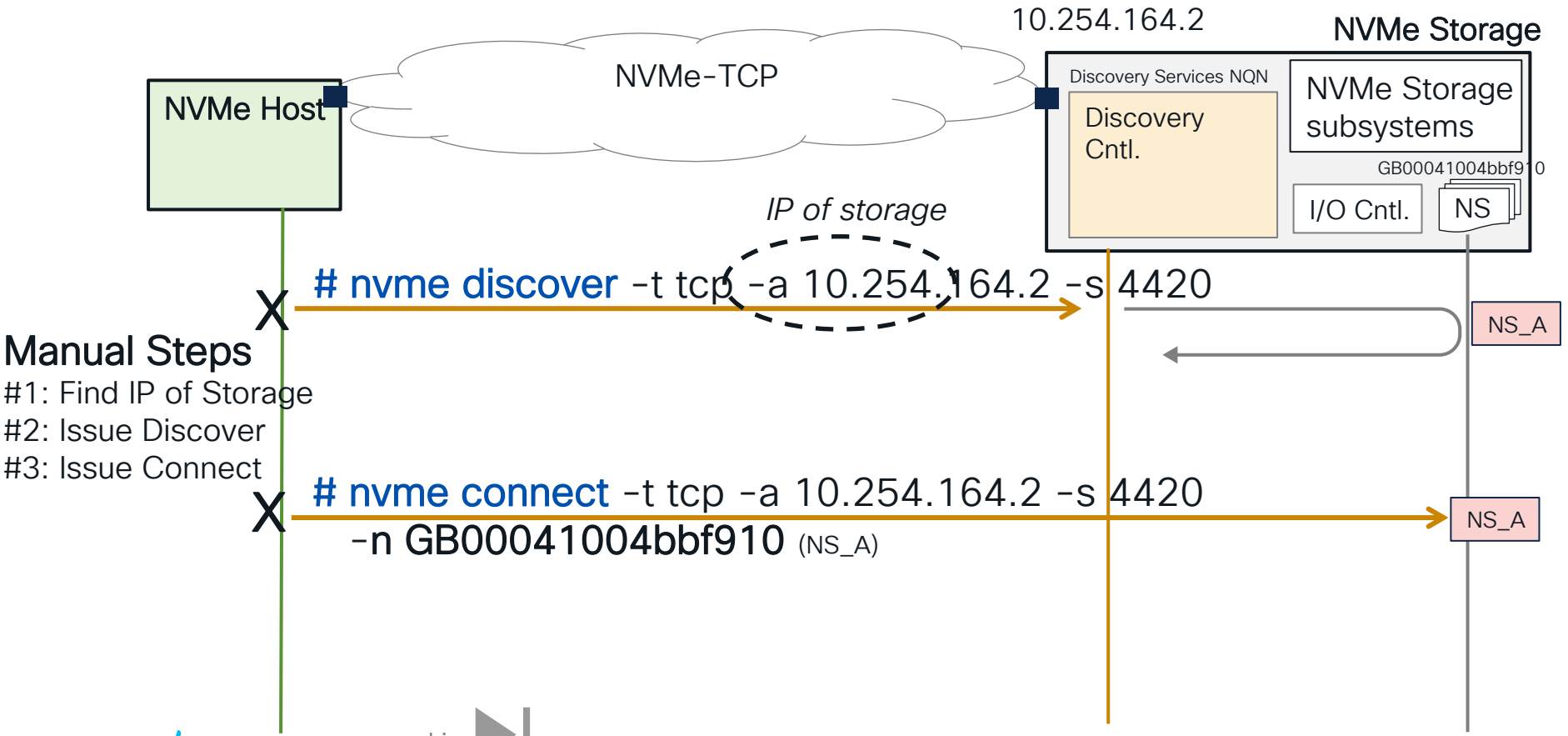
3-NVMe Transport Options (FC, TCP, RoCEv2)

4-NVMe Datacenter Design

5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

1-NVMe CDC (Problem: How to Discover Storage Resources automatically?)



skip

1-NVMe CDC (Bonjour)

mDNS (rfc 6762)
DNS-SD (rfc 6763)

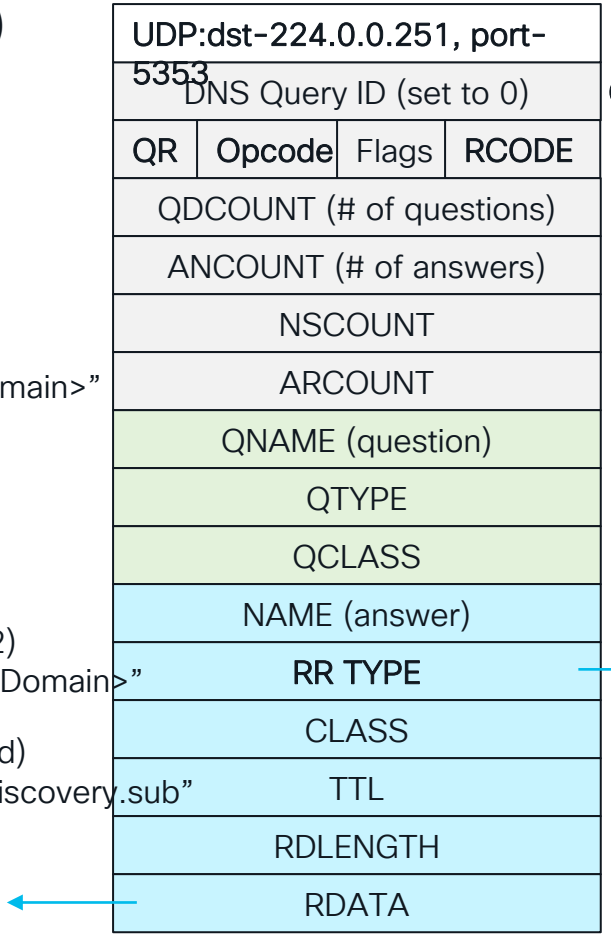
Type(12): DNS-PTR (Pointer Record)
"[<service name>].<protocol>.<Domain>"
"[_<subtype>._sub._nvme-disc].<protocol>.<domain>"

_nvme-disc.tcp.local
_cdc._sub._nvme-disc.tcp.local
_ddcpull._sub._nvme-disc.tcp.local

Type(33): DNS-SRV (Service Record-rfc2782)
"<Instance-Name>.<service name>.<protocol>.<Domain>"

Type(16): DNS-TXT additional info (K/V record)
"<length byte>p=tcp<length byte>nqn=NQN.of.Discovery.sub"

Type(1): A record
IP Address



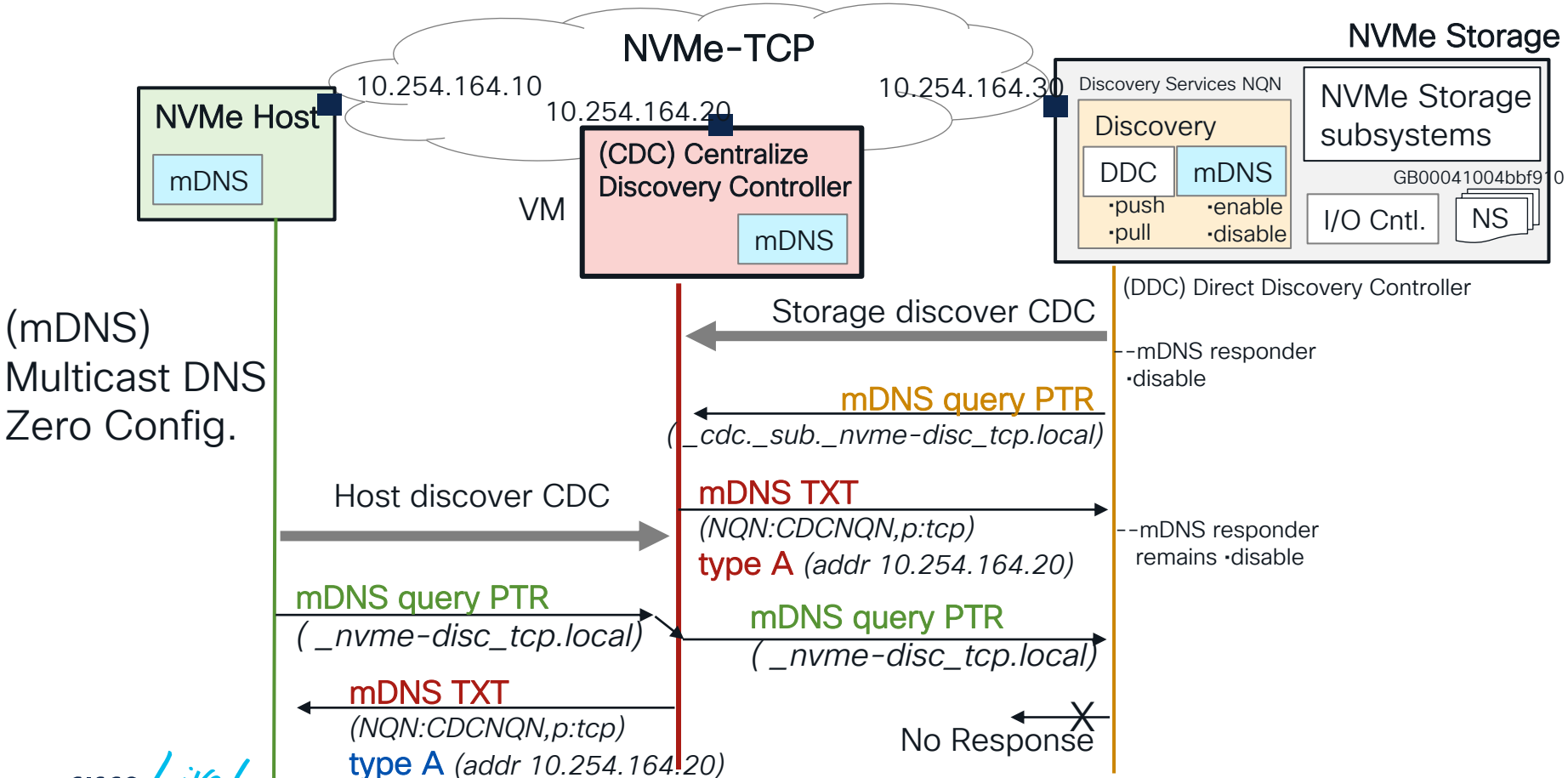
QR: 0 query
1 response

Opcode:
0-query (mDNS
1-Iquery
2-status
4-notify
5-update
6-DSO

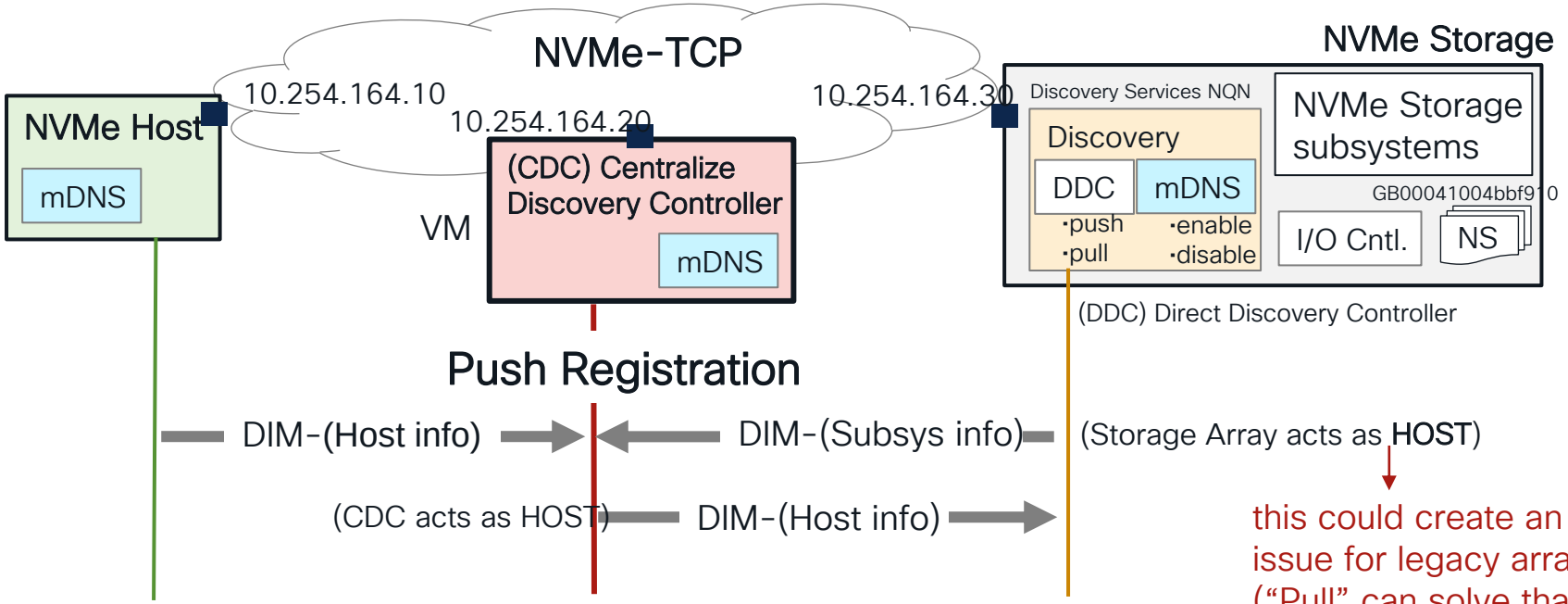
RCODE:(Response)
0 (mDNS)
1 format error
2 server failure
3 name error
4 not implemented
5 refused

RR: Resource Record
RR TYPE codes
01 -Address record
12 -PTR pointer record
16 -TXT text record
33 -SRV service record

1-NVMe CDC (Centralized & Direct Discovery Controller)



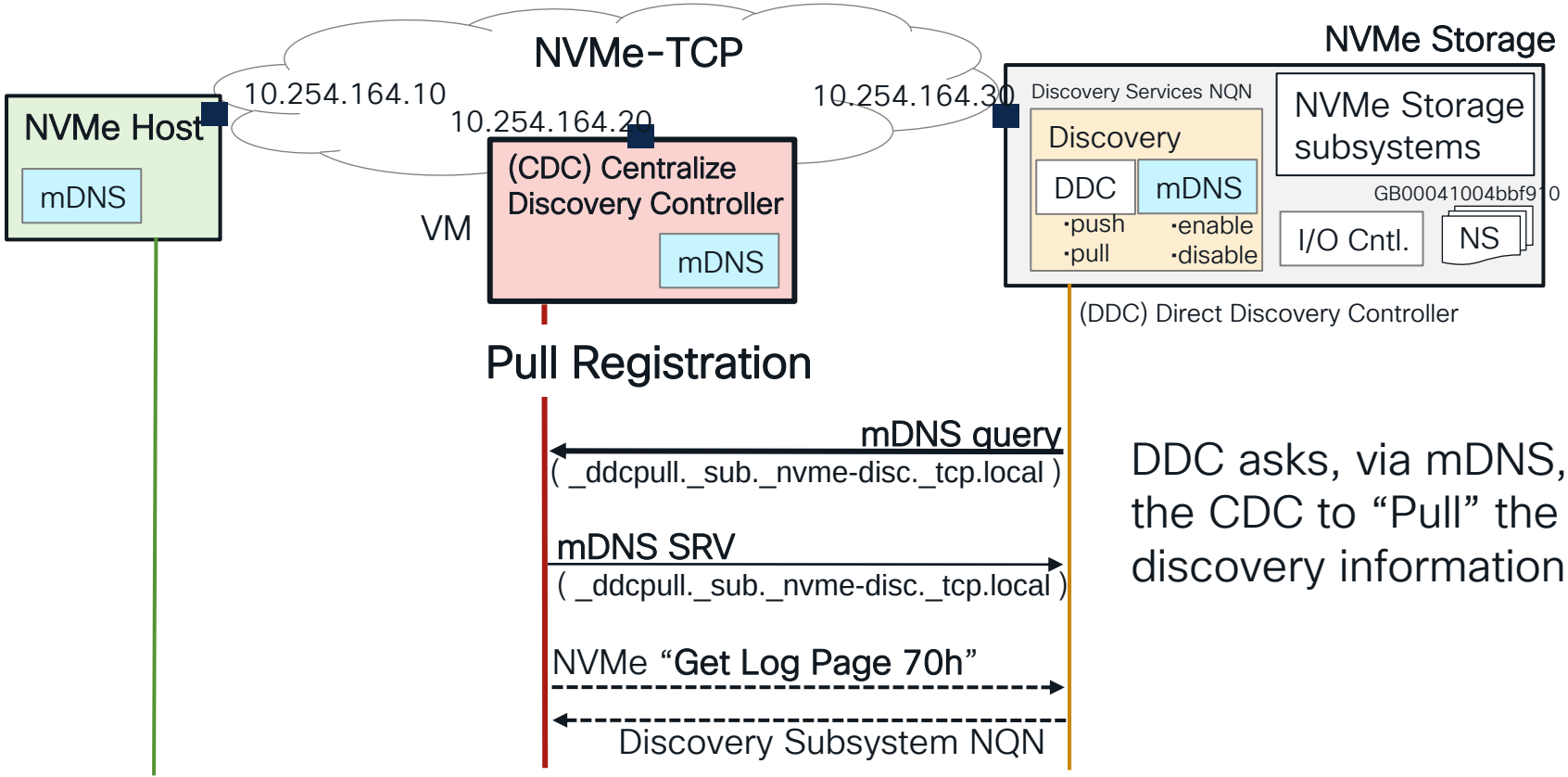
1-NVMe CDC (Discovery Information Mgmt. -PUSH)



Discovery Information Mgmt. (DIM) *"NVMe ADMIN CMD (21h)"*

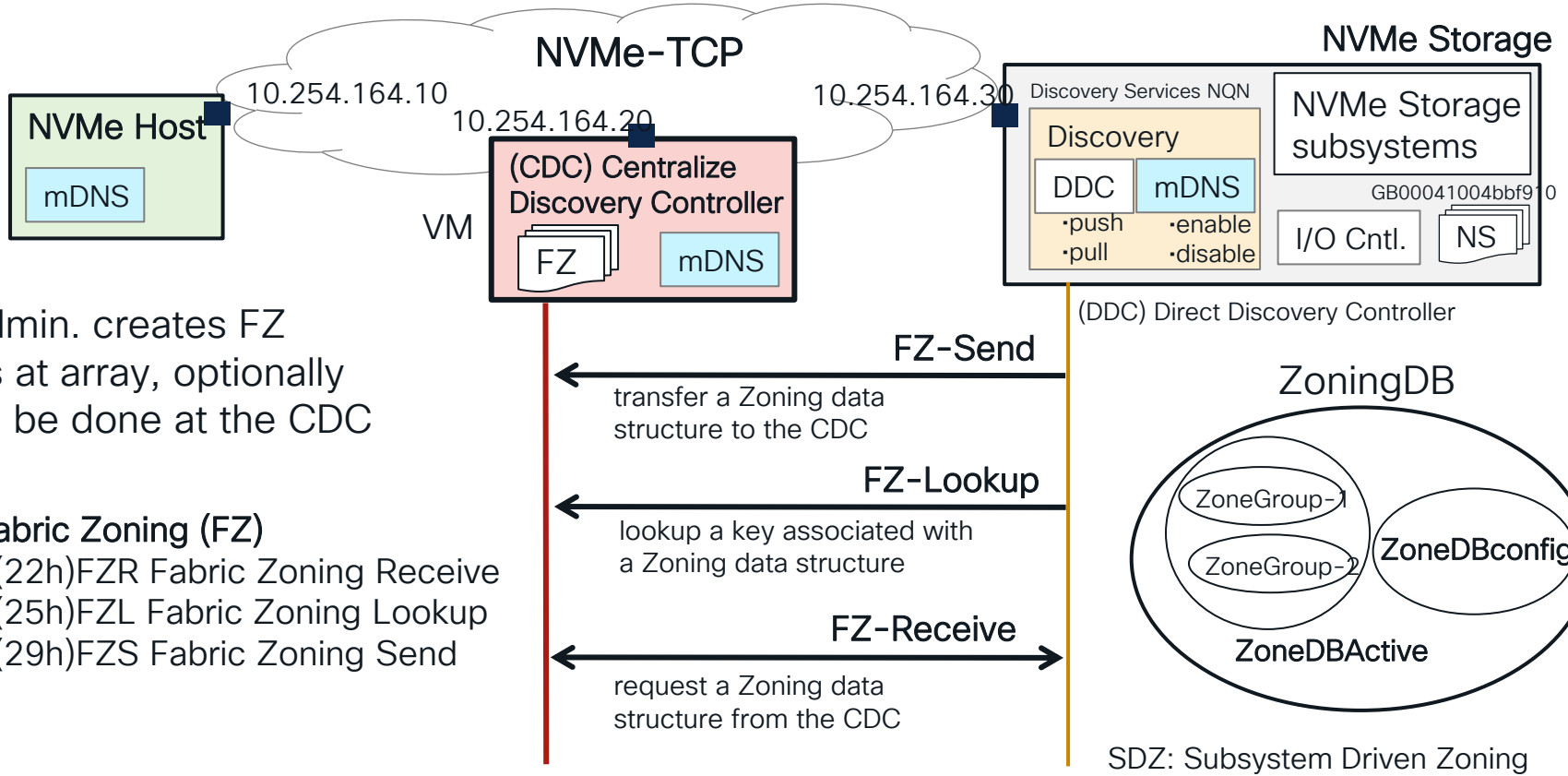
Task Field: (0h) Register, (1h) Unregister, (2h) Update
Entry Type: (1h) Host is pushing, (2h) DDC is pushing, (3h) CDC is pushing
Entry Format: (1h) Basic Discovery Info.(subsystem), (2h) Extended Discovery Info. (includes

1-NVMe CDC (Discovery Information Mgmt. -PULL)



DDC asks, via mDNS, the CDC to "Pull" the discovery information

1-NVMe CDC (Fabric Zoning)

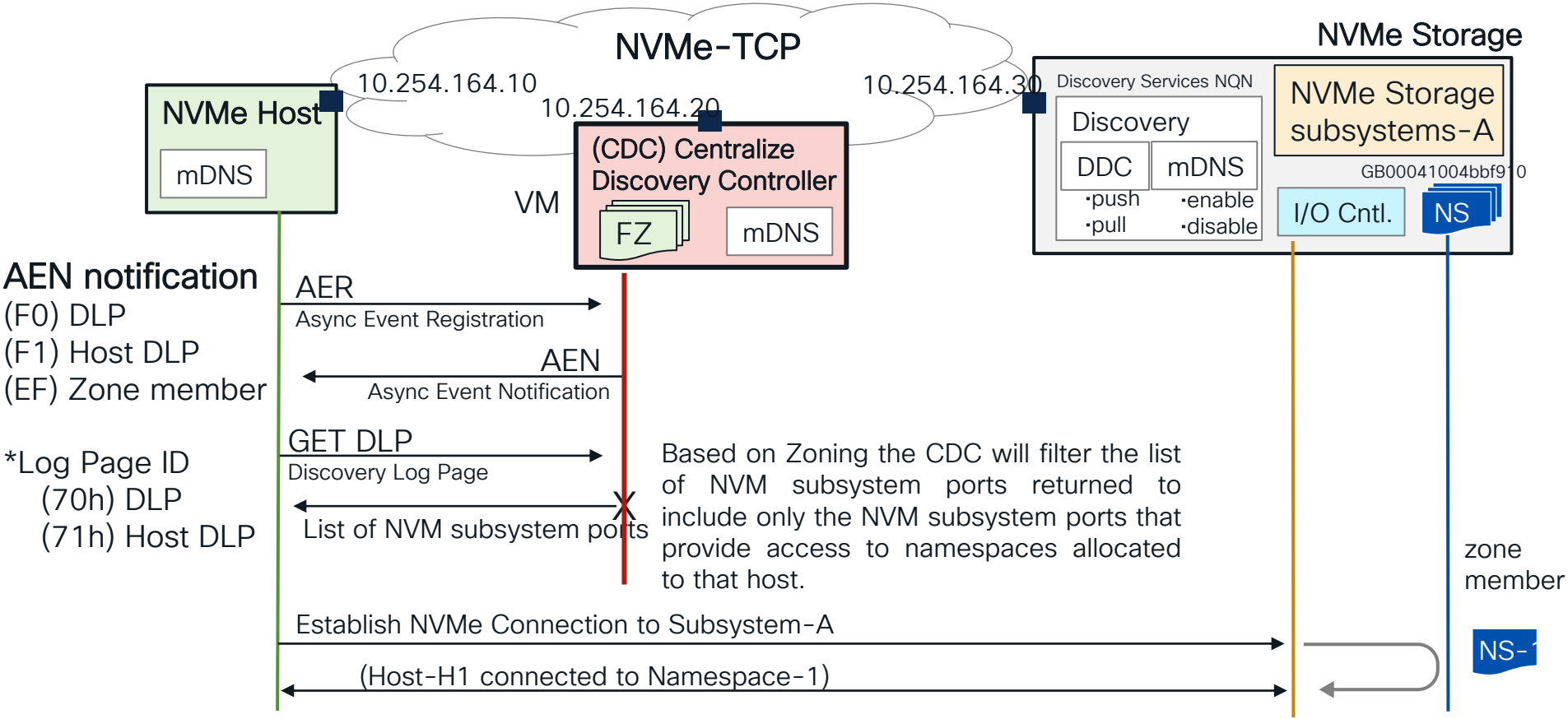


Sys Admin. creates FZ groups at array, optionally FZ can be done at the CDC also.

Fabric Zoning (FZ)
- (22h)FZR Fabric Zoning Receive
- (25h)FZL Fabric Zoning Lookup
- (29h)FZS Fabric Zoning Send

SDZ: Subsystem Driven Zoning
A.K.A = TDZ: Target Driven Zoning

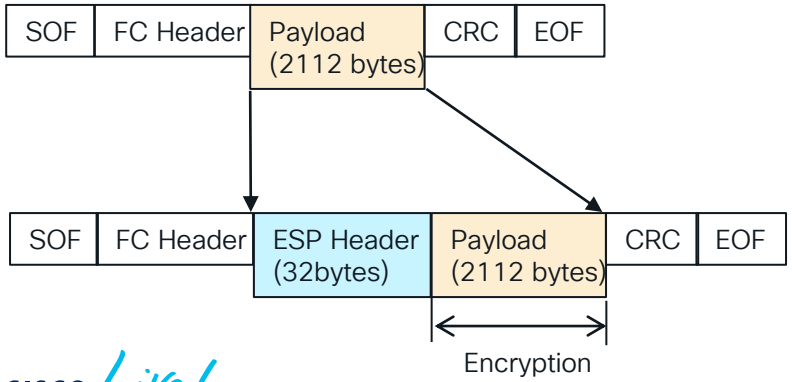
1-NVMe CDC (Async. Event Registration/Notification)



2-NVMe-TCP (TLS Security)

NVMe/FC

FC-SP2 provides a security framework which includes authentication (using Diffie-Hellman Challenge Handshake Authentication Protocol (DHCHAP) or IKEv2), cryptographically secure key exchange, and cryptographically secure communication between Fibre Channel devices.



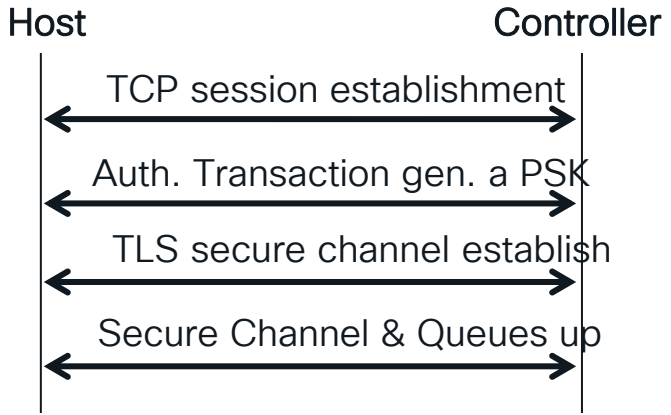
NVMe/TCP

TP 8006

Authentication: DH-HMAC-CHAP

TP8011 TLS 1.3 for NVMe/TCP

Secure channel: Authentication, Cryptographic

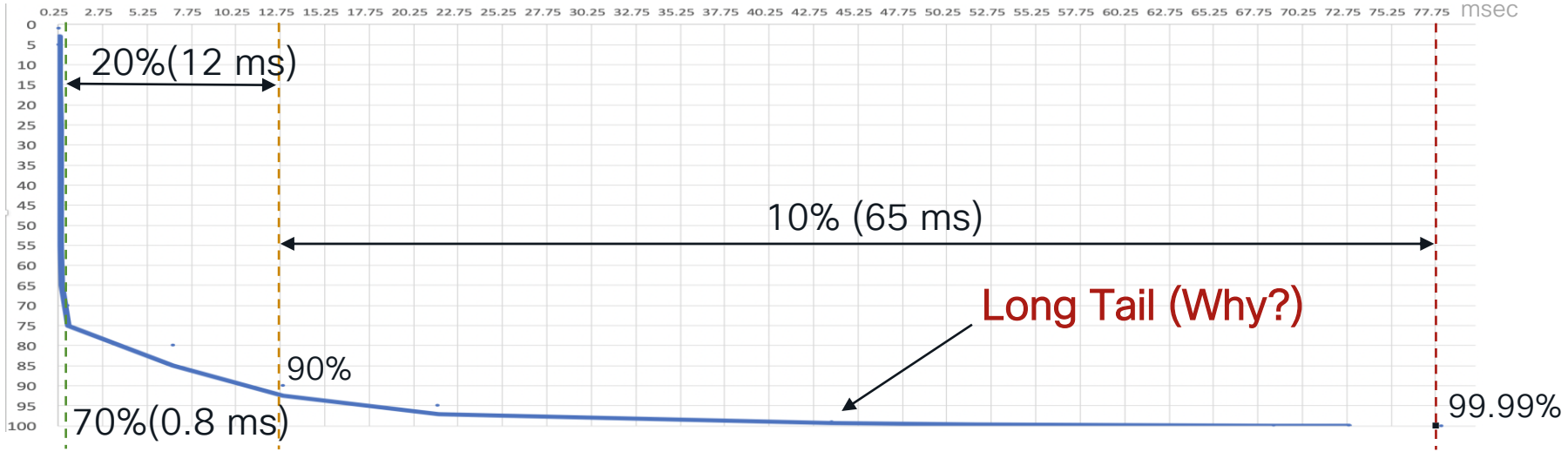


3-ZNS (How to Shorten the Long Tails)

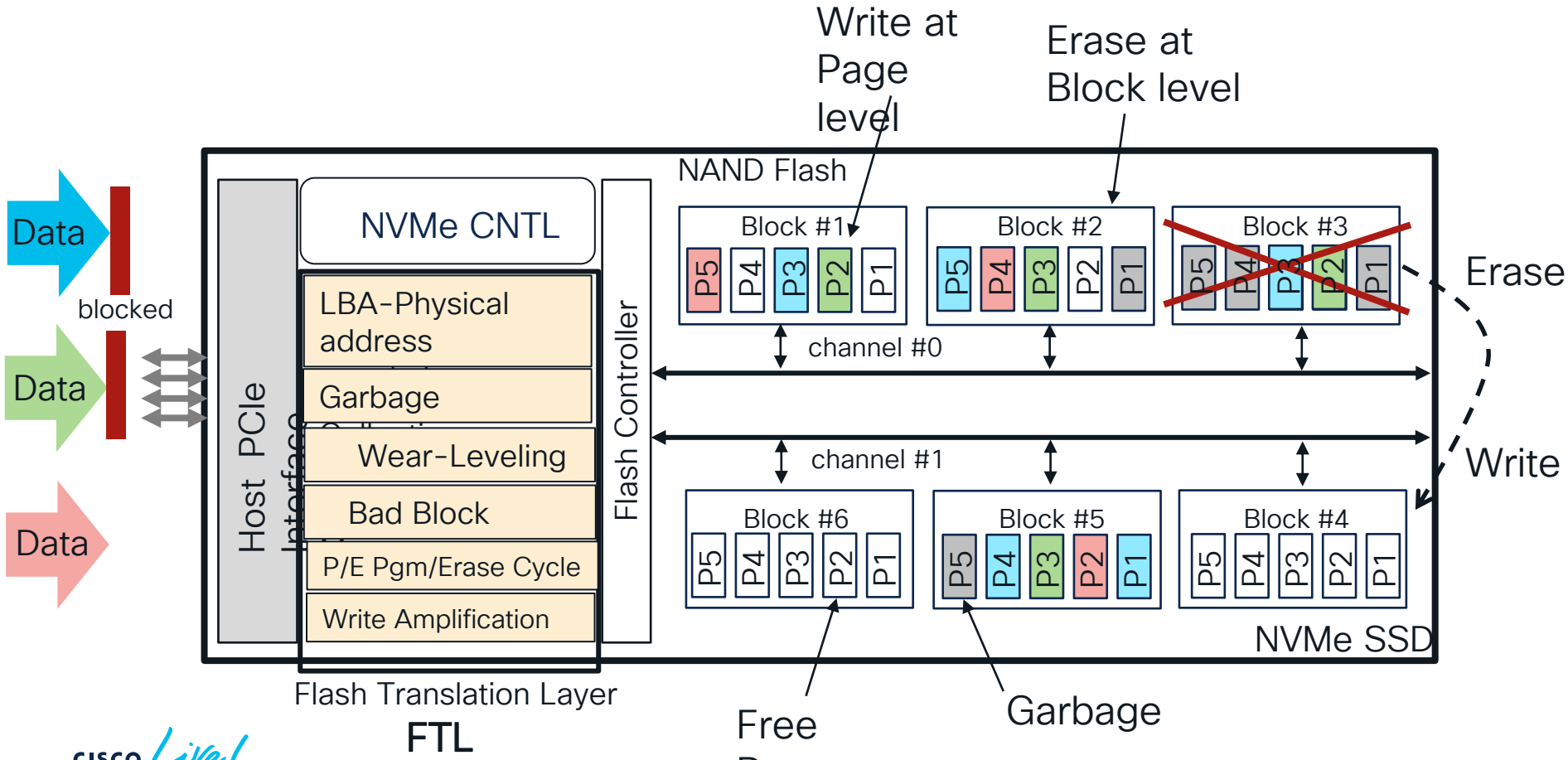
Back

I/O Completion Latency

```
clat percentiles (usec):
 |  1.00th=[  302],  5.00th=[  326], 10.00th=[  343], 20.00th=[  363],
 | 30.00th=[  392], 40.00th=[  404], 50.00th=[  416], 60.00th=[  445],
 | 70.00th=[  816], 80.00th=[ 6718], 90.00th=[12911], 95.00th=[21627],
 | 99.00th=[43779], 99.50th=[51643], 99.90th=[68682], 99.95th=[72877],
 | 99.99th=[78119]
```



3-ZNS (Flash Internals)

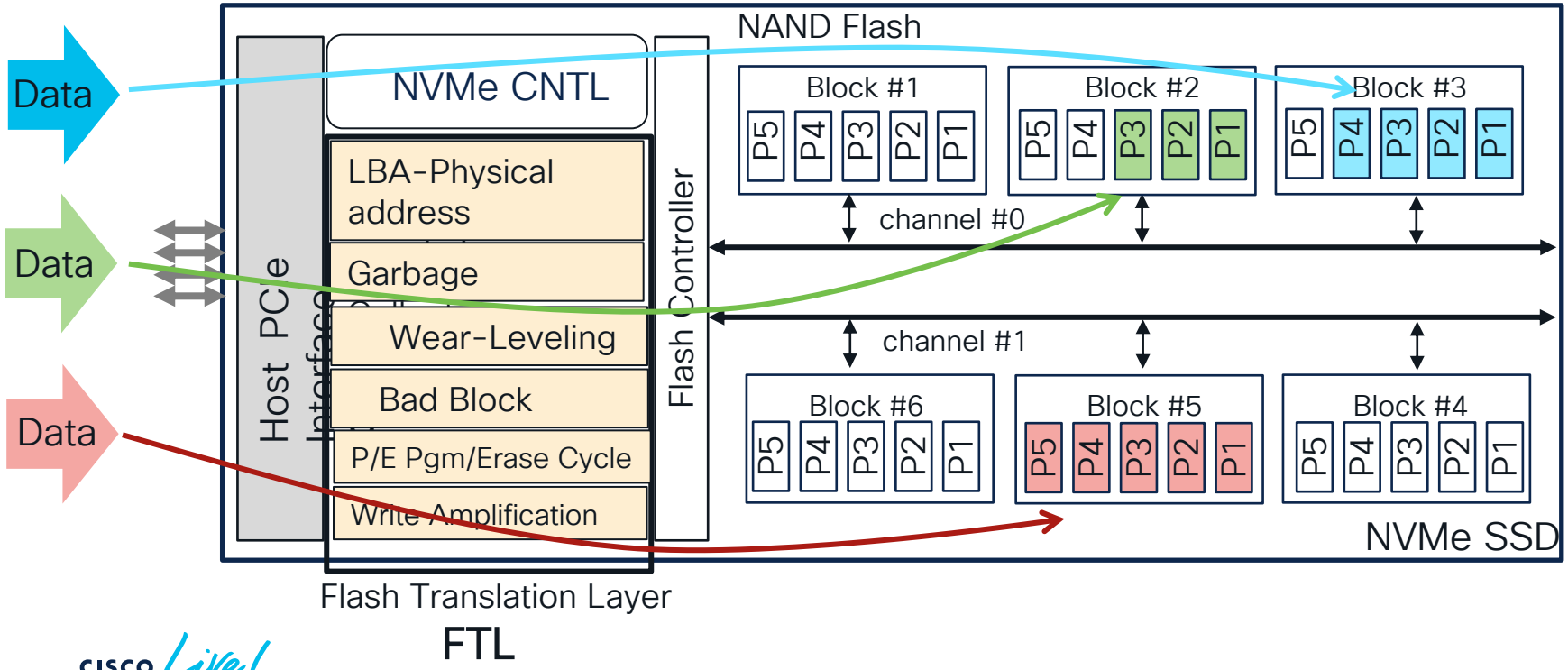


3-ZNS (Garbage collection)

I/O Determinism
- -NVMe Streams
- -NVMe Sets

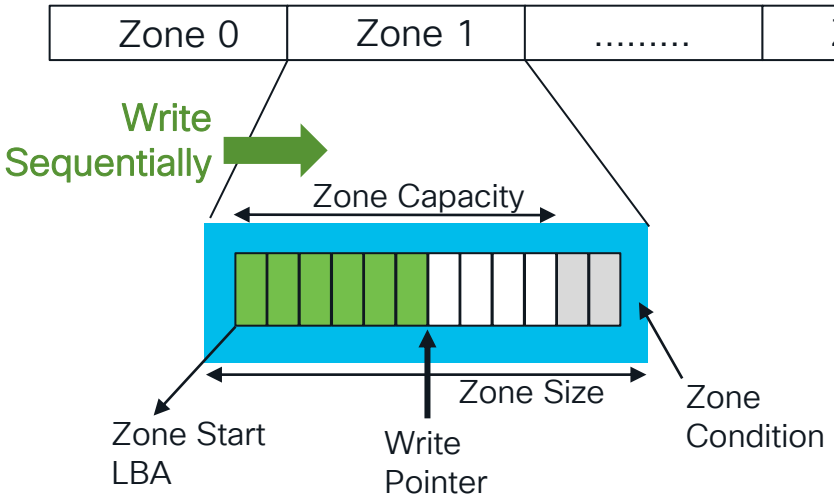
Open Channel
- -Parallel Units/Chunks
- -LightNVM

-NVMe ZNS
(Zoned Namespaces)



3-ZNS (Write Amplification)

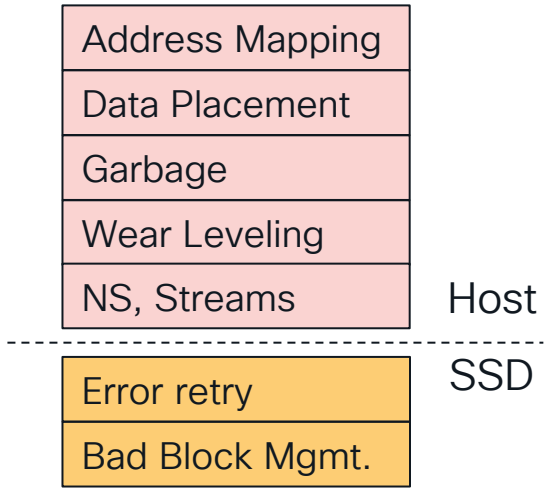
Zoned Namespaces SSD



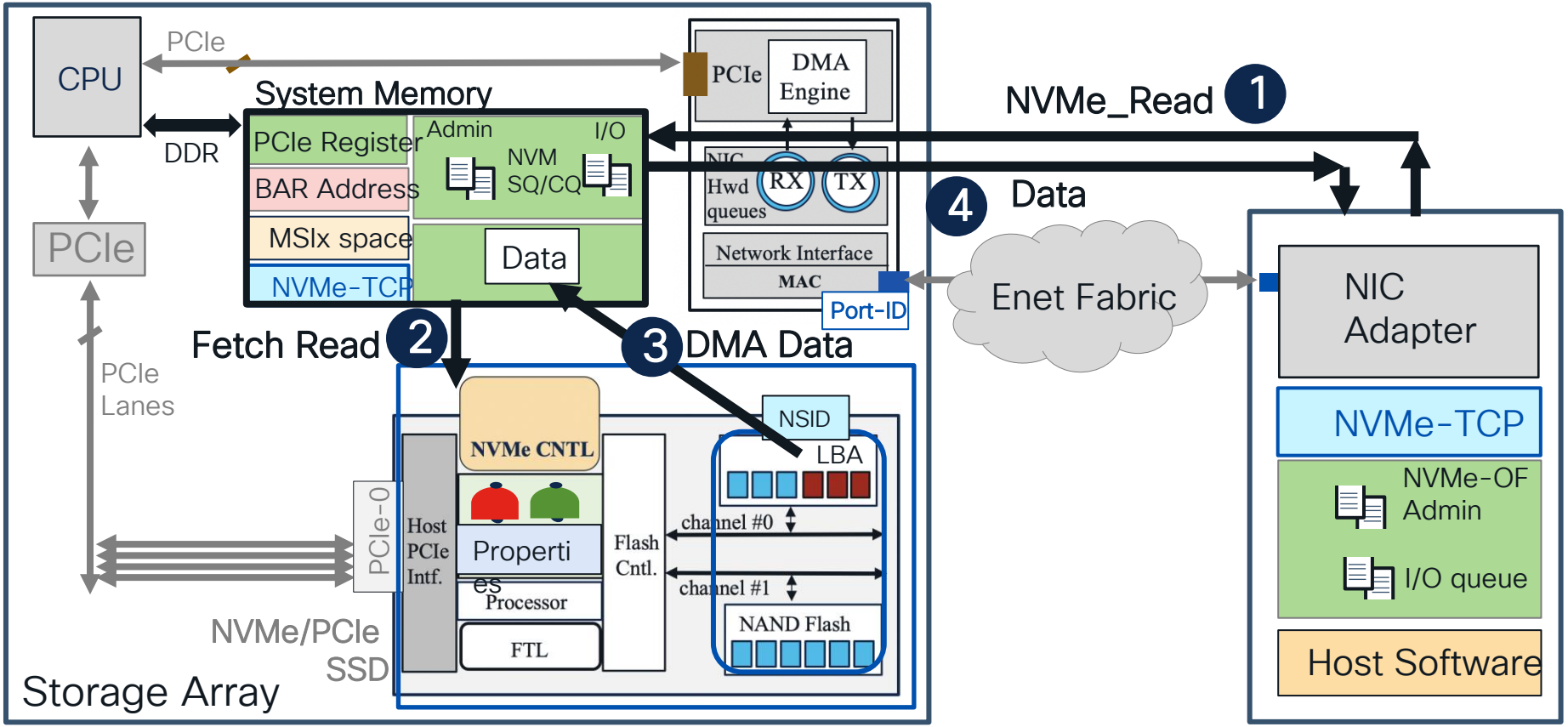
Zone Info
- -Total Zone
- -Zone Type
- -Zone State
- -Zone Attribute
- -Zone Capacity
- -Zone start LBA
- -Write Pointer

New NVMe Commands
- -Zone Mgmt. Send/Rcv
- -Zone Append
- -Zone Copy
- -Zone Commit

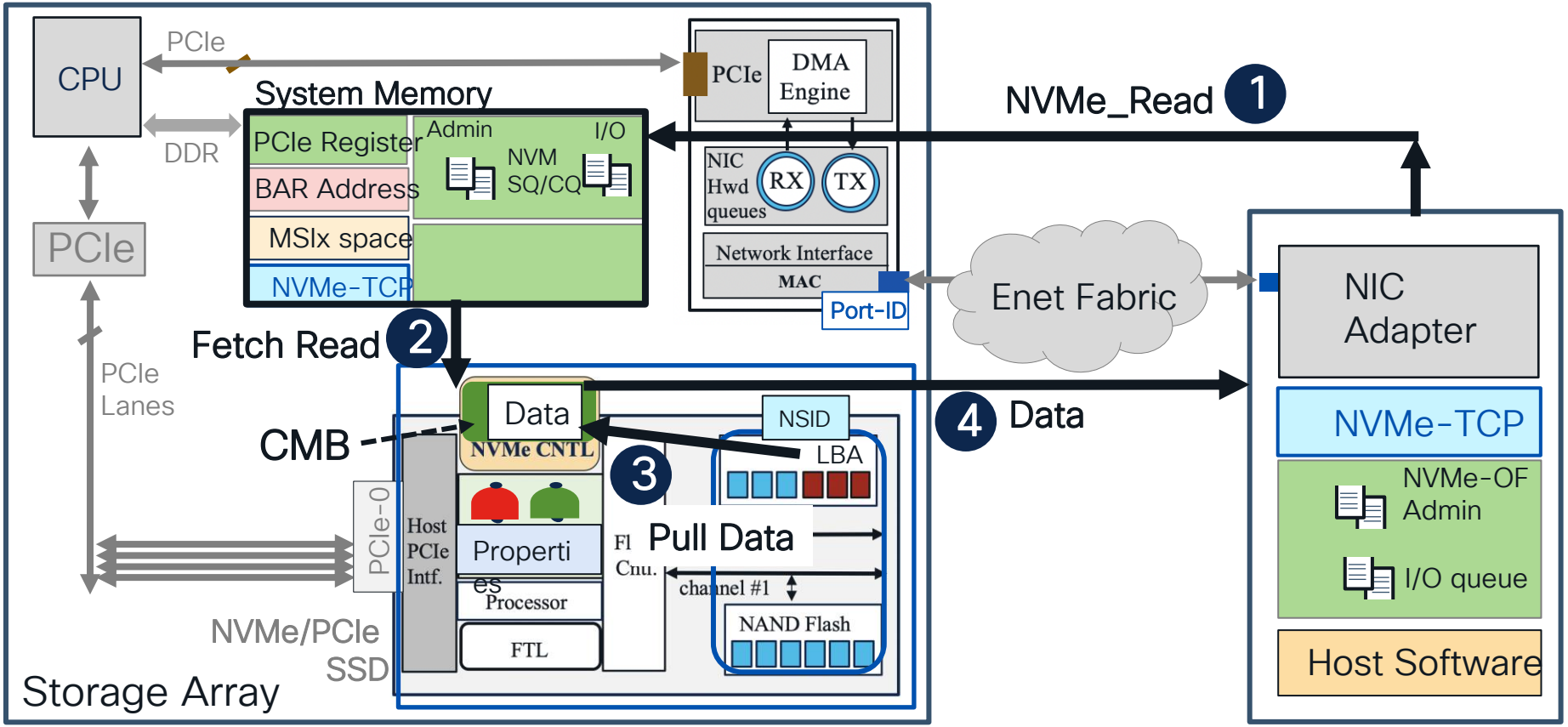
-Lower P/E cycle (increased SSD lif
-Predictable Latency



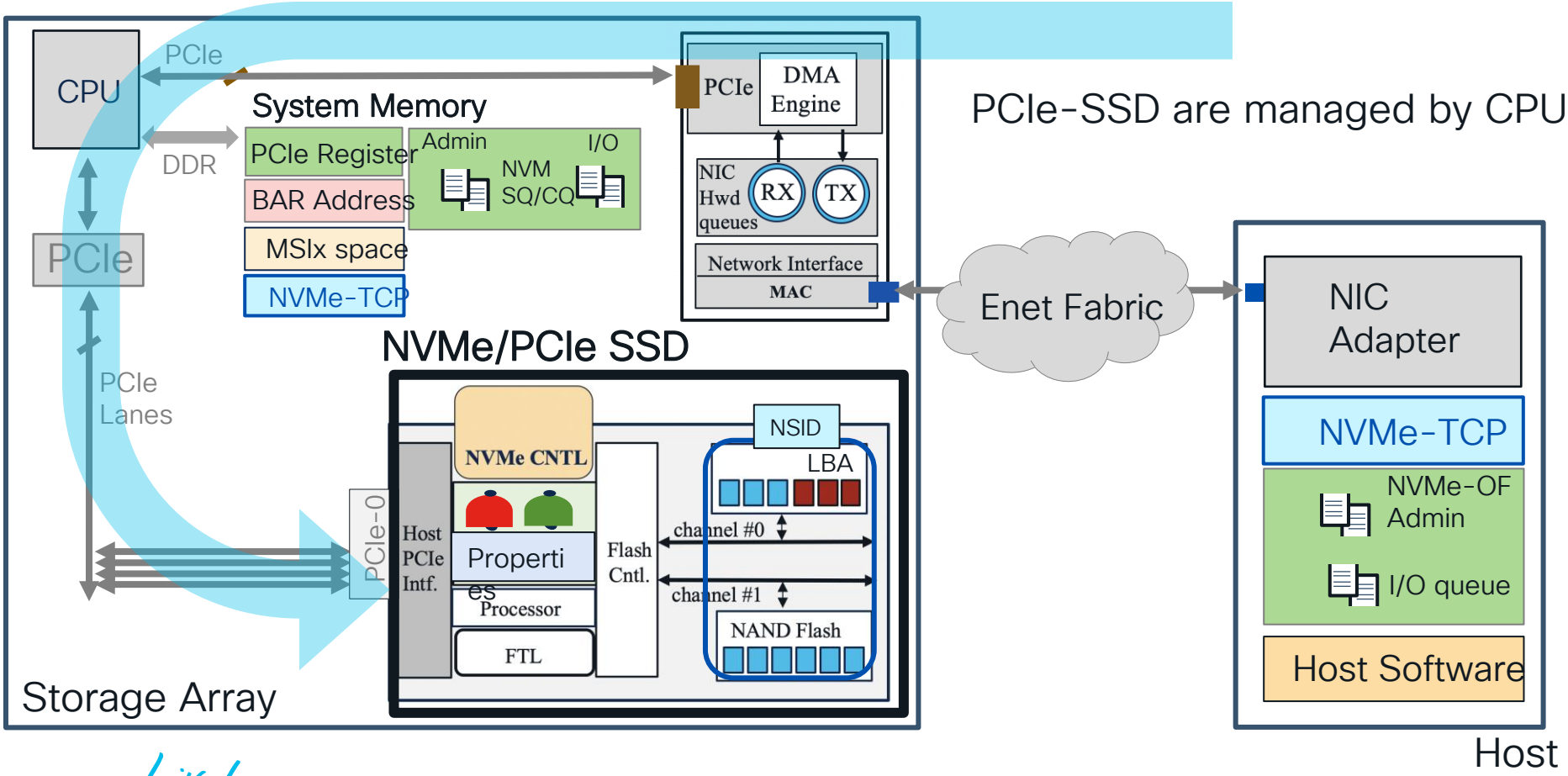
4-Controller Memory Block (CMB)



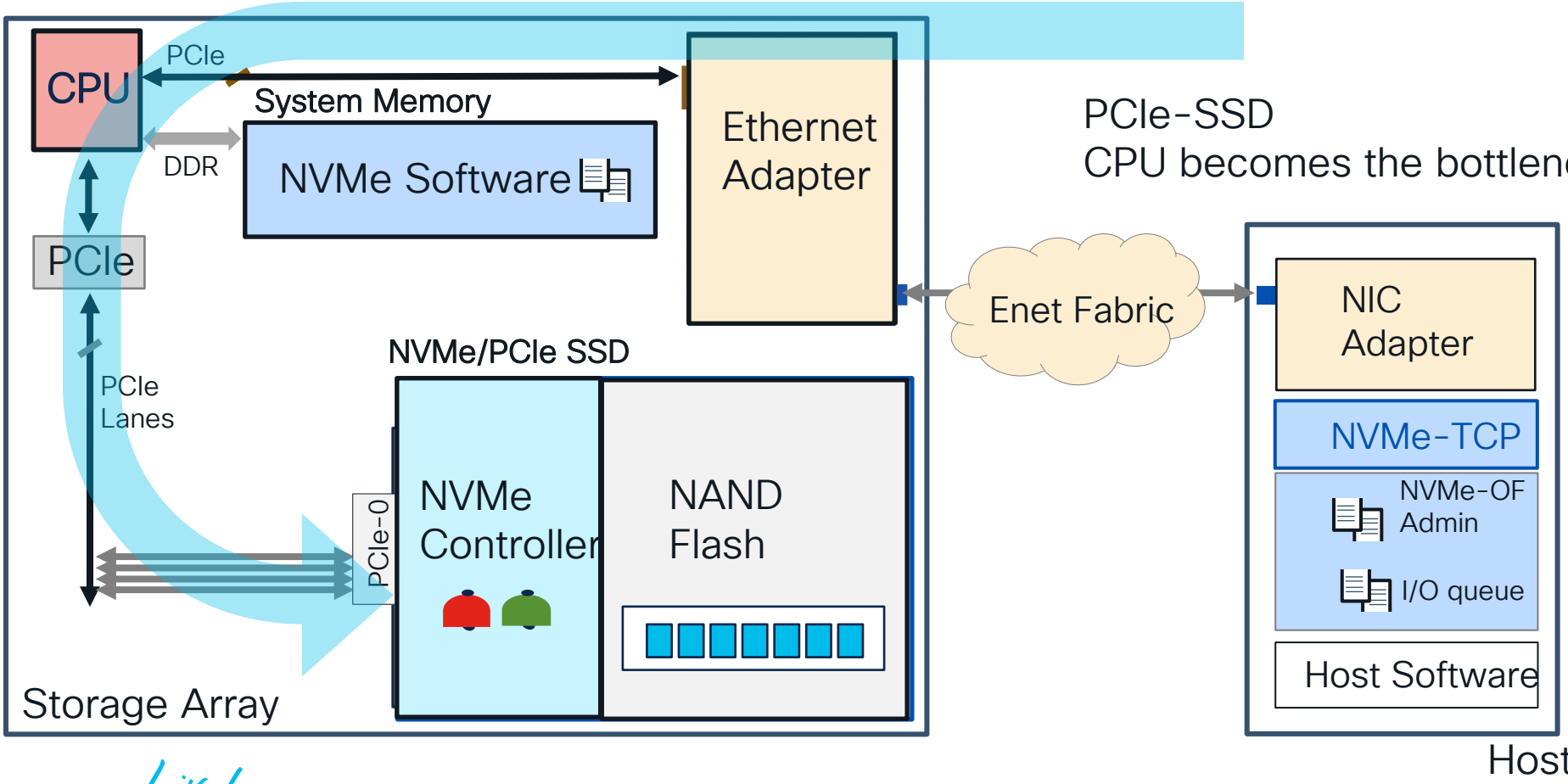
4-Controller Memory Block (CMB)



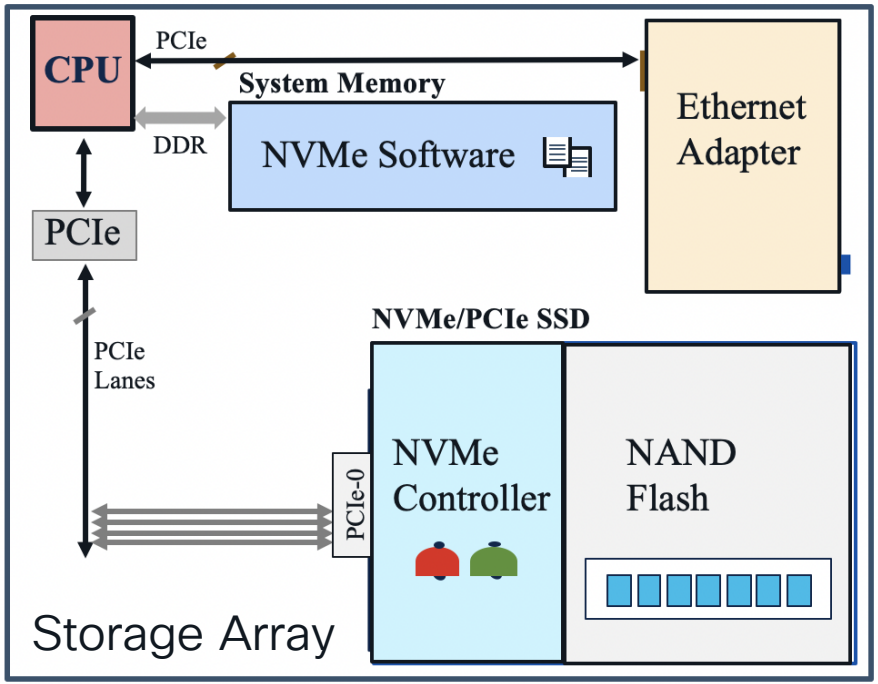
5-Ethernet-SSD

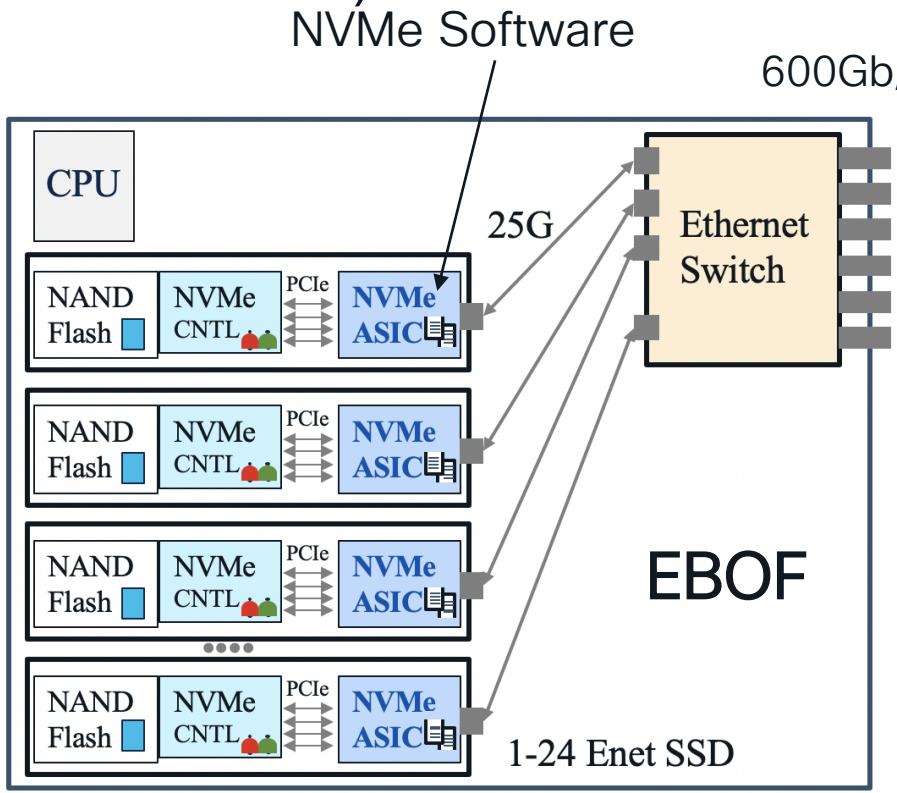


5-Ethernet-SSD

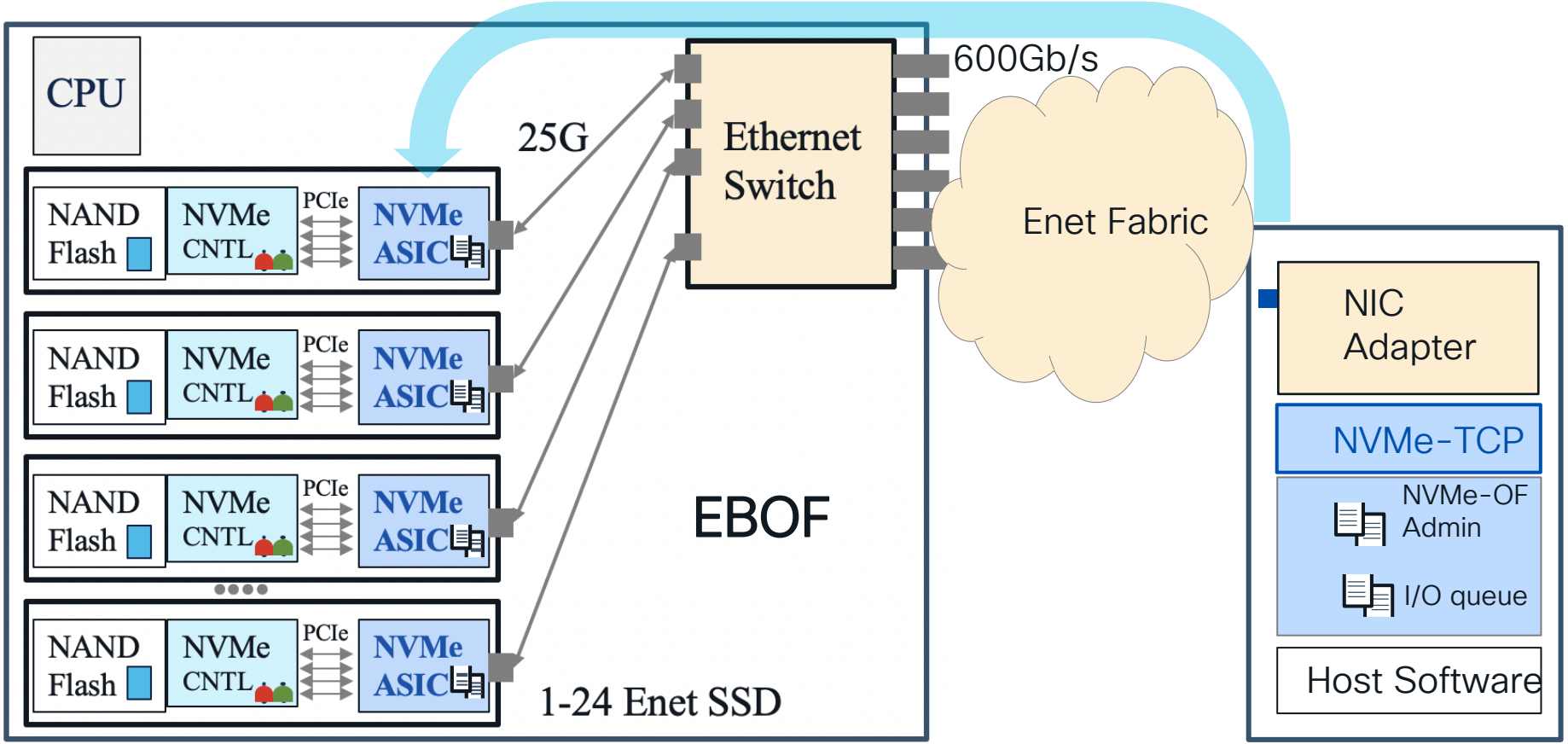


5-Ethernet-SSD (Ethernet Bunch of Flash)





5-Ethernet-SSD (CPU is offloaded)



6-NVMe Key-Value SSD

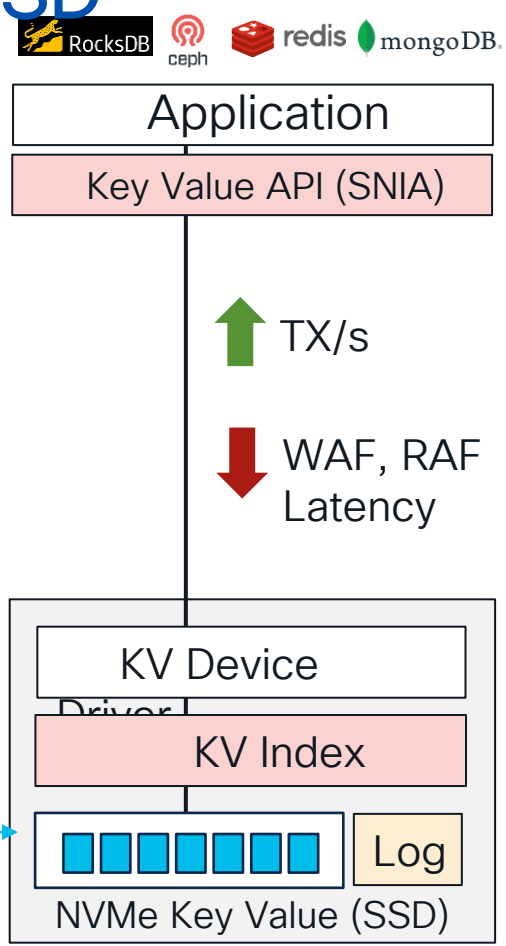
Today all storage protocols (Block, NFS or Object) uses LBA block addressing scheme.

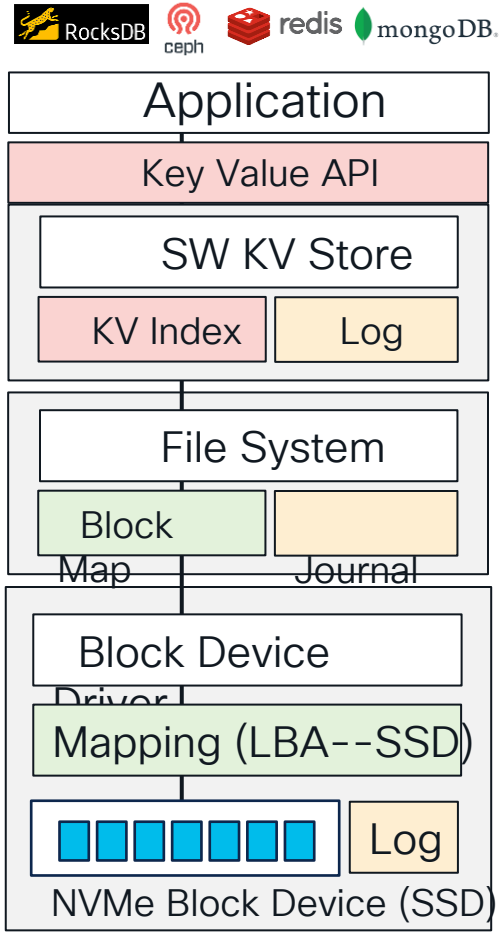
KV protocol maps an address (Key, 32 bytes max.) to a physical location where (Value, 4GB max) is storage. No LBA, hence no translation in FTL.

Key Value API (SNIA)

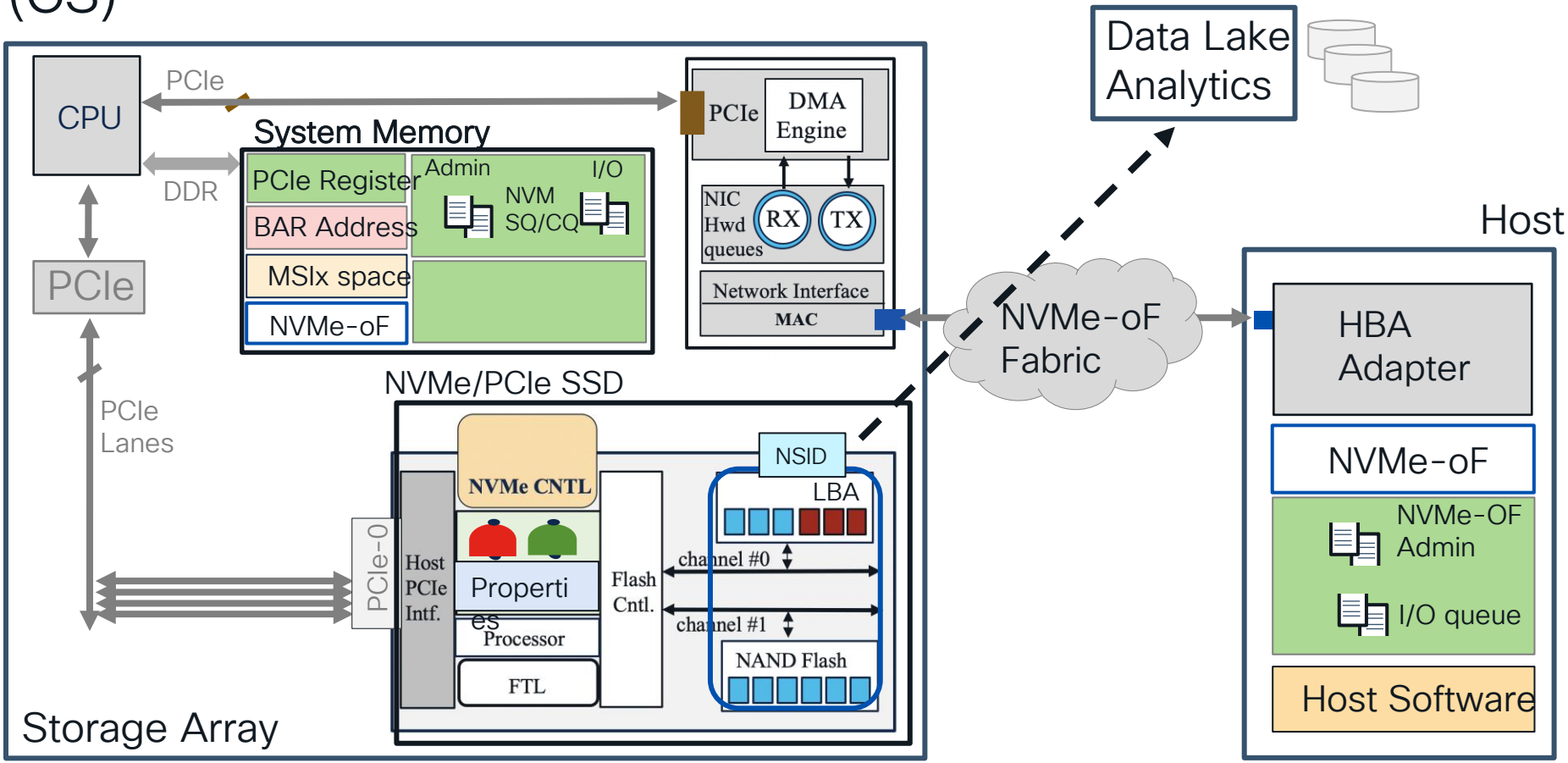
- -Open/Retrieve Device
- -Create/Delete Key Space
- -Store, Retrieve, Delete,
- -List, Delete Group

NVMe KV I/O Commands
(Store, Retrieve, List, Exist, Delete)

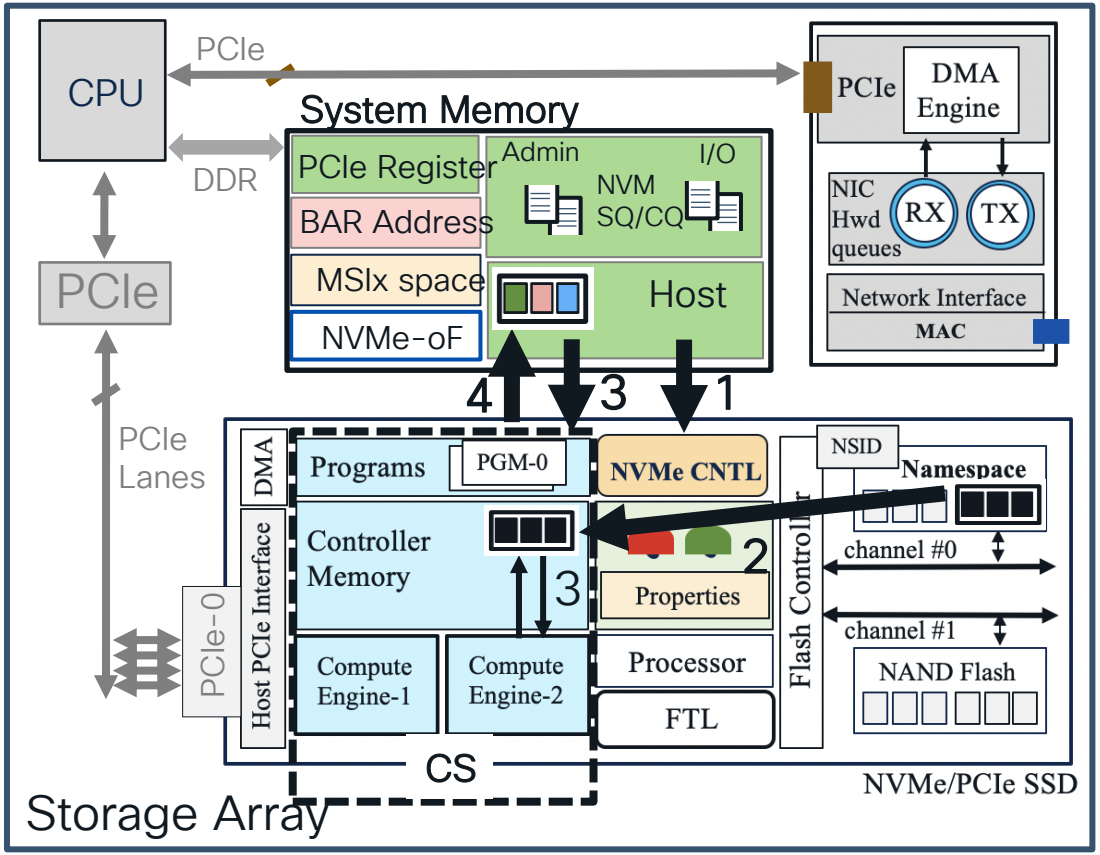




7-NVMe Computational Storage (CS)



7-NVMe Computational Storage (CS)



1 NVMe Read (NS) is issued to CN

2 CNTL moves the (NS) data to CM

3 Execute PGM-0 on CE-2

4 Read CM Output Data back to Ho

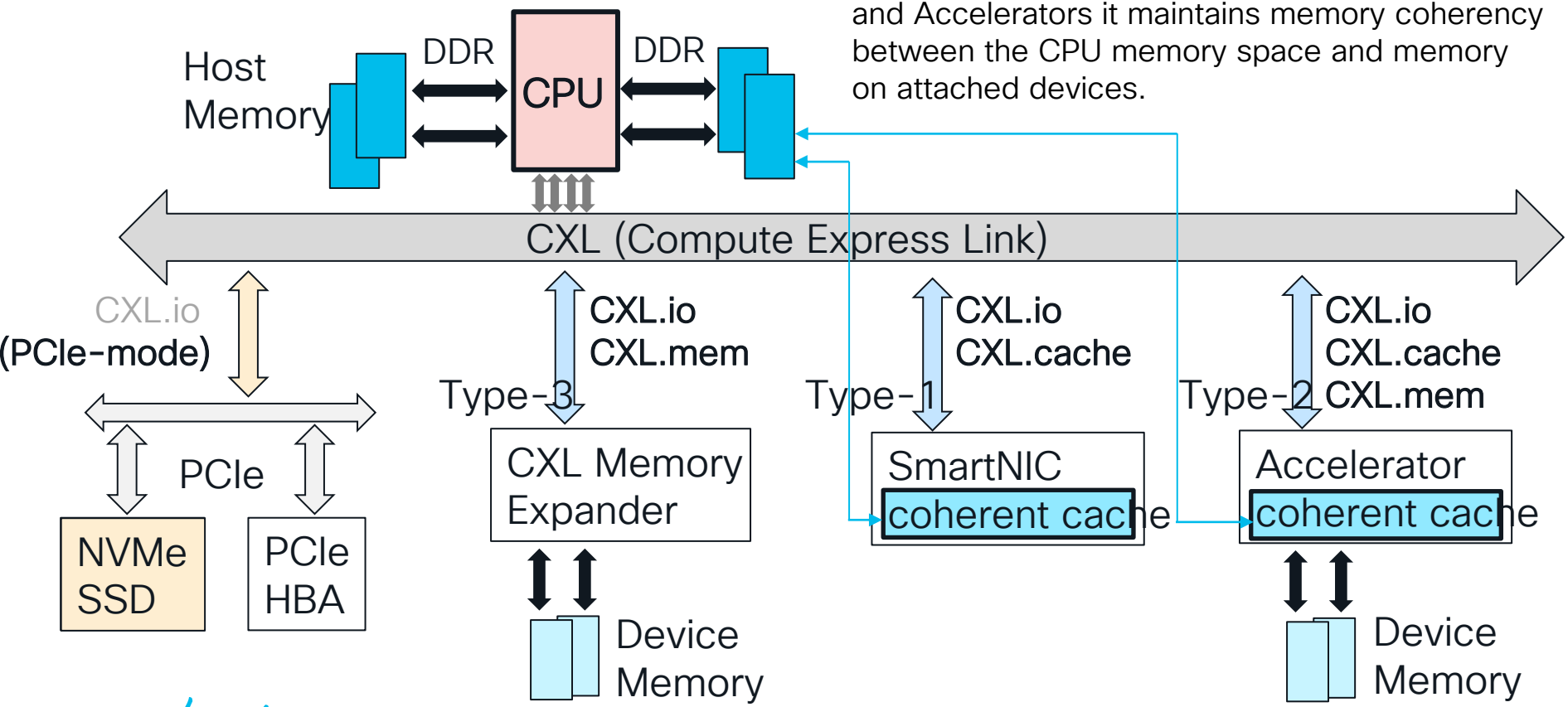
TP4091 Computational Programs

- -Execute Program
- -Load Program
- -Activate Program
- -Create/Delete Memory range set

TP4131 Controller Local Memory

8-CXL (Compute eXpress Link)

CXL is an industry-supported Cache-Coherent Interconnect for Processors, Memory Expansion and Accelerators it maintains memory coherency between the CPU memory space and memory on attached devices.



Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

3-NVMe Transport Options (FC, TCP, RoCEv2)

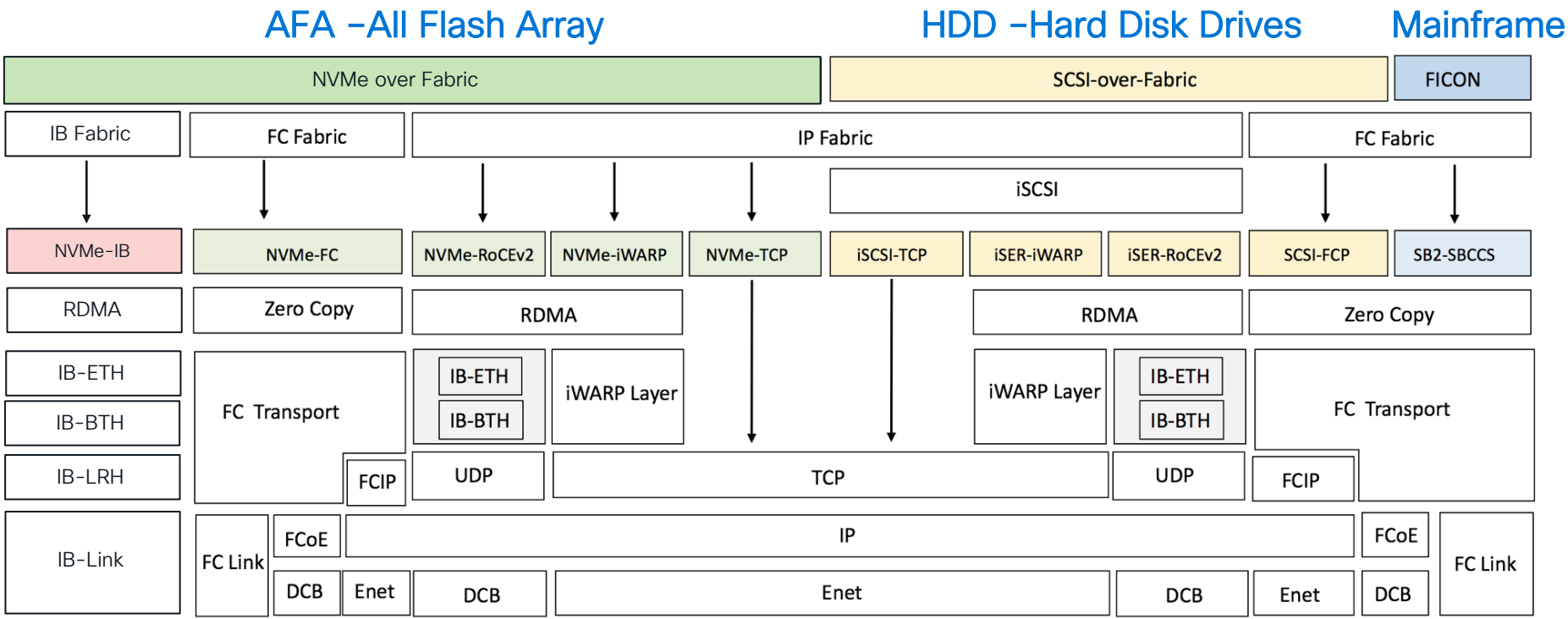
4-NVMe Datacenter Design

5-Additional Information

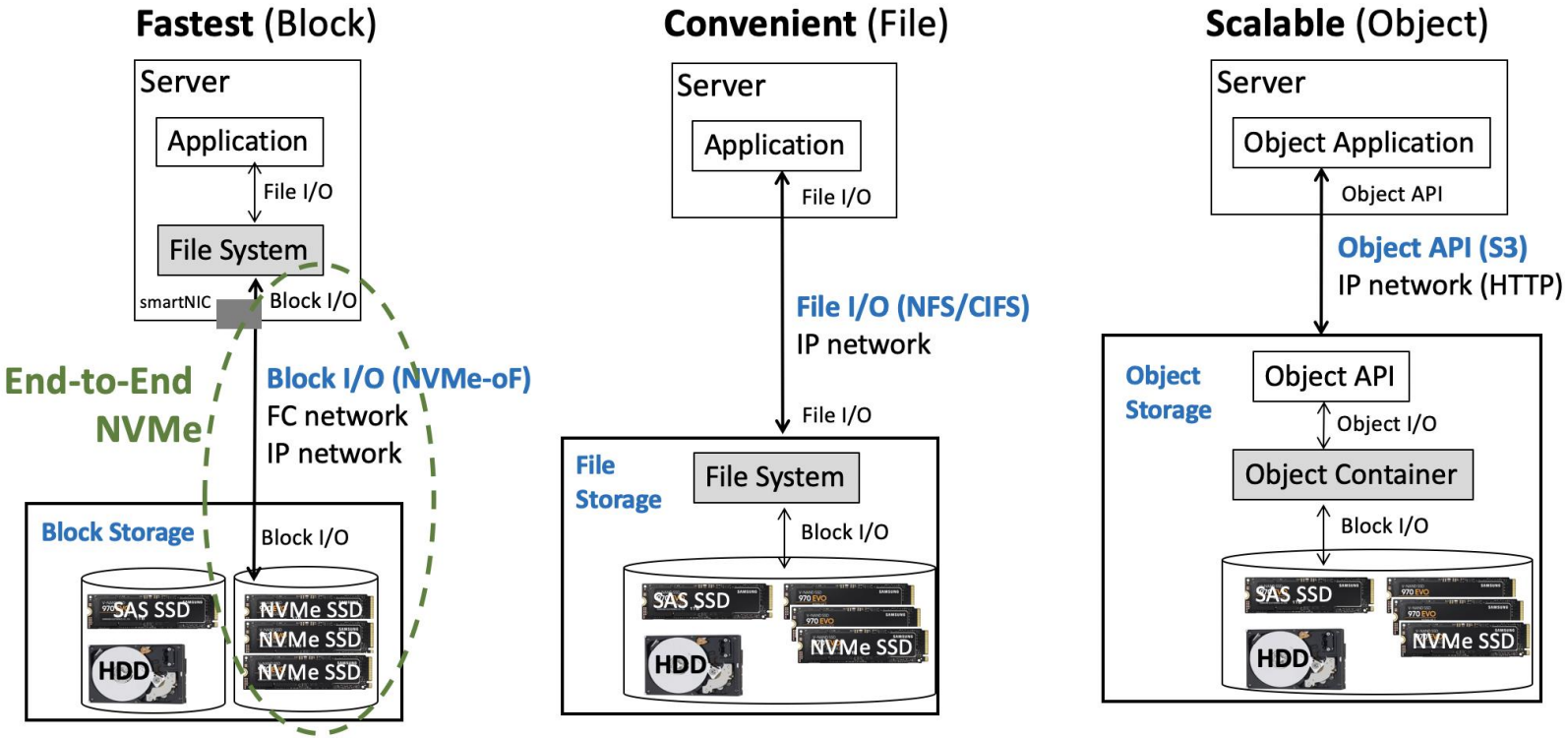
- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

NVMe Additional Information

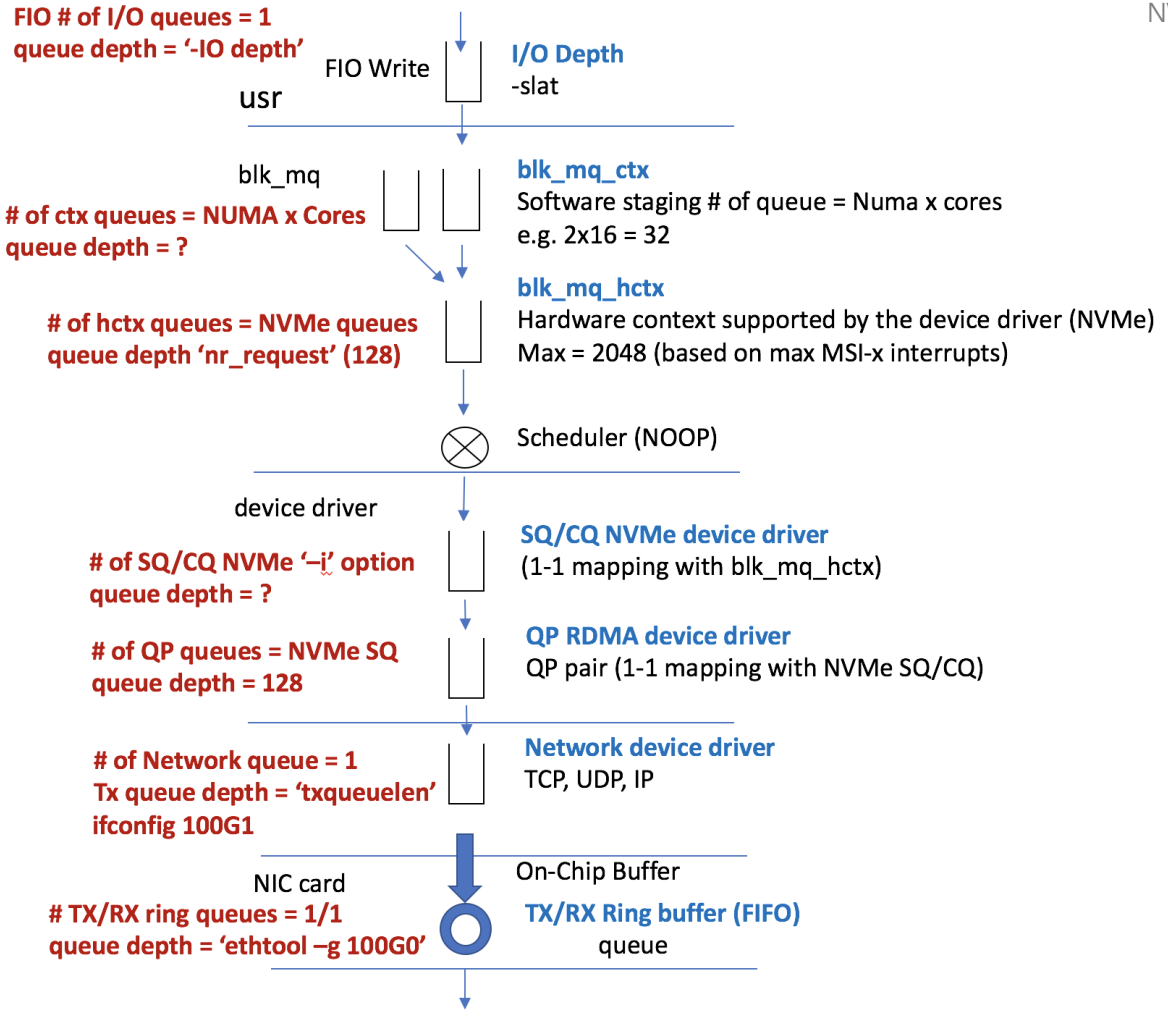
Storage Protocols Stack



NVMe Backend Storage –Block, File, Object Storage



Queues



NVMe CLI Commands (debian)

- nvme-admin-passthru(1)
- nvme-ana-log(1)
- nvme-attach-ns(1)
- nvme-boot-part-log(1)
- nvme-capacity-mgmt(1)
- nvme-changed-ns-list-log(1)
- nvme-cmdset-ind-id-ns(1)
- nvme-compare(1)
- nvme-connect-all(1)
- nvme-connect(1)
- nvme-copy(1)
- nvme-create-ns(1)
- nvme-delete-ns(1)
- nvme-dera-stat(1)
- nvme-detach-ns(1)
- nvme-device-self-test(1)
- nvme-dir-receive(1)
- nvme-dir-send(1)
- nvme-disconnect-all(1)
- nvme-disconnect(1)
- nvme-discover(1)
- nvme-dsm(1)
- nvme-effects-log(1)
- nvme-endurance-event-agg-log(1)
- nvme-endurance-log(1)
- nvme-error-log(1)
- nvme-fid-support-effects-log(1)
- nvme-flush(1)
- nvme-format(1)
- nvme-fw-activate(1)
- nvme-fw-commit(1)
- nvme-fw-download(1)
- nvme-fw-log(1)
- nvme-gen-hostnqn(1)
- nvme-get-feature(1)
- nvme-get-lba-status(1)
- nvme-get-log(1)
- nvme-get-ns-id(1)
- nvme-get-property(1)
- nvme-help(1)
- nvme-huawei-id-ctrl(1)
- nvme-huawei-list(1)
- nvme-id-ctrl(1)
- nvme-id-domain(1)
- nvme-id-iocs(1)
- nvme-id-ns(1)
- nvme-id-nvmset(1)
- nvme-intel-id-ctrl(1)
- nvme-intel-internal-log(1)
- nvme-intel-lat-stats(1)
- nvme-intel-market-name(1)
- nvme-intel-smart-log-add(1)
- nvme-intel-temp-stats(1)
- nvme-io-passthru(1)
- nvme-lba-status-log(1)
- nvme-list-ctrl(1)
- nvme-list-endgrp(1)
- nvme-list-ns(1)
- nvme-list-subsys(1)
- nvme-list(1)
- nvme-lnvm-create(1)
- nvme-lnvm-diag-bbtbl(1)
- nvme-lnvm-diag-set-bbtbl(1)
- nvme-lnvm-factory(1)
- nvme-lnvm-id-ns(1)
- nvme-lnvm-info(1)
- nvme-lnvm-init(1)
- nvme-lnvm-list(1)
- nvme-lnvm-remove(1)
- nvme-lockdown(1)
- nvme-micron-clear-pcie-errors(1)
- nvme-micron-internal-log(1)
- nvme-micron-nand-stats(1)
- nvme-micron-pcie-stats(1)
- nvme-micron-selective-download(1)
- nvme-micron-smart-add-log(1)
- nvme-micron-temperature-stats(1)
- nvme-netapp-ontapdevices(1)
- nvme-netapp-smdevices(1)
- nvme-ns-descs(1)
- nvme-ns-rescan(1)
- nvme-nvm-id-ctrl(1)
- nvme-persistent-event-log(1)
- nvme-pred-lat-event-agg-log(1)
- nvme-predictable-lat-log(1)
- nvme-primary-ctrl-caps(1)
- nvme-read(1)
- nvme-reset(1)
- nvme-resv-acquire(1)
- nvme-resv-notif-log(1)
- nvme-resv-register(1)
- nvme-resv-release(1)
- nvme-resv-report(1)
- nvme-rpmb(1)
- nvme-sanitize-log(1)
- nvme-sanitize(1)
- nvme-security-recv(1)
- nvme-security-send(1)
- nvme-self-test-log(1)
- nvme-set-feature(1)
- nvme-set-property(1)
- nvme-show-hostnqn(1)
- nvme-show-regs(1)
- nvme-smart-log(1)
- nvme-subsystem-reset(1)
- nvme-supported-log-pages(1)
- nvme-telemetry-log(1)
- nvme-toshiba-clear-pcie-correctable-errors(1)
- nvme-toshiba-vs-internal-log(1)
- nvme-toshiba-vs-smart-add-log(1)
- nvme-transcend-badblock(1)
- nvme-transcend-healthvalue(1)
- nvme-verify(1)
- nvme-virtium-save-smart-to-vtview-log(1)
- nvme-virtium-show-identify(1)
- nvme-wdc-cap-diag(1)
- nvme-wdc-capabilities(1)
- nvme-wdc-clear-assert-dump(1)
- nvme-wdc-clear-fw-activate-history(1)
- nvme-wdc-clear-pcie-corr(1)
- nvme-wdc-clear-pcie-correctable-errors(1)
- nvme-wdc-cloud-SSD-plugin-version(1)
- nvme-wdc-drive-essentials(1)
- nvme-wdc-drive-log(1)
- nvme-wdc-drive-resize(1)
- nvme-wdc-enc-get-log(1)
- nvme-wdc-get-crash-dump(1)
- nvme-wdc-get-drive-status(1)
- nvme-wdc-get-latency-monitor-log(1)
- nvme-wdc-get-pfail-dump(1)
- nvme-wdc-id-ctrl(1)
- nvme-wdc-log-page-directory(1)
- nvme-wdc-namespace-resize(1)
- nvme-wdc-purge-monitor(1)
- nvme-wdc-purge(1)
- nvme-wdc-smart-add-log(1)
- nvme-wdc-smart-log-add(1)
- nvme-wdc-vs-drive-info(1)
- nvme-wdc-vs-error-reason-identifier(1)
- nvme-wdc-vs-fw-activate-history(1)
- nvme-wdc-vs-internal-log(1)
- nvme-wdc-vs-nand-stats(1)
- nvme-wdc-vs-smart-add-log(1)
- nvme-wdc-vs-telemetry-controller-option(1)
- nvme-wdc-vs-temperature-stats(1)
- nvme-write-uncor(1)
- nvme-write-zeroes(1)
- nvme-write(1)
- nvme-zns-changed-zone-list(1)
- nvme-zns-close-zone(1)
- nvme-zns-finish-zone(1)
- nvme-zns-id-ctrl(1)
- nvme-zns-id-ns(1)
- nvme-zns-offline-zone(1)
- nvme-zns-open-zone(1)
- nvme-zns-report-zones(1)
- nvme-zns-reset-zone(1)
- nvme-zns-set-zone-desc(1)
- nvme-zns-zone-append(1)
- nvme-zns-zone-mgmt-recv(1)
- nvme-zns-zone-mgmt-send(1)
- nvme(1)

NVMe-oF Comparison

FC-SB/CKD FICON (FC)

(Not NVMe)

IBM Z mainframes process 30 billion transactions each day, including 87% of all credit card transactions on the planet.

-96 of the world's top 100 banks and 9 out of 10 of the world's biggest insurance companies still depend on mainframes (source google)

-Mainframe storage standard

FC-SCSI (FCP)

-120millions* FC ports shipped
-46millions* in use (FCIA* website)
-Dedicated purpose built Storage Network
-Built in Discovery & Name services
-Zoning & Security
-Lossless Fabric/Zero Copy
-Certified designs
-Gold standard in Enterprise storage

NVMe-FC

-Faster than FC-SCSI
-Advance Error detection & recovery
-Same FC transport

32G/64G Fibre Channel Transport

NVMe-IB

-Infiniband Transport
-Lossless Infiniband Links
-HPC supercomputer
-RDMA, Zero Copy
-Low Latency
-IB stack offload

200G Infiniband Transport

NVMe-UPD/RoCEv2

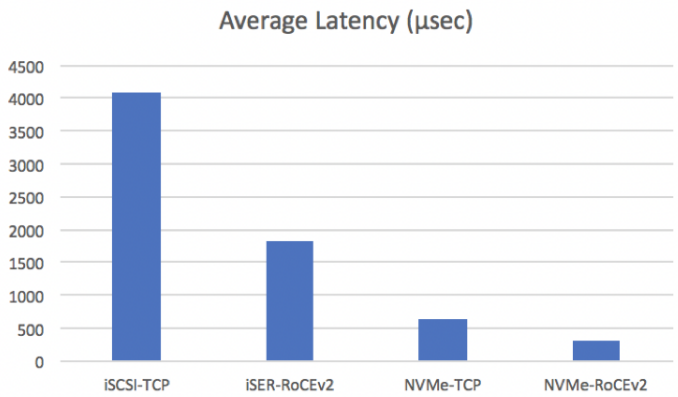
-Infiniband Transport
-Lossless Ethernet Links
-RDMA, Zero Copy
-Low Latency
-IB stack offload
-Higher Performance than TCP

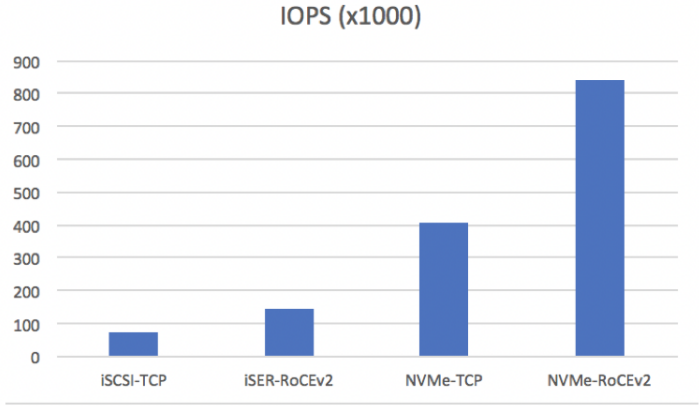
NVMe-TCP

-Ubiquitous
-Scalable, simpler
-Price/Performance benefit
-Ample skillset
-(Faster than iSCSI)

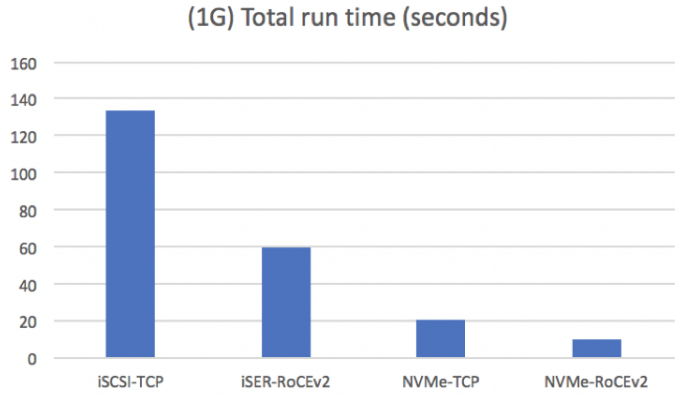
400G Ethernet Transport

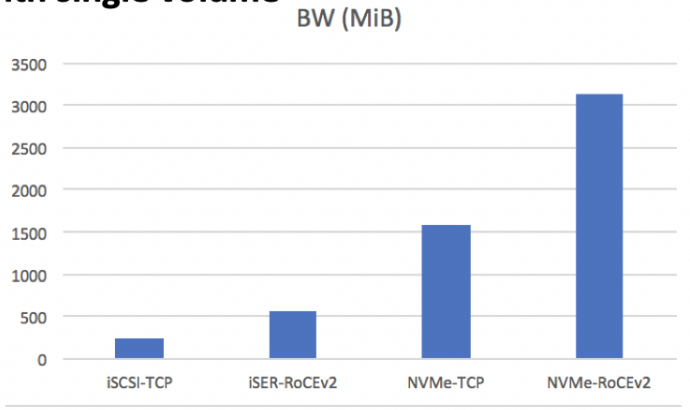
Performance (iSCSI vs NVMe-IP)





4KB Random Reads 1G with single volume





NVMe Commands

Control Plane

NVMe-Admin

-Create I/O SQ
-Delete I/O SQ
-Create I/O CQ
-Delete I/O CQ
-Get Features
-Set Features
-Keep Alive
-Identify
-Get Log Pages
-Abort
-Directive Send
-Directive Receive
-Async. Event Req.
-Namespace Mgmt.
-Namespace Attachment
-Virtualization Mgmt.
-Firmware Image Download
-Firmware Commit
-Device Self test
-NVMe-MI Send
-NVMe-MI Receive
-Door bell Buffer Config.
-Format NVM
-Sanitize
-Get LBA Status
-Security Send
-Security Receive

Transport over Fabric

NVMe-oF

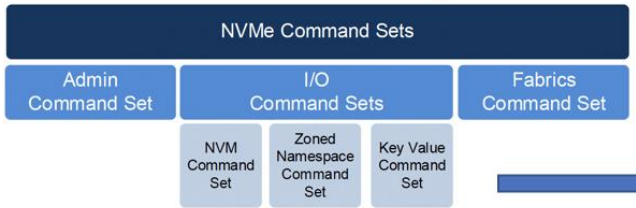
-Connect
-Disconnect
-Authentication Send
-Authentication Receive
-Property Get
-Property Set

Data Exchange

NVMe-I/O

-Write
-Write Uncorrectable
-Write Zeroes
-Flush
-Read
-Compare
-Verify
-Dataset Management
-Reservation Report
-Reservation Acquire
-Reservation Release

NVMe 2.0

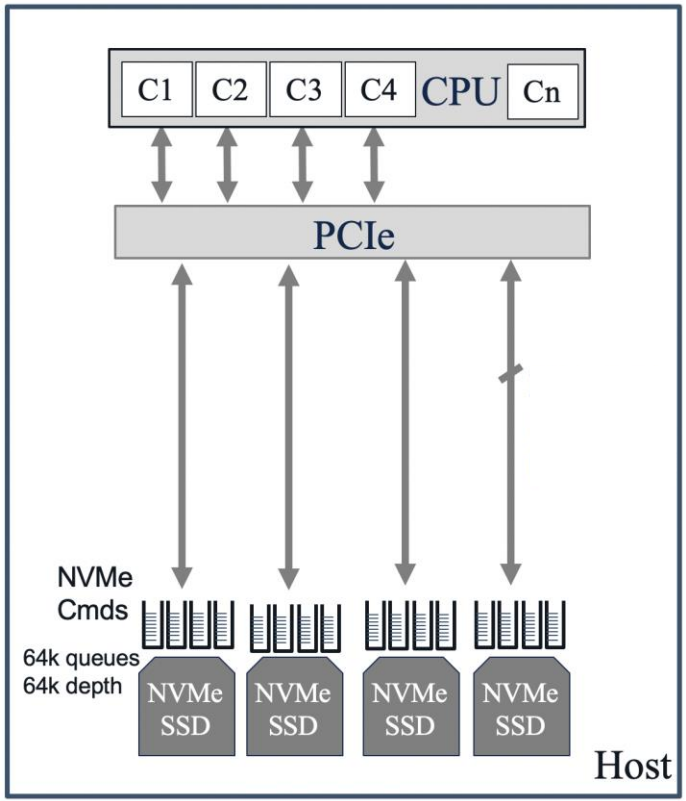


NVMe SSD

ZNS NVMe

Key Value NVMe

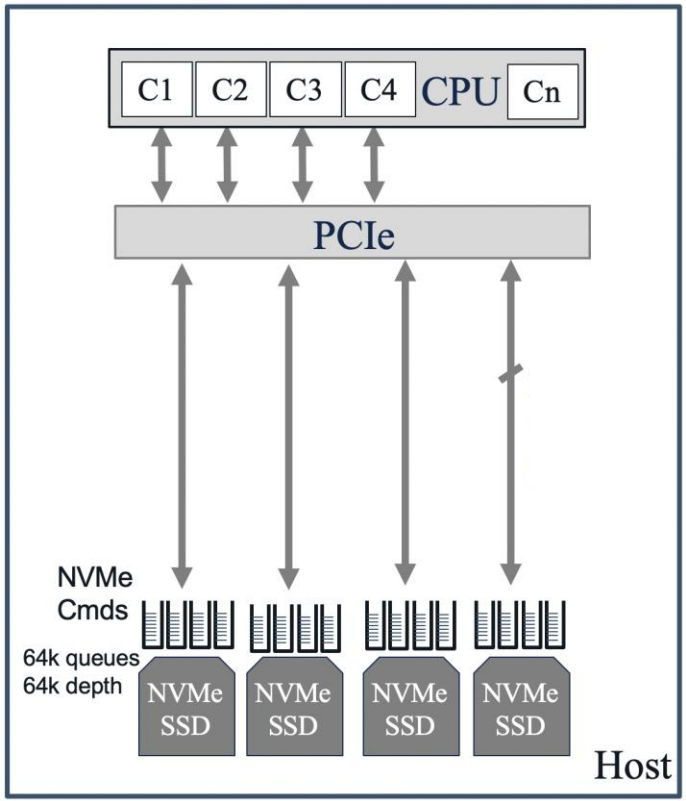
NVMe SSD Form Factors (M.2)



M.2 is a form factor specification for internally mounted SSDs. Formerly known as Next Generation Form Factor (NGFF) and comes in various widths and lengths.

Dimensions
16mm x 20mm
22mm x 30mm
22mm x 80mm
22mm x 110mm

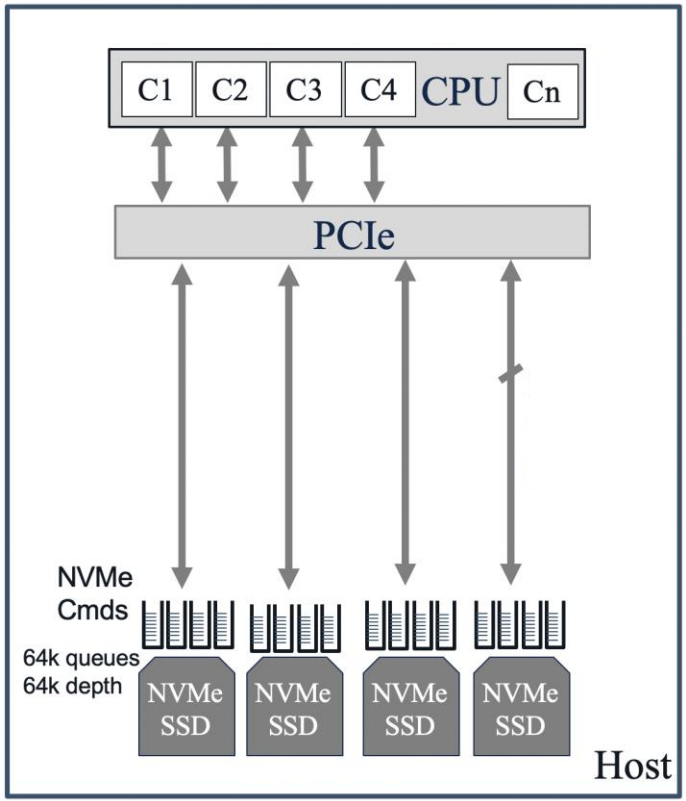
NVMe SSD Form Factors (U.2)



U.2 is defined as compliance with the PCI Express SFF-8639 Module specification, and no longer typically references SAS or SATA SSDs.

Dimensions
2.5-inch(7mm) [69.85x100x7 mm]
2.5-inch(15mm) [69.85x100x15mm]

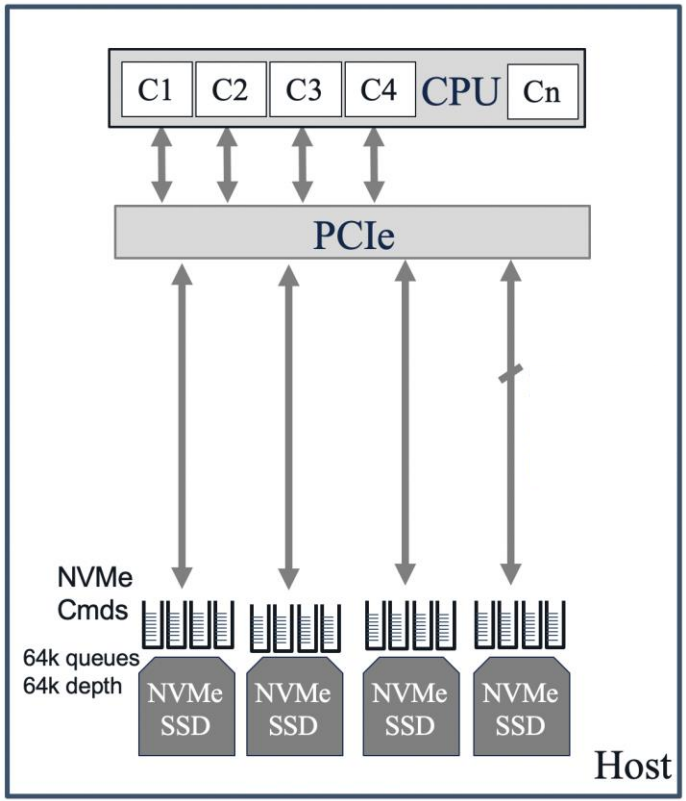
NVMe SSD Form Factors (AIC)



An Add-in Card (AIC) is a solid-state device that utilizes a stan card form factor such as a PCIe card. In addition, the larger size allows for the potential to add computational function to the storage device.

Dimensions
- -Full Height
- -Half Height
- -Low Profile

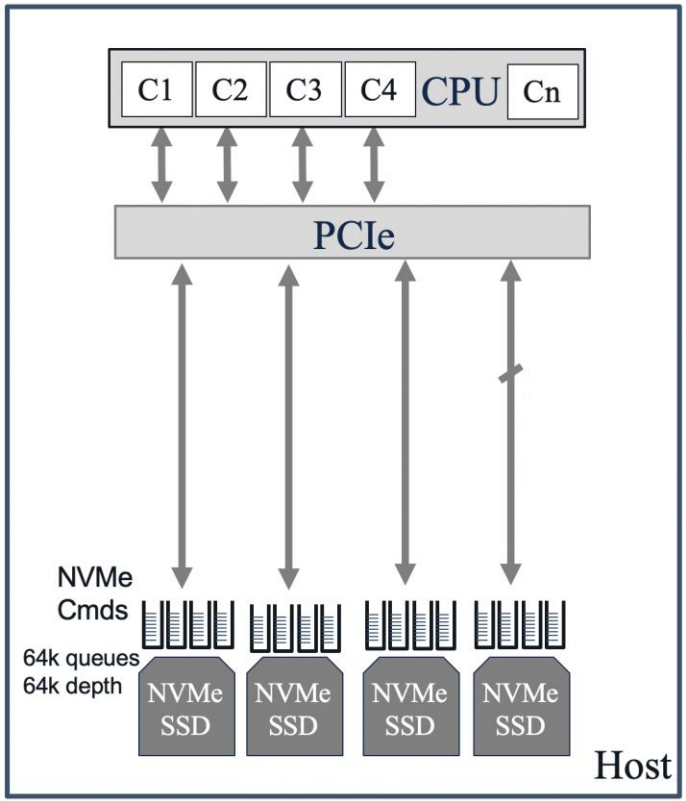
NVMe SSD Form Factors (EDSFF)



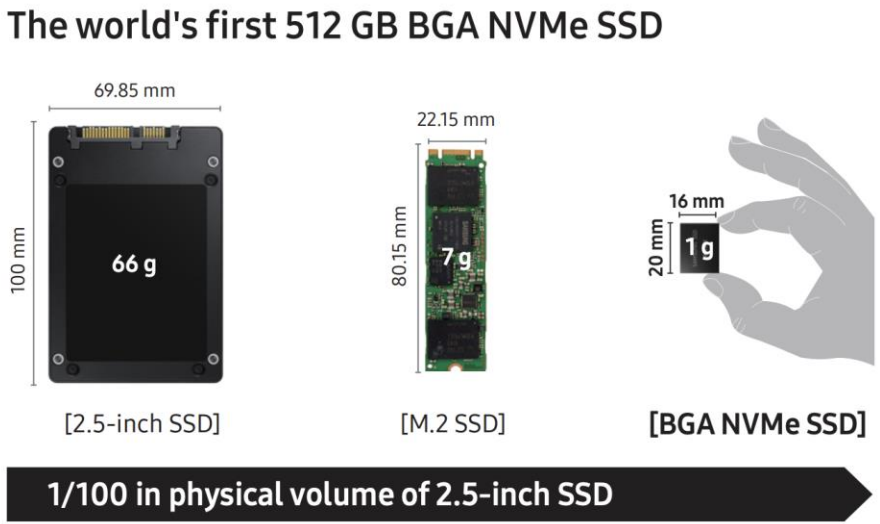
EDSFF stands for Enterprise and Data Center Standard Form Fac
The family of specifications were developed by a group of 15 co
working together to address the concerns of data center storage
now maintained by SNIA as part of the SFF Technology Affiliate
Technical Work Group (SFF TA TWG).

Dimensions (thickness)
E1.L (long) 9.5mm, 18mm
E1.S (short) 5.9mm, 8.01mm, 9.5mm, 15mm, 25mm
E3.S (short) 7.5mm, E3.S 2T 16.8mm
E3.L (long) 7.5mm, E3.L 2T 16.8mm

NVMe SSD Form Factors (BGA)



In 2016 Samsung started to mass produce the industry's first NVMe PCIe solid state drive (SSD) in a single ball grid array (BGA) package, for use in next-generation PCs and ultra-slim notebook PCs.



NVMe Controller Properties Registers

Offset (0h-E1Ch)

CAP: Controller Capabilities Supported
VS: Version
INTM: Interrupt Mask Set/Clear
CC: Controller Configuration
CSTS: Controller Status
NSSR: NVM Subsystem Reset
NSSD: NVM Subsystem Shutdown
AQA: Admin Queue Attributes
ASQ: Admin Submission Queue Base Address
ACQ: Admin Completion Queue Base Address
CMB: Controller Memory Info
BP: Boot Partition Info
CRTO: Controller Ready Timeouts
PM: Persistent Memory Info

Admin Queues Details

Offset (1000h) Transport Specific

The PCIe transport supports Controller Properties as memory mapped registers that are located in the address range specified in the MLBAR/MUBAR registers (PCI BAR0 and BAR1).

For NVMe-oF controller properties may be read with the “Property Get” command and may be written with the “Property Set” command with controllers using the message-based transport model.

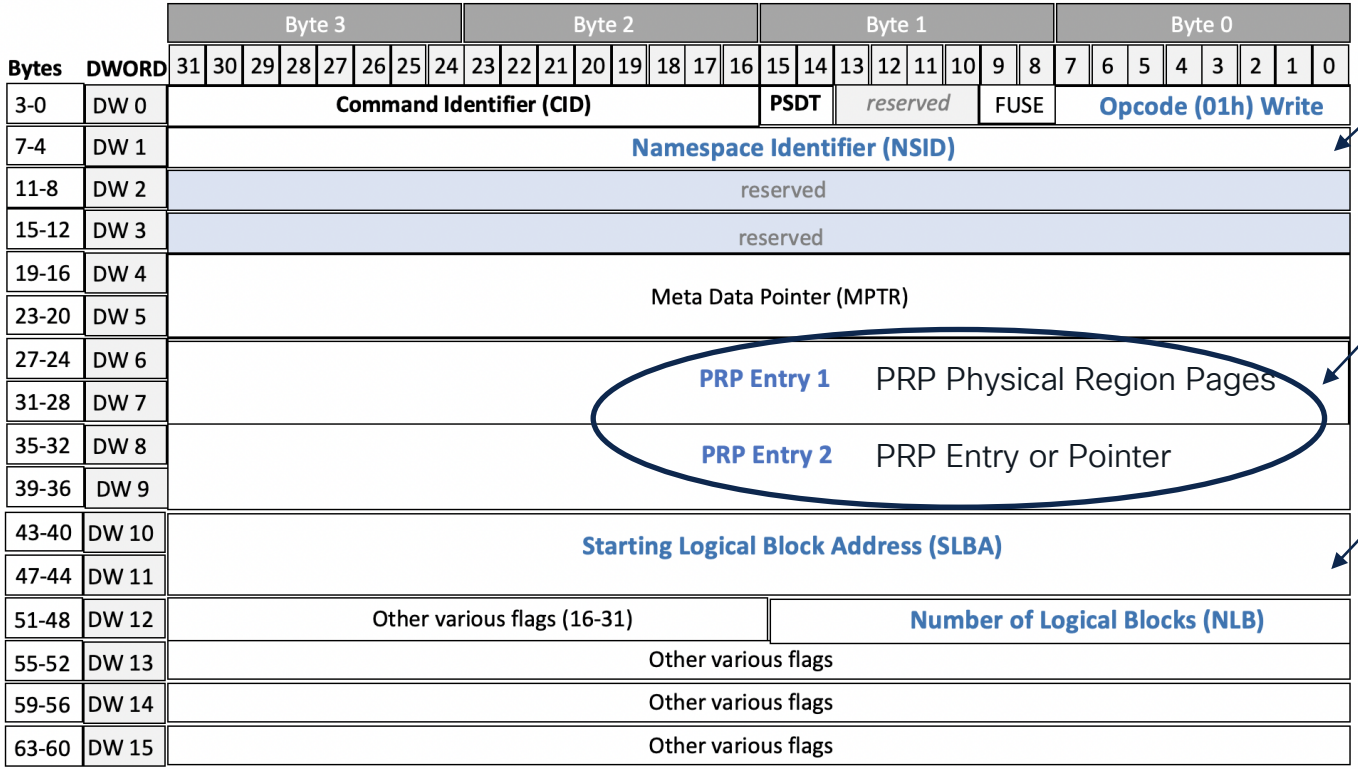
Doorbells Details

Offset 1000h

SQ0TDBL: Submission Queue 0 Tail Doorbell (Admin)
CQ0HDBL: Completion Queue 0 Head Doorbell (Admin
SQ1TDBL: Submission Queue 1 Tail Doorbell
CQ1TDBL: Completion Queue 1 Head Doorbell

SQyTDBL: Submission Queue y Tail Doorbell
CQyTDBL: Completion Queue y Head Doorbell

SQE packet (Write command, opcode '01h')



Bytes	DWORD	Byte 3 (31–24)	Byte 2 (23–16)	Byte 1 (15–8)	Byte 0 (7–0)
3-0	DW 0	Command Identifier (CID)		PSDT / reserved / FUSE	Opcode (01h) Write
7-4	DW 1	Namespace Identifier (NSID)			
11-8	DW 2	reserved			
15-12	DW 3	reserved			
19-16	DW 4	Meta Data Pointer (MPTR)			
23-20	DW 5				
27-24	DW 6	PRP Entry 1 — PRP Physical Region Pages			
31-28	DW 7				
35-32	DW 8	PRP Entry 2 — PRP Entry or Pointer			
39-36	DW 9				
43-40	DW 10	Starting Logical Block Address (SLBA)			
47-44	DW 11				
51-48	DW 12	Other various flags (16-31)		Number of Logical Blocks (NLB)	
55-52	DW 13	Other various flags			
59-56	DW 14	Other various flags			
63-60	DW 15	Other various flags			

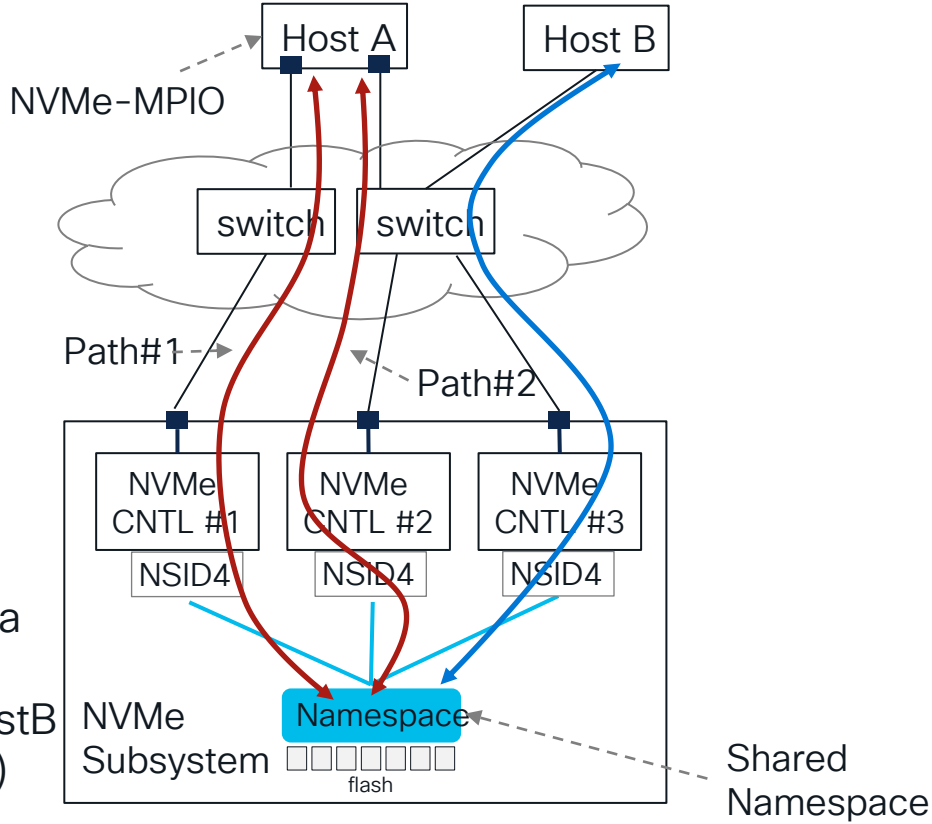
NSID that will be used for write

Host sets the buffer addresses of the da to be transferred

LBA information

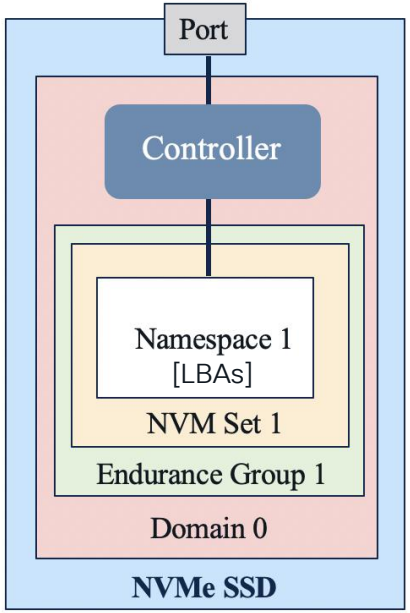
Shared Namespace/ANA

1-Multipathing at Host A (DM, NVMe)
2-Discovery Log page will show controller addresses
3-Identify Controller will show ANA support with various ANA capability flags
4-Optional “NVMe Reservation” can coordinate host access to the Namespace
5-NVMe subsystem must have at-least 2 controllers.
6-A given controller can only talk to one host at a time
7-Same Namespace is shared by HostA and HostB asymmetrically (different access characteristics)

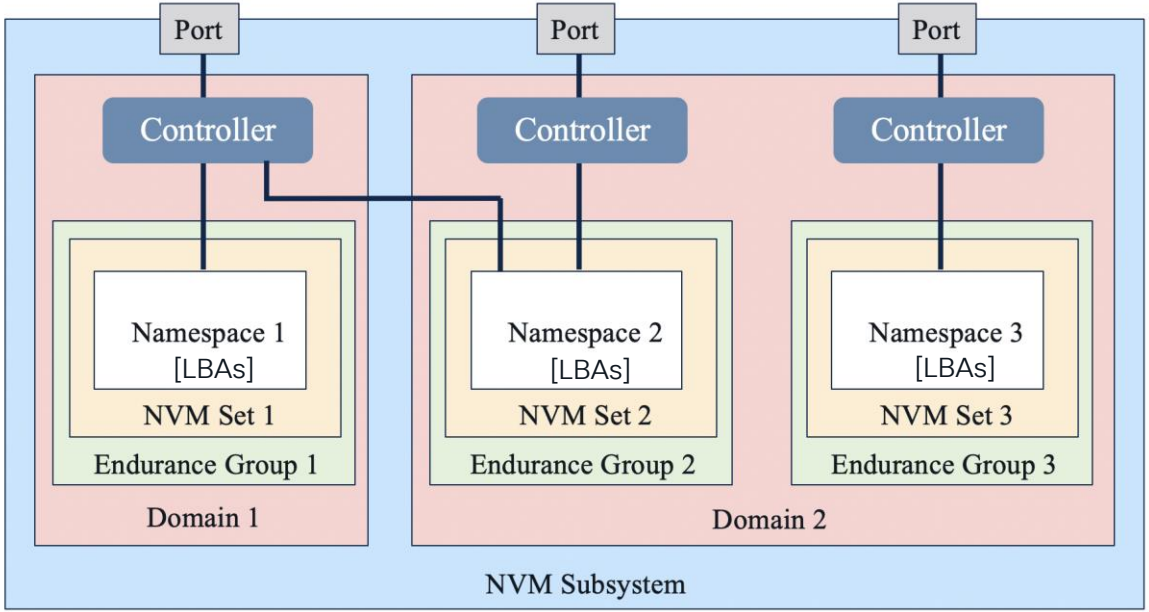


ANA = Asymmetric Namespace Access
MPIO = Multi-Path I/O
DM-MPIO = Device Mapper MPIO

Namespace Hierarchy

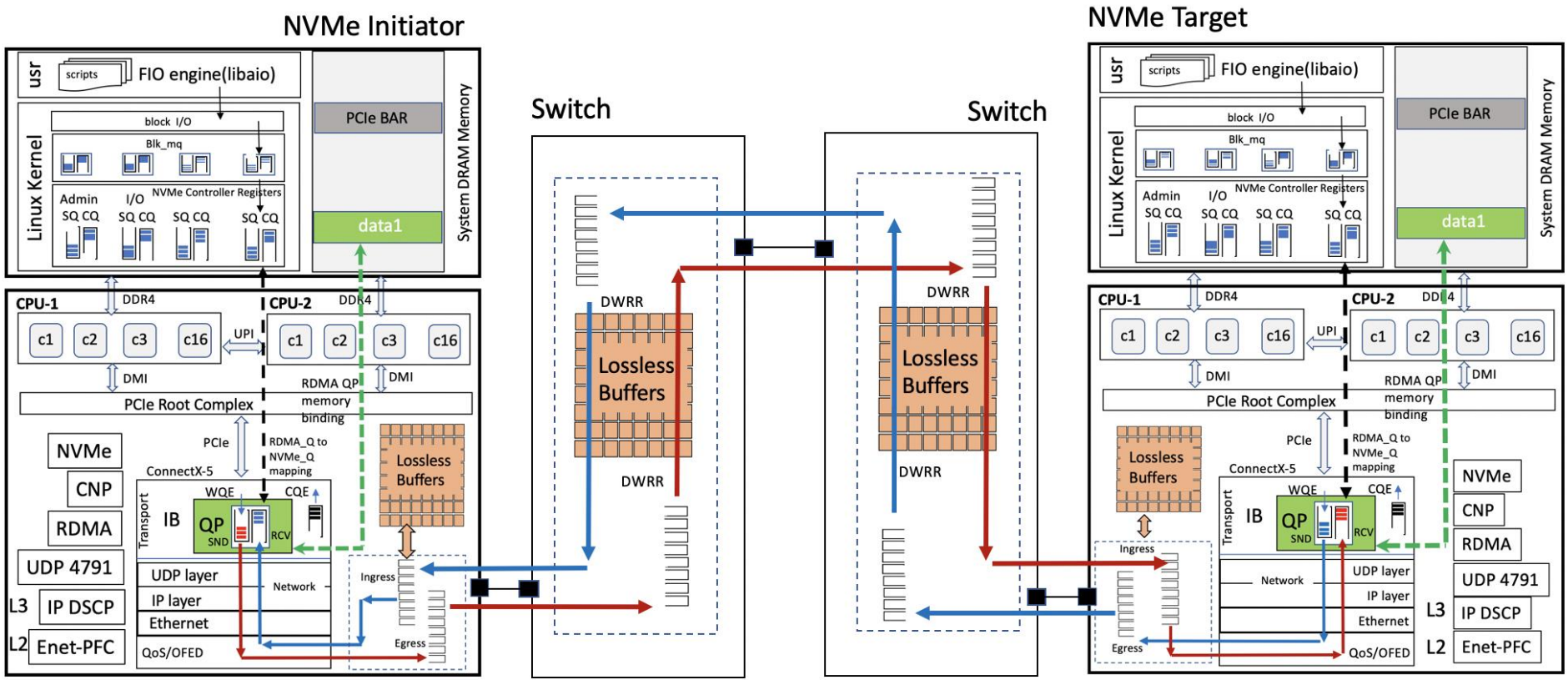


Single-Namespace NVM Subsystem



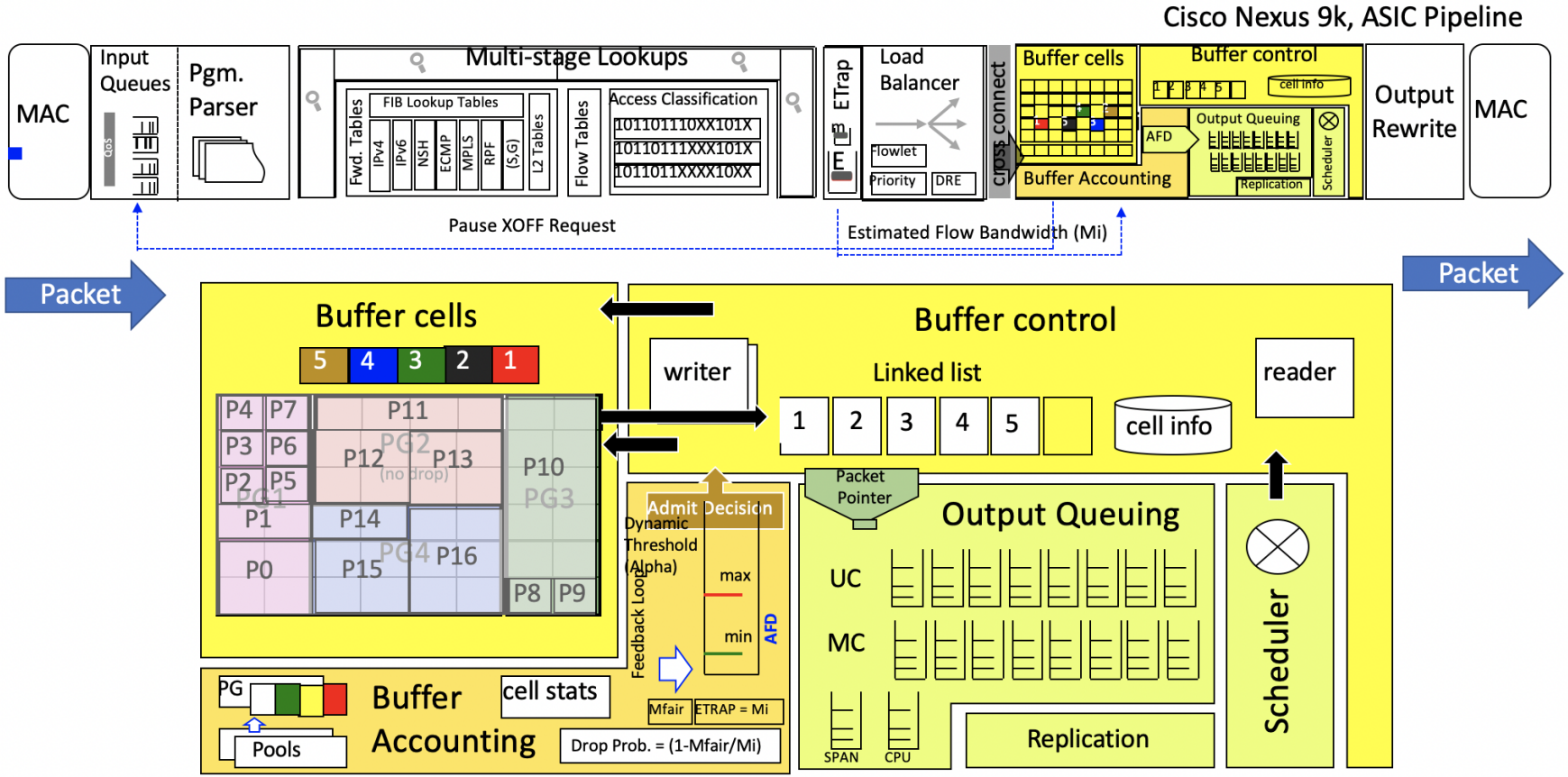
Complex NVM Subsystem

NVMe End to End Traffic Engineering



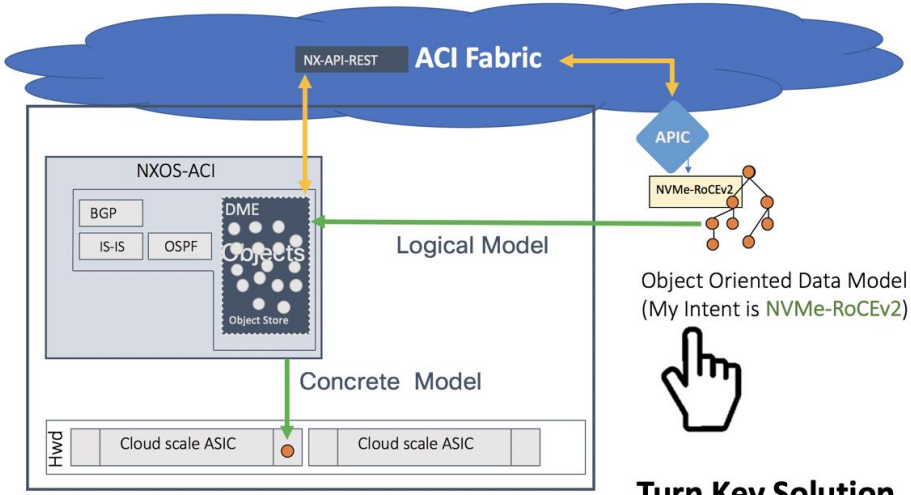
IB/CNP, DSCP, PFC, ECN

Cisco Cloud Scale ASIC -Smart Buffering



NVMe-oF Automation with Cisco ACI/APIC

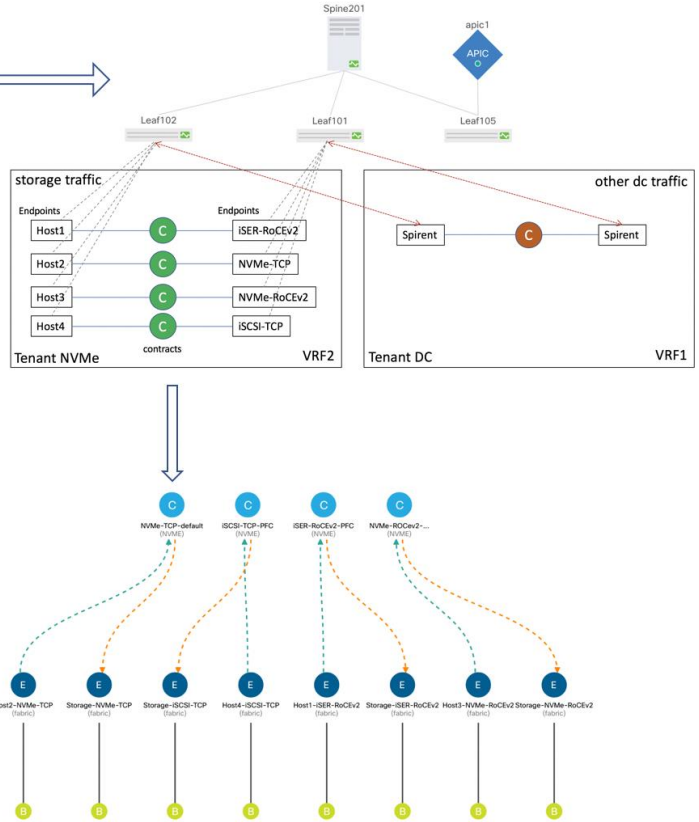
ACI –Zero Trust Security
(Tenants/VRF, End-Points, Contracts)



Nexus 9k ACI Switch

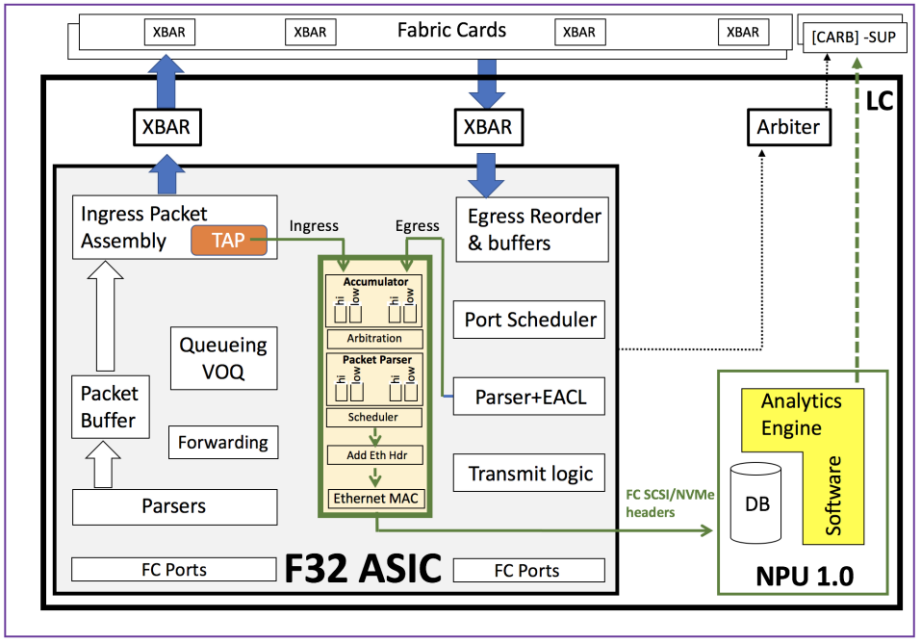
Object Oriented Data Model
(My Intent is NVMe-RoCEv2)

Turn Key Solution

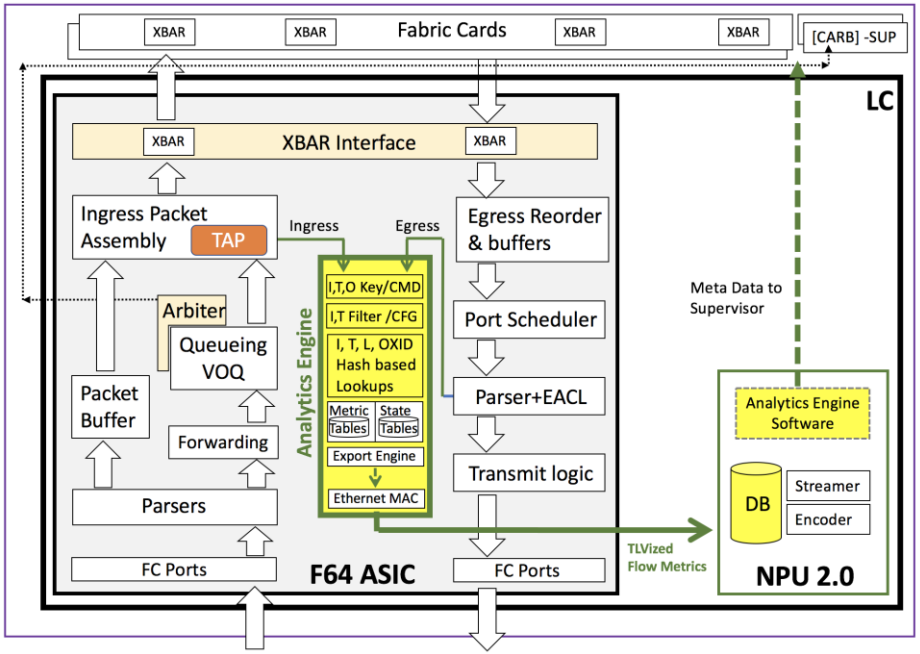


Cisco MDS 32G/64G Analytics

F32 GFC Line Card – Analytics



F64 GFC Line Card – Analytics



Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

3-NVMe Transport Options (FC, TCP, RoCEv2)

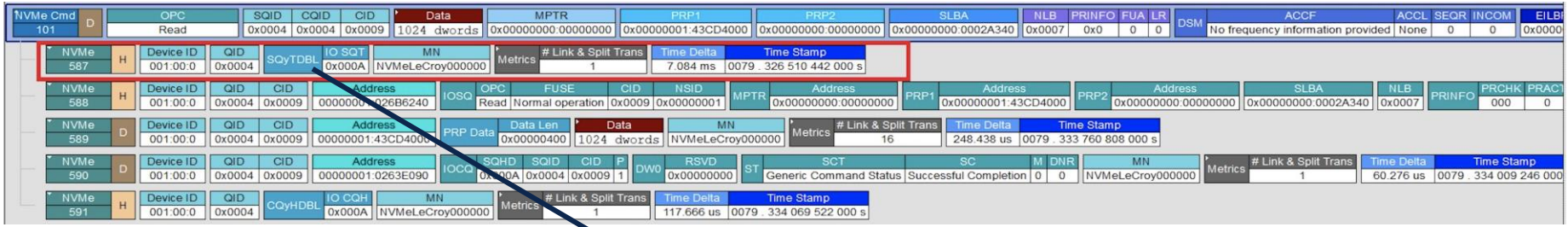
4-NVMe Datacenter Design

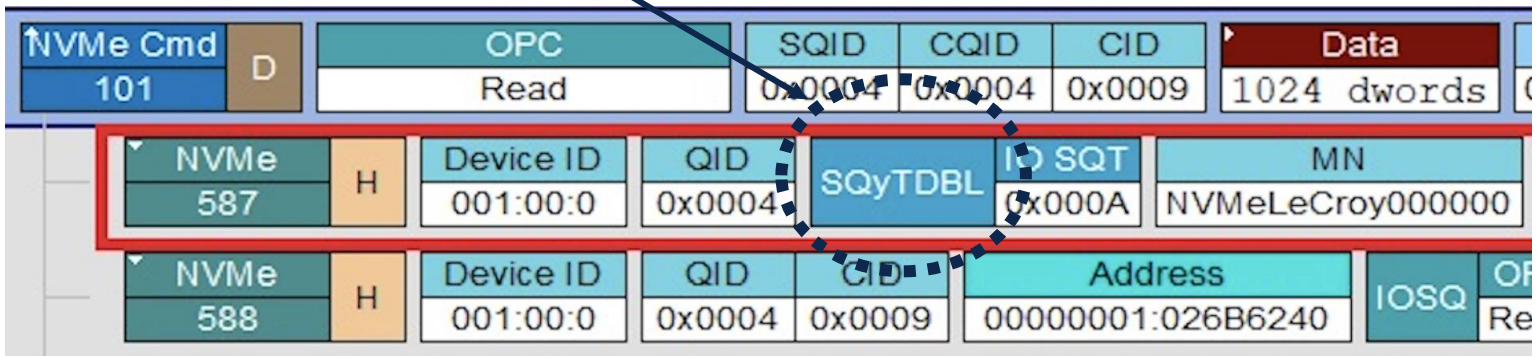
5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

NVMe/PCIe Traces

NVMe-PCIe Trace of a Doorbell Message



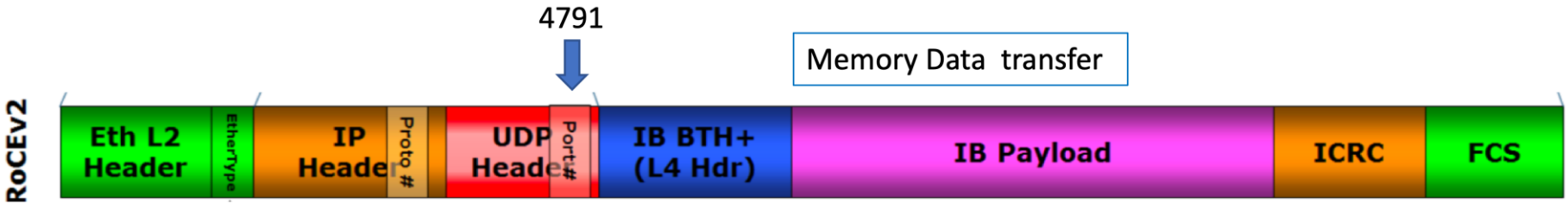


SQ Tail Doorbell

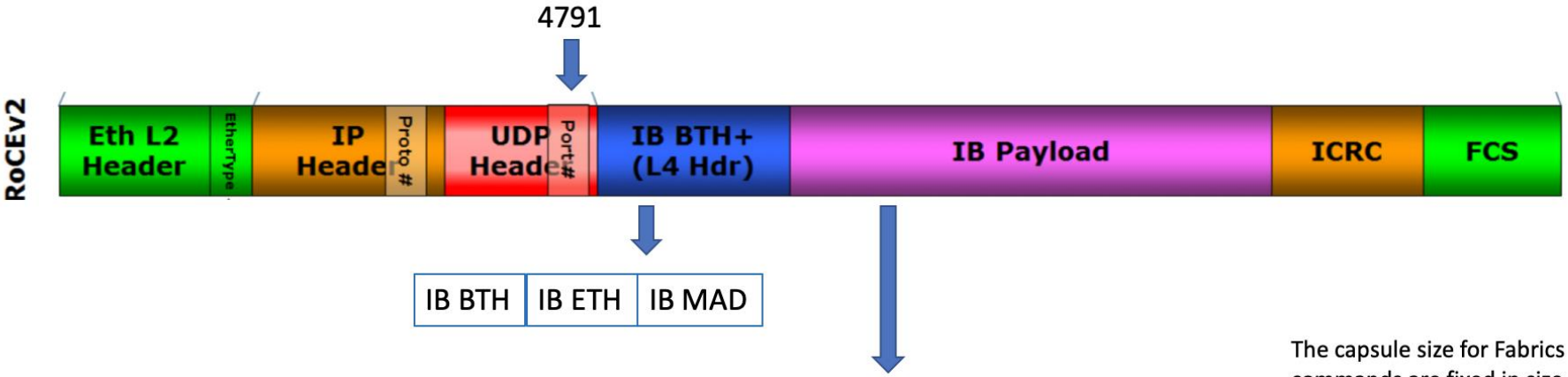
Trace: Courtesy of Teledyne Technologies

NVMe/RoCEv2 Traces

RoCEv2 Packet



RoCEv2 Packet



The five IBA transport service types are:

1) Reliable Connection (RC)
2) Reliable Datagram (RD)
3) Extended Reliable Connection (XRC)
4) Unreliable Datagram (UD)
5) Unreliable Connection (UC)

NVMe Command (Fabric/Admin/IO) (Host → Controller)

NVMe Response (Controller → Host)

Memory Data transfer

The capsule size for Fabrics commands are fixed in size regardless of whether commands are submitted on an Admin Queue or an I/O Queue. The command capsule size is 64 bytes and the response capsule size is 16 bytes.
The capsule sizes for the Admin Queue are fixed in size. The command capsule size is 64 bytes and the response capsule size is 16 bytes. In-capsule data is not supported for the Admin Queue.

BTH

OpCode[7-5]
000 RC
001 UC
010 RD
011 UD
100 CNP
101 XRC

OpCode[4-0]
SEND
RDMA WRITE
RDMA READ
Acknowledge
CmpSwap
FetchAdd

bits / bytes	31-24			23-16				15-8	7-0
0-3	OpCode			SE	M	Pad	TVer	Partition Key	
4-7	F/Res1 [a]	B/Res1 [a]	Reserved 6 [a]	Destination QP					
8-11	A	Reserved 7		PSN - Packet Sequence Number					

Figure 66 Base Transport Header (BTH)

DestQP#: This field specifies the destination queue pair (QP) identifier. *Note: SrcQP# is in ETH*

AckReq: Requests responder to schedule an acknowledgment on the associated QP.

PSN: This field is used to identify the position of a packet within a sequence of packets. Depending upon the transport service type and / or implementation requirements, a responder may validate the PSN to detect missing packets.

Transport Function	Transport Service					
	Reliable Connection	Unreliable Connection	XRC	Reliable Datagram	Unreliable Datagram	Raw Datagram
SEND	supported	supported	supported	supported	supported	not applicable
RESYNC	not supported	not supported	not supported	supported	not supported	not supported
RDMA WRITE	supported	supported	supported	supported	not supported	not applicable
RDMA READ	supported	not supported	supported	supported	not supported	not applicable
ATOMIC Operations	optional support	not supported	optional support	optional support	not supported	not applicable

The InfiniBand Architecture defines a Raw service which does not use the InfiniBand transport (InfiniBand Specification Vol.1 Rev 1.2.1 Section 9.8.4). The Raw services as defined in the base specification are provided by the InfiniBand link layer. Similarly to RoCE, since RoCEv2 does not use the InfiniBand link layer, IB RAW datagrams, namely Raw Ethertype and Raw IPv6, are not applicable for RoCEv2.

BTH Opcode (Transport Type)

00	RC	Send First
01	RC	SEND Middle
02	RC	SEND Last
03	RC	SEND Last with Immediate
04	RC	SEND Only
05	RC	SEND Only with Immediate
06	RC	RDMA WRITE First
07	RC	RDMA WRITE Middle
08	RC	RDMA WRITE Last
09	RC	RDMA WRITE Last with Immediate
0A	RC	RDMA WRITE Only
0B	RC	RDMA WRITE Only with Immediate
0C	RC	RDMA READ Request
0D	RC	RDMA READ response First
0E	RC	RDMA READ response Middle
0F	RC	RDMA READ response Last
10	RC	RDMA READ response Only
11	RC	Acknowledge
12	RC	ATOMIC Acknowledge
13	RC	CmpSwap
14	RC	FetchAdd
15	RC	Reserved
16	RC	SEND Last with Invalidate
17	RC	SEND Only with Invalidate

RC

A0	XRC	Send First
A1	XRC	SEND Middle
A2	XRC	SEND Last
A3	XRC	SEND Last with Immediate
A4	XRC	SEND Only
A5	XRC	SEND Only with Immediate
A6	XRC	RDMA WRITE First
A7	XRC	RDMA WRITE Middle
A8	XRC	RDMA WRITE Last
A9	XRC	RDMA WRITE Last with Immediate
AA	XRC	RDMA WRITE Only
AB	XRC	RDMA WRITE Only with Immediate
AC	XRC	RDMA READ Request
AD	XRC	RDMA READ response First
AE	XRC	RDMA READ response Middle
AF	XRC	RDMA READ response Last
B0	XRC	RDMA READ response Only
B1	XRC	Acknowledge
B2	XRC	ATOMIC Acknowledge
B3	XRC	CmpSwap
B4	XRC	FetchAdd
B5	XRC	Reserved
B6	XRC	SEND Last with Invalidate
B7	XRC	SEND Only with Invalidate

XRC

40	RD	Send First
41	RD	SEND Middle
42	RD	SEND Last
43	RD	SEND Last with Immediate
44	RD	SEND Only
45	RD	SEND Only with Immediate
46	RD	RDMA WRITE First
47	RD	RDMA WRITE Middle
48	RD	RDMA WRITE Last
49	RD	RDMA WRITE Last with Immediate
4A	RD	RDMA WRITE Only
4B	RD	RDMA WRITE Only with Immediate
4C	RD	RDMA READ Request
4D	RD	RDMA READ response First
4E	RD	RDMA READ response Middle
4F	RD	RDMA READ response Last
50	RD	RDMA READ response Only
51	RD	Acknowledge
52	RD	ATOMIC Acknowledge
53	RD	CmpSwap
54	RD	FetchAdd
55	RD	RESYNC

RD

20	UC	Send First
21	UC	SEND Middle
22	UC	SEND Last
23	UC	SEND Last with Immediate
24	UC	SEND Only
25	UC	SEND Only with Immediate
26	UC	RDMA WRITE First
27	UC	RDMA WRITE Middle
28	UC	RDMA WRITE Last
29	UC	RDMA WRITE Last with Immediate
2A	UC	RDMA WRITE Only
2B	UC	RDMA WRITE Only with Immediate

UC

64	UD	SEND only
65	UD	SEND only with Immediate
80	CNP	CNP

UD

CNP

Extended Transport Headers

RD-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	Reserve	EE-Context		

Reliable Datagram Extended Transport Header (RDETH)

D-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	Queue Key			
4-7	Reserve	Source QP		

Datagram Extended Transport Header (DETH)

R-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	Virtual Address (63-32)			
4-7	Virtual Address (31-0)			
8-11	R_Key			
12-15	DMA Length			

RDMA Extended Transport Header (RETH)

Atomic-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	Virtual Address (63-32)			
4-7	Virtual Address (31-0)			
8-11	R_Key			
12-15	Swap (or Add) Data (63-32)			
16-19	Swap (or Add) Data (31-0)			
20-23	Compare Data (63-32)			
24-27	Compare Data (31-0)			

ATOMIC Extended Transport Header (AtomicETH)

BTH OpCode has ETH details

RDMA WRITE Last	PayLd
RDMA WRITE Last with Immediate	ImmDt, PayLd
RDMA WRITE Only	RETH, PayLd
RDMA WRITE Only with Immediate	RETH, ImmDt, PayLd

A-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	Syndrome	MSN		

Acknowledge Extended Transport Header (AETH)

AtomicAck-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	Original Remote Data (63-32)			
4-7	Original Remote Data (31-0)			

ATOMIC Acknowledge Extended Transport Header (AtomicAckETH)

ImmDT

bits / bytes	31-24	23-16	15-8	7-0
0-3	Immediate Data			

Immediate Extended Transport Header (ImmDt)

I-ETH

bits / bytes	31-24	23-16	15-8	7-0
0-3	R_Key			

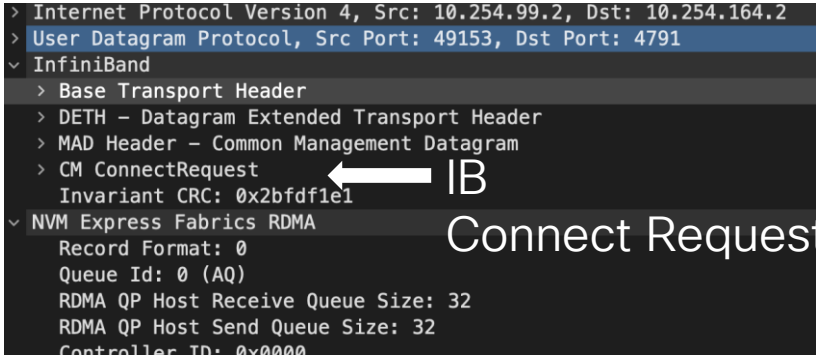
Invalidate Extended Transport Header (IETH)

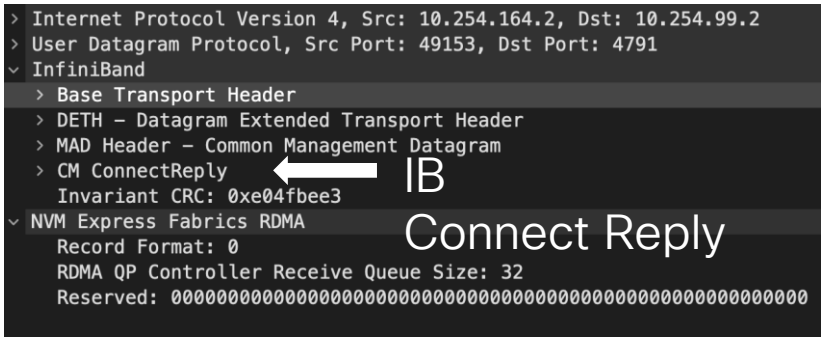
XRC-ETH

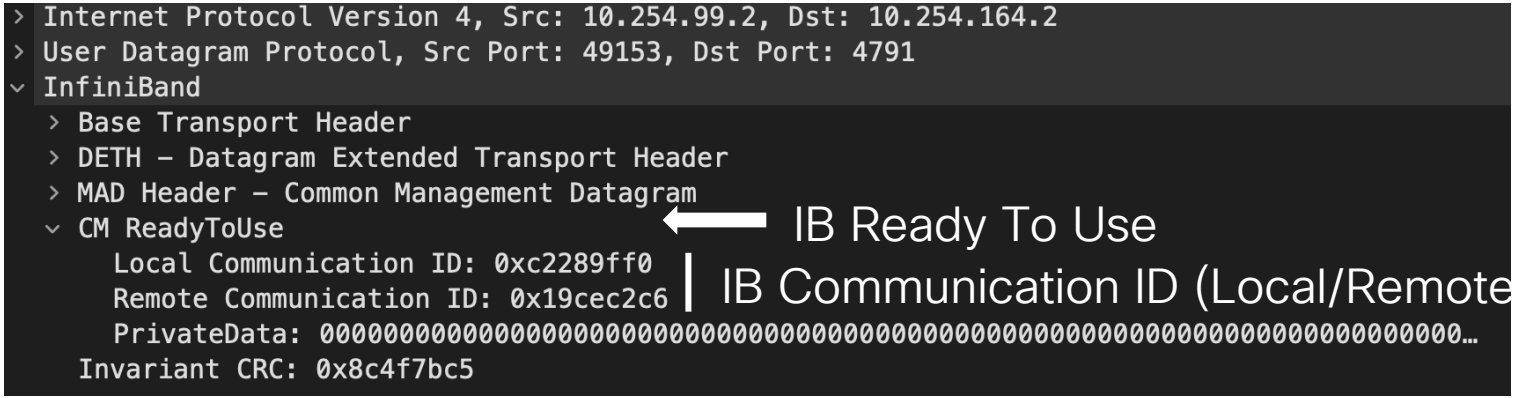
bits / bytes	31-24	23-16	15-8	7-0
0-3	Reserved	XRCSRQ		

XRC Extended Transport Header (XRCETH)

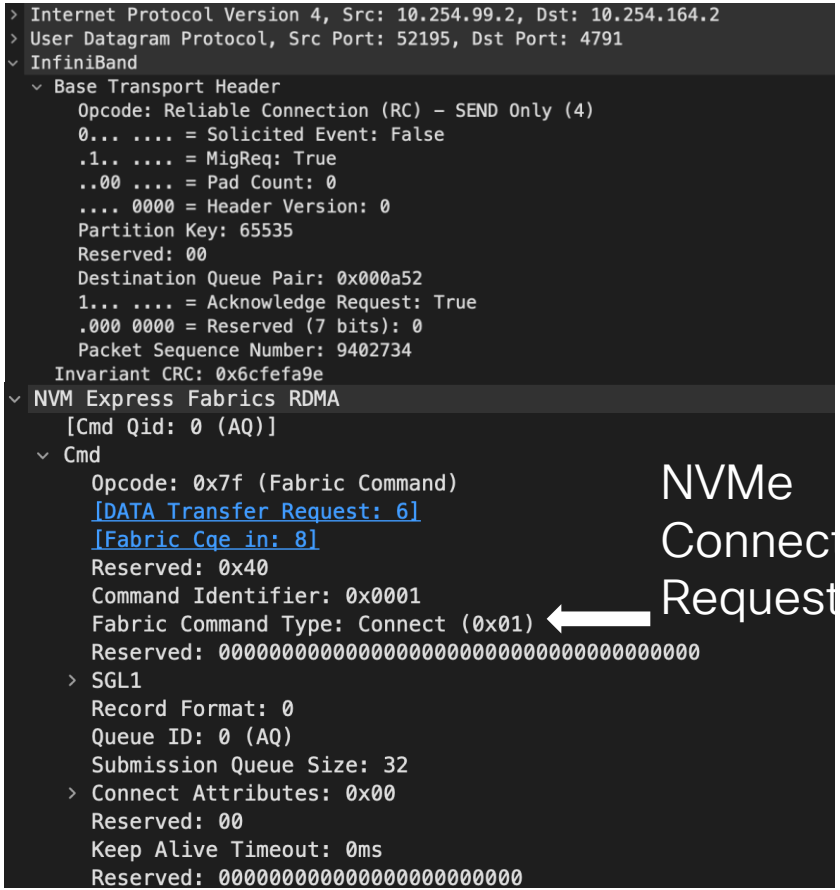
NVMe-RoCEv2 (IB -CM Connection Manager)



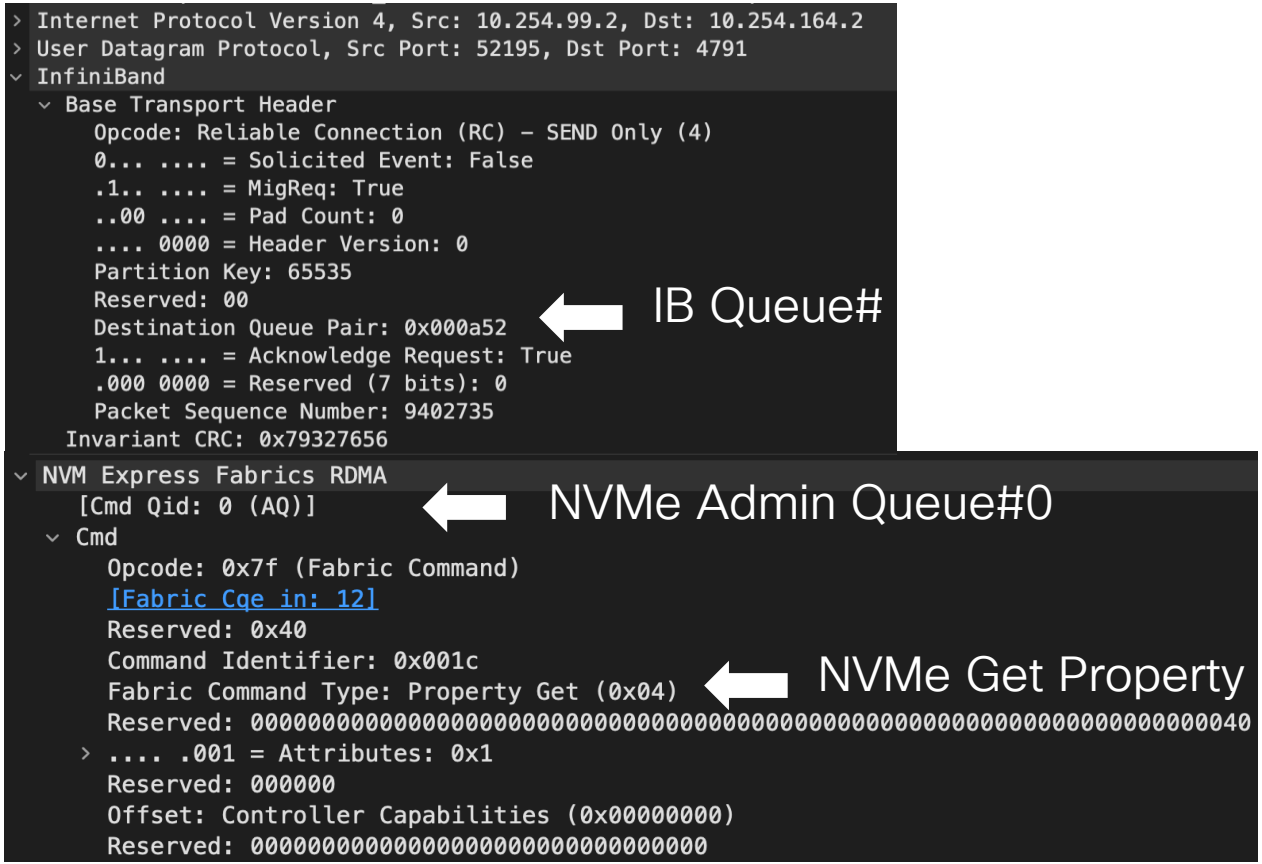




NVMe-RoCEv2 (NVMe-Connect)

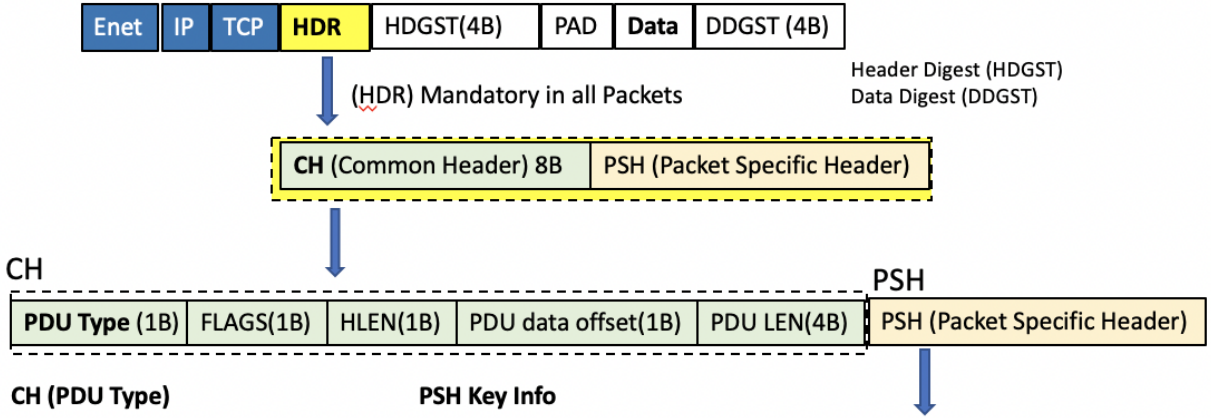


NVMe-RoCEv2 (NVMe Get Property)



NVMe/TCP Traces

NVMe-TCP PDU



CH (PDU Type)	PSH Key Info
00-ICReq -H2C	--- maxR2T Outstanding, HDGST/DDGST enable option
01-ICResp –C2H	--- max allowed H2C Data Length bytes (min 4096), HDGST/DDGST enable confirm
02-H2CTermReq	--- Fatal Error (Invalid Header, Seq. Error, R2T limit exceeded etc..)
03-C2HTermReq	--- Fatal Errors see above
04-CapsuleCmd –H2C	--- **SQE**
05-CapsuleResp	--- **CQE**
06-H2CData	---SQE.CID value of the command capsule PDU associate with the command Data Buffer
07-C2HData	---SQE.CID value of the command capsule PDU associated with the host resident data
09-R2T –C2H	---SQE.CID value of the command capsule PDU associated with the host resident data +R2TO byte offset from the start of the host-resident data +R2TL number of bytes of command data buffer requested by the controller

NVMe-TCP (ICReq)


```
No.     Time        Source          Destination     Protocol Length Info
      5 0.207373    10.254.161.2    10.254.164.2    NVMe/…   198    Initialize Connection Request

▶ Frame 5: 198 bytes on wire (1584 bits), 198 bytes captured (1584 bits) on interface 0
▶ Ethernet II, Src: Cisco_20:42:4f (00:ea:bd:20:42:4f), Dst: Cisco_35:a5:93 (78:0c:f0:35:a5:93)
▶ Internet Protocol Version 4, Src: 10.254.161.2, Dst: 10.254.164.2
▶ Transmission Control Protocol, Src Port: 43946, Dst Port: 4420, Seq: 1, Ack: 1, Len: 128
▼ NVM Express Fabrics TCP
    [Cmd Qid: 0 (AQ)]
    Pdu Type: ICReq (0)
  ▼ Pdu Specific Flags: 0x00
      .... ...0 = PDU Header Digest: Not set
      .... ..0. = PDU Data Digest: Not set
      .... .0.. = PDU Data Last: Not set
      .... 0... = PDU Data Success: Not set
    Pdu Header Length: 128
    Pdu Data Offset: 0
    Packet Length: 128
  ▼ ICReq
      Pdu Version Format: 0
      Maximum r2ts per request: 0
      Host Pdu data alignment: 0
      Digest Types Enabled: 0
```

```
0000  78 0c f0 35 a5 93 00 ea  bd 20 42 4f 08 00 45 00   x··5···· · BO··E·
0010  00 b4 4c 58 40 00 3f 06  93 eb 0a fe a1 02 0a fe   ··LX@·?· ········
0020  a4 02 ab aa 11 44 11 6b  11 b6 01 99 a3 df 80 18   ·····D·k ········
0030  01 f6 61 8c 00 00 01 01  08 0a 3c bb 24 d1 15 a8   ··a····· ··<·$···
0040  ba f4 00 00 80 00 80 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00c0  00 00 00 1e fd a0                                  ······
```

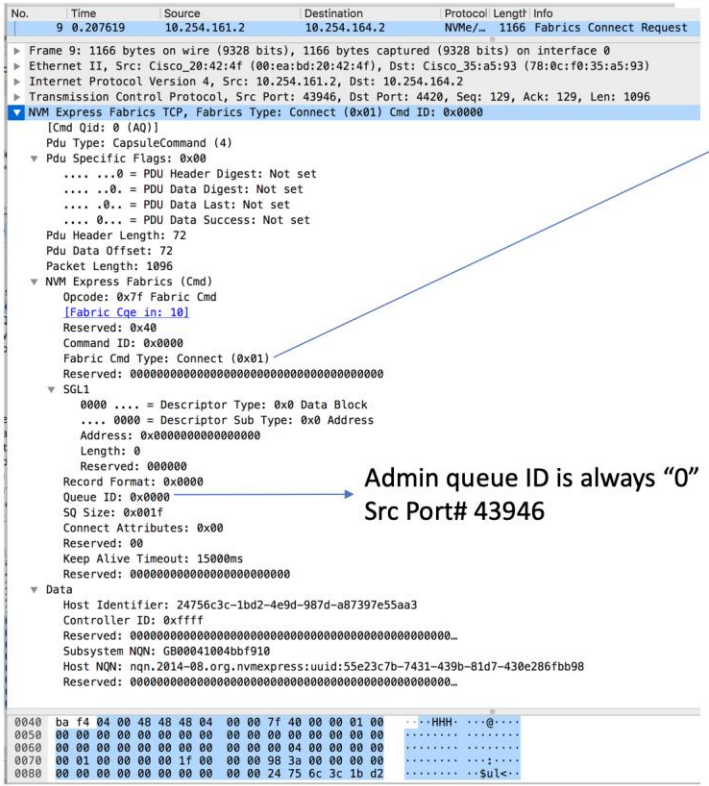
ICReq

Bytes	PDU Section	Description
00	CH	**PDU-Type:** 00h
01		**FLAGS:** Reserved
02		**HLEN:** Fixed length of 128 bytes (80h).
03		**PDO:** Reserved
07:04		**PLEN:** Fixed length of 128 bytes (80h).
09:08	PSH	**PDU Format Version (PFV):** Specifies the format version of NVMe/TCP PDUs. The format of the record specified in this definition shall be cleared to 0h.
10		**Host PDU Data Alignment (HPDA):** Specifies the data alignment for all PDUs transferred from the controller to the host that contain data. This value is 0's based value in units of dwords and must be a value in the range 0 to 31 (e.g., values 0, 1, and 2 correspond to 4 byte, 8 byte, and 12 byte alignment).
11		**DGST:** Host PDU header and Data digest enable options.
15:12		**Maximum Number of Outstanding R2T (MAXR2T):** Specifies the maximum number of outstanding R2T PDUs for a command at any point in time on the connection. This is a 0's based value.
127:16		Reserved

Bits	Definition
7:2	Reserved
1	**DDGST_ENABLE:** If set to '1', the use of data digest is requested by the host for the connection. If cleared to '0', data digest shall not be used for the connection.
0	**HDGST_ENABLE:** If set to '1', the use of header digest is requested by the host for the connection. If cleared to '0', header digest shall not be used for the connection.

NVMe-TCP (Connect Req.)

Host → Connect Request



Capsule Cmd

Enet	IP	TCP	PDU Type (04)	FLAGS(1B)	HLEN(1B)	PDU data offset(1B)	PDU LEN(4B)	(7F) SQE Header	Data	FCS

Connect Command – Submission Queue Entry

Bytes	Description
00	**Opcode (OPC):** Set to 7Fh to indicate a Fabrics command.
01	Reserved
03:02	**Command Identifier (CID):** This field specifies a unique identifier for the command. Refer to the definition in Figure 7.
04	**Fabrics Command Type (FCTYPE):** Set to 01h to indicate a Connect command.
23:05	Reserved
39:24	**SGL Descriptor 1 (SGL1):** This field contains a Transport SGL Data Block descriptor or a Keyed SGL Data Block descriptor that describes the entire data transfer. Refer to section 4.4 of the NVMe Base specification for the definition of SGL descriptors.
41:40	**Record Format (RECFMT):** Specifies the format of the Connect command capsule. The format of the record specified in this definition shall be 0h. If the NVM subsystem does not support the value specified, then a status value of Incompatible Format shall be returned.
43:42	**Queue ID (QID):** Specifies the Queue Identifier for the Admin Queue or I/O Queue to be created. The identifier is used for both the Submission and Completion Queue. The identifier for the Admin Submission Queue and Completion Queue is 0h. The identifier for an I/O Submission and Completion Queue is in the range 1 to 65,534.
45:44	**Submission Queue Size (SQSIZE):** This field indicates the size of the Submission Queue to be created. If the size is 0h or larger than the controller supports, then a status value of Connect Invalid Parameters shall be returned. The maximum size of the Admin Submission Queue is specified in the Discovery Log entry for the NVM subsystem. Refer to section 4.1.3 of the NVMe Base specification. This is a 0's based value.
46	**Connect Attributes (CATTR):** This field indicates attributes for the connection. Bits 7:4 are reserved. Bit 3 indicates support for deleting individual I/O Queues. If this bit is set to '1', then the host supports the deletion of individual I/O Queues. If this bit is cleared to '0', then the host does not support the deletion of individual I/O Queues. Bit 2 if set to '1', then the host is requesting that SQ flow control be disabled. If cleared to '0', then SQ flow control shall not be disabled. Bits 1:0 indicate the priority class to use for commands within this Submission Queue. This field is only used when the weighted round robin with urgent priority class is the arbitration mechanism selected, the field is ignored if weighted round robin with urgent priority class is not used. Refer to section 4.13 of the NVMe Base specification. This field is only valid for I/O Queues. It shall be set to 00b for Admin Queue connections.
47	Reserved

queue QoS →

Value	Definition
00b	Urgent
01b	High
10b	Medium
11b	Low

NVMe-TCP (Connect Req. Data)

Host → Connect Request (Data)

```
No.  Time      Source          Destination     Protocol Length Info
   9 0.207619  10.254.161.2    10.254.164.2    NVMe/…   1166   Fabrics Connect Request

▶ Frame 9: 1166 bytes on wire (9328 bits), 1166 bytes captured (9328 bits) on interface 0
▶ Ethernet II, Src: Cisco_20:42:4f (00:ea:bd:20:42:4f), Dst: Cisco_35:a5:93 (78:0c:f0:35:a5:93)
▶ Internet Protocol Version 4, Src: 10.254.161.2, Dst: 10.254.164.2
▶ Transmission Control Protocol, Src Port: 43946, Dst Port: 4420, Seq: 129, Ack: 129, Len: 1096
▼ NVM Express Fabrics TCP, Fabrics Type: Connect (0x01) Cmd ID: 0x0000
    [Cmd Qid: 0 (AQ)]
    Pdu Type: CapsuleCommand (4)
  ▼ Pdu Specific Flags: 0x00
      .... ...0 = PDU Header Digest: Not set
      .... ..0. = PDU Data Digest: Not set
      .... .0.. = PDU Data Last: Not set
      .... 0... = PDU Data Success: Not set
    Pdu Header Length: 72
    Pdu Data Offset: 72
    Packet Length: 1096
  ▼ NVM Express Fabrics (Cmd)
      Opcode: 0x7f Fabric Cmd
      [Fabric Cqe in: 10]
      Reserved: 0x40
      Command ID: 0x0000
      Fabric Cmd Type: Connect (0x01)
      Reserved: 0000000000000000000000000000000000000000
    ▼ SGL1
        0000 .... = Descriptor Type: 0x0 Data Block
        .... 0000 = Descriptor Sub Type: 0x0 Address
        Address: 0x0000000000000000
        Length: 0
        Reserved: 000000
      Record Format: 0x0000
      Queue ID: 0x0000
      SQ Size: 0x001f
      Connect Attributes: 0x00
      Reserved: 00
      Keep Alive Timeout: 15000ms
      Reserved: 00000000000000000000
  ▼ Data
      Host Identifier: 24756c3c-1bd2-4e9d-987d-a87397e55aa3
      Controller ID: 0xffff
      Reserved: 0000000000000000000000000000000000000000000000000000…
      Subsystem NQN: GB00041004bbf910
      Host NQN: nqn.2014-08.org.nvmexpress:uuid:55e23c7b-7431-439b-81d7-430e286fbb98
      Reserved: 0000000000000000000000000000000000000000000000000000…

0040  ba f4 04 00 48 48 48 04  00 00 7f 40 00 00 01 00   ····HHH· ···@····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 04 00 00 00 00   ········ ········
0070  00 01 00 00 00 00 1f 00  00 00 98 3a 00 00 00 00   ········ ···:····
0080  00 00 00 00 00 00 00 00  00 00 24 75 6c 3c 1b d2   ········ ··$ul<··
```

Capsule Cmd

Enet	IP	TCP	PDU Type (04)	FLAGS(1B)	HLEN(1B)	PDU data offset(1B)	PDU LEN(4B)	(7F) SQE Header	Data	FCS

Connect Command – Data

Bytes	Description
15:00	**Host Identifier (HOSTID):** This field has the same definition as the Host Identifier defined in section 5.21.1.26 (Host Identifier) of the the NVMe Base specification. The controller shall set the Host Identifer Feature to this value.
17:16	**Controller ID (CNTLID):** Specifies the controller ID requested. This field corresponds to the Controller ID (CNTLID) value returned in the Identify Controller data structure for a particular controller. If the NVM subsystem uses the dynamic controller model, then the value shall be FFFFh for the Admin Queue and any available controller may be returned. If the NVM subsystem uses the static controller model and the value is FFFEh for the Admin Queue, then any available controller may be returned.
255:18	Reserved
511:256	**NVM Subsystem NVMe Qualified Name (SUBNQN):** NVMe Qualified Name (NQN) that uniquely identifies the NVM subsystem. Refer to section 7.9 (NVMe Qualified Names) of the NVMe Base specification.
767:512	**Host NVMe Qualified Name (HOSTNQN):** NVMe Qualified Name (NQN) that uniquely identifies the host. Refer to section 7.9 (NVMe Qualified Names) of the NVMe Base specification.
1023:768	Reserved

NVMe-TCP (Property Get)

Host → Property Get (AQ0/43946)

```
No.     Time        Source          Destination     Protocol Length Info
    12  0.230180    10.254.161.2    10.254.164.2    NVMe/…      142 Fabrics Property Get Request

▶ Frame 12: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0
▶ Ethernet II, Src: Cisco_20:42:4f (00:ea:bd:20:42:4f), Dst: Cisco_35:a5:93 (78:0c:f0:35:a5:93)
▶ Internet Protocol Version 4, Src: 10.254.161.2, Dst: 10.254.164.2
▶ Transmission Control Protocol, Src Port: 43946, Dst Port: 4420, Seq: 1225, Ack: 153, Len: 72
▼ NVM Express Fabrics TCP, Fabrics Type: Property Get (0x04) Cmd ID: 0x000e
    [Cmd Qid: 0 (AQ)]
    Pdu Type: CapsuleCommand (4)
  ▼ Pdu Specific Flags: 0x00
      .... ...0 = PDU Header Digest: Not set
      .... ..0. = PDU Data Digest: Not set
      .... .0.. = PDU Data Last: Not set
      .... 0... = PDU Data Success: Not set
    Pdu Header Length: 72
    Pdu Data Offset: 0
    Packet Length: 72
  ▼ NVM Express Fabrics (Cmd)
      Opcode: 0x7f Fabric Cmd
      [Fabric Cqe in: 13]
      Reserved: 0x40
      Command ID: 0x000e
      Fabric Cmd Type: Property Get (0x04)
      Reserved: 0000000000000000000000000000000000000000000000000…
      0000 0... = Reserved: 0x00
      .... .001 = Property Size: 0x18 bytes
      Reserved: 000000
      Offset: 0x00000000Controller Capabilities
      Reserved: 00000000000000000000000000000000
```

```
0000  78 0c f0 35 a5 93 00 ea  bd 20 42 4f 08 00 45 00   x··5····· ·BO··E·
0010  00 7c 4c 5c 40 00 3f 06  94 1f 0a fe a1 02 0a fe   ·|L\@·?· ········
0020  a4 02 ab aa 11 44 11 6b  16 7e 01 99 a4 77 80 18   ·····D·k ·~···w··
0030  01 f5 35 9e 00 00 01 01  08 0a 3c bb 24 e8 15 a8   ··5····· ··<·$···
0040  bb 0a 04 00 48 00 48 00  00 00 7f 40 0e 00 04 00   ····H·H· ···@····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 5a 01 00 00 00 00 00  00 00 00 00 00 00 00 00   ·Z····· ········
0080  00 00 00 00 00 00 00 00  00 00 12 b5 f6 db        ········ ······
```

Property Get Command

Bytes	Description
00	**Opcode (OPC):** Set to 7Fh to indicate a Fabrics command.
01	Reserved
03:02	**Command Identifier (CID):** This field specifies a unique identifier for the command. Refer to the definition in Figure 7.
04	**Fabrics Command Type (FCTYPE):** Set to 04h to indicate a Property Get command.
39:05	Reserved
40	**Attributes (ATTRIB):** Specifies attributes for the Property Get command. Bits 7:3 are reserved. Bits 2:0 specifies the size of the property to return. Valid values are shown in the table below.

Value	Definition
000b	4 bytes
001b	8 bytes

Bytes	Description
43:41	Reserved
47:44	**Offset (OFST):** Specifies the offset to the property to get. Refer to section 3.6.1.
63:48	Reserved

NVMe-TCP (Property Set)

Host → Property Set (AQ0/43946)

No.	Time	Source	Destination	Protocol	Length	Info
15	0.230230	10.254.161.2	10.254.164.2	NVMe/…	142	Fabrics Property Set Request

```
▶ Frame 15: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0
▶ Ethernet II, Src: Cisco_20:42:4f (00:ea:bd:20:42:4f), Dst: Cisco_35:a5:93 (78:0c:f0:35:a5:93)
▶ Internet Protocol Version 4, Src: 10.254.161.2, Dst: 10.254.164.2
▶ Transmission Control Protocol, Src Port: 43946, Dst Port: 4420, Seq: 1297, Ack: 177, Len: 72
▼ NVM Express Fabrics TCP, Fabrics Type: Property Set (0x00) Cmd ID: 0x000f
    [Cmd Qid: 0 (AQ)]
    Pdu Type: CapsuleCommand (4)
  ▼ Pdu Specific Flags: 0x00
      .... ...0 = PDU Header Digest: Not set
      .... ..0. = PDU Data Digest: Not set
      .... .0.. = PDU Data Last: Not set
      .... 0... = PDU Data Success: Not set
    Pdu Header Length: 72
    Pdu Data Offset: 0
    Packet Length: 72
  ▼ NVM Express Fabrics (Cmd)
      Opcode: 0x7f Fabric Cmd
      [Fabric Cqe in: 16]
      Reserved: 0x40
      Command ID: 0x000f
      Fabric Cmd Type: Property Set (0x00)
      Reserved: 0000000000000000000000000000000000000000000000000…
      0000 0... = Reserved: 0x00
      .... .000 = Property Size: 0x04 bytes
      Reserved: 000000
      Offset: 0x00000014Controller Configuration
      Value: 0x00460001
      Reserved: 0x00000000
      Reserved: 0000000000000000
```

```
0000  78 0c f0 35 a5 93 00 ea  bd 20 42 4f 08 00 45 00   x··5···· · BO··E·
0010  00 7c 4c 5e 40 00 3f 06  94 1d 0a fe a1 02 0a fe   ·|L^@·?· ········
0020  a4 02 ab aa 11 44 11 6b  16 c6 01 99 a4 8f 80 18   ·····D·k ········
0030  01 f5 de 3d 00 00 01 01  08 0a 3c bb 24 e8 15 a8   ···=···· ··<·$···
0040  bb 0a 04 00 48 00 48 00  00 00 7f 40 0f 00 00 00   ····H·H· ···@····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 5a 00 00 00 00 14 00  00 00 01 00 46 00 00 00   ·Z······ ····F···
0080  00 00 00 00 00 00 00 00  00 00 cb 7d 69 7f         ········ ···}i·
```

Property Set Command

Bytes	Description
00	**Opcode (OPC):** Set to 7Fh to indicate a Fabrics command.
01	Reserved
03:02	**Command Identifier (CID):** This field specifies a unique identifier for the command. Refer to the definition in Figure 7.
04	**Fabrics Command Type (FCTYPE):** Set to 00h to indicate a Property Set command.
39:05	Reserved
40	**Attributes (ATTRIB):** Specifies attributes for the Property Set command. Bits 7:3 are reserved. Bits 2:0 specifies the size of the property to update. Valid values are shown in the table below. Value / Definition: 000b – 4 bytes 001b – 8 bytes 010b to 111b – Reserved
43:41	Reserved
47:44	**Offset (OFST):** Specifies the offset to the property to set. Refer to section 3.6.1.
55:48	**Value (VALUE):** Specifies the value used to update the property. If the size of the property is four bytes, then the value is specified in bytes 51:48 and bytes 55:52 are reserved.
63:56	Reserved

NVMe/FC Traces

NVMe-FC Command IU

R_CTL (Routing)

To Target Port	To Initiator Port
06 NVMe_CMND	05 NVMe_XFER_RDY
01 NVMe_DATA	01 NVMe_DATA
03 NVMe_CONF	07 NVMe_RSP
09 NVMe_SR	08 NVMe_ERSP
	0A NVMe_SR_RSP

FC ID
28 NVMe-FC

Category
0001b Admin SQE
0xxxb I/O SQE

Flag
0 Write, 1 Read

Connection ID
Host queues mapping to the Controller's NVMe queues

Type
28 NVMe Dataset (Link Services)
08 FCP Dataset
(if Type=08, first byte = FD for NVMe-FC)

DPS
Data Protection Type Setting

LBADS
LBA Data Size

MS
Meta Data Size

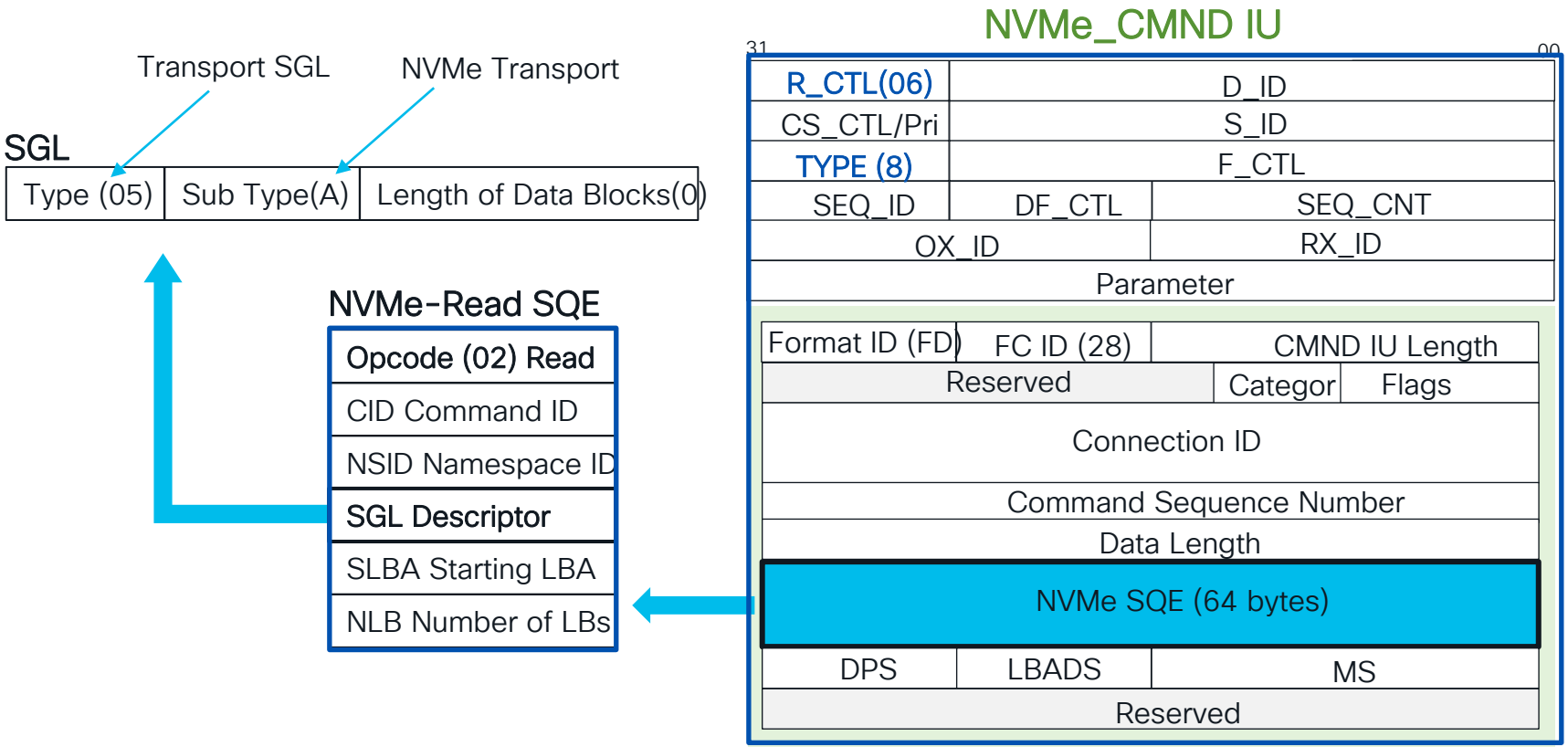
0001b for Admin queue
1xxxb for I/O queue
xxx = CSS

write
read(bit)

NVMe_CMND IU

31		00
R_CTL(06)		D_ID
CS_CTL/Pri		S_ID
TYPE (8)		F_CTL
SEQ_ID	DF_CTL	SEQ_CNT
OX_ID		RX_ID
Parameter		
Format ID (FD)	FC ID (28)	CMND IU Length
Reserved		Categor / Flags
Connection ID		
Command Sequence Number		
Data Length		
NVMe SQE (64 bytes)		
DPS	LBADS	MS
Reserved		

NVMe-FC Read Command IU



NVMe-FC Data Transfer

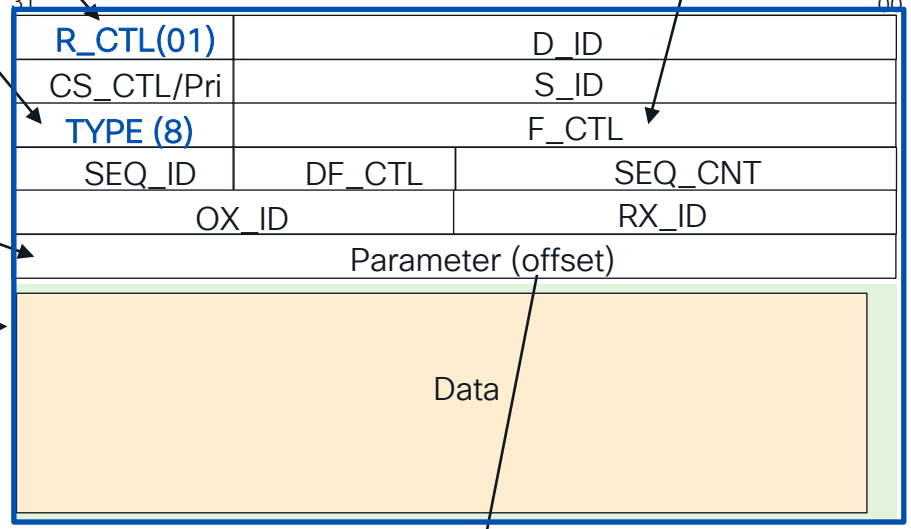
NVMe-Data

FCP Dataset

Bit set to 1 indicating Offset present

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

Port(1,1,1)	FCP	FC4SData		FC4SData; SCSI FCP; Offset = 0x00000000; Len = 0x0800;	2084	023E
Port(1,1,1)	FCP	FC4SData		FC4SData; SCSI FCP; Offset = 0x00000800; Len = 0x0800;	2084	023E
Port(1,1,1)	FCP	FC4ExtStatus		Success;	68	023E
Port(1,1,2)	FCP		FC4Cmd	Read; NSID = 0x00000004; LBA = 0x00000000; NbBlocks =	132	023F
Port(1,1,1)	FCP	FC4SData		FC4SData; SCSI FCP; Offset = 0x00000000; Len = 0x0800;	2084	023F
Port(1,1,1)	FCP	FC4SData		FC4SData; SCSI FCP; Offset = 0x00000800; Len = 0x0800;	2084	023F
Port(1,1,1)	FCP	FC4Status		Good Status;	48	023F
Port(1,1,2)	FCP		FC4Cmd	Read; NSID = 0x00000004; LBA = 0x00000008; NbBlocks =	132	0240
Port(1,1,1)	FCP	FC4SData		FC4SData; SCSI FCP; Offset = 0x00000000; Len = 0x0800;	2084	0240
Port(1,1,1)	FCP	FC4SData		FC4SData; SCSI FCP; Offset = 0x00000800; Len = 0x0800;	2084	0240
Port(1,1,1)	FCP	FC4Status		Good Status;	48	0240

NVMe-FC (PRLI -Process Log In)

```
SOF = SOFi3;
RCtl = ExtLinkReq; D_Id = 0xAE00C1;
CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00E0;
Type = EX_LNK_SRV; F_Ctl [Exchange Context = Originator; First_Sequence; End_Sequence; Sequence Initiative = Transfer Sequence Initiative];
SEQ_Id = 0x00; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
OX_Id = 0x0203; RX_Id = 0xFFFF;
PARA = 0x00000000;
Command Code = PRLI  (Interesting Event Found); Page Length = 16  Bytes; Payload Length = 20  Bytes;
Type Code = SCSI FCP; Flags [Established Image Pair];
Originator Process_Associator = 0x00000000;
Responder Process_Associator = 0x00000000;
Service Parameters [Rec_Support; Task Retry Identification Requested; Retry; Confirm Completion Allowed; Initiator Function; RXferRdyDisabled];
CRC = 0x028125FE    (Correct);
EOF = EOFt;
```

PRLI

Service Parameter
(Initiator Function = NVMe-FC/reply)

NVMe-FC (Create Association)

```
SOF = SOFi3;
RCtl = FC4LinkUCtl; D_Id = 0xAE00C3;
CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00E0;
Type = FC-NVMe; F_Ctl [Exchange Context = Originator; First_Sequence; End_Sequence; Sequence Initiative = Transfer Sequence Initiative];
SEQ_Id = 0x00; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
OX_Id = 0x021D; RX_Id = 0xFFFF;
PARA = 0x00000000;
Command Code = NVMe Create Association;
Descriptor list length = 1016  Bytes;
Descriptor tag = NVMe Create Association;
Descriptor length = 1008  Bytes;
NVMe_ERSP Ratio = 0x0008;

Controller ID = 0xFFFF; Admin Submission Queue Size = 0x001F;

Host Identifier = ED9D0705 6B4F425D A99B99E8 FF67FC80;

Host NVMe Qualified Name = nqn.2014-08.org.nvmexpress:uuid:290ecc27-d30e-4f08-9a73-474e3802c9d8;
```

Command
(NVMe Create Association)

Controller ID
(Dynamic)

Admin Queue
depth

Host NQN

NVMe-FC (Accept Create Association)

```
SOF = SOFi3;
RCtl = FC4LinkSCtl; D_Id = 0xAE00E0;
CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00C3;
Type = FC-NVMe; F_Ctl [Exchange Context = Responder; Last_Sequence; End_Sequence];
SEQ_Id = 0x00; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
OX_Id = 0x021D; RX_Id = 0x00D1;
PARA = 0x00000000;
Command Code = Accept;
Descriptor list length = 48  Bytes;
Descriptor tag = NVMe Link Service Request Information;
Descriptor length = 8  Bytes;
Accepted Command Code = NVMe Create Association;

Descriptor tag = NVMe Association Identifier;
Descriptor length = 8  Bytes;
NVMe Association Identifier = 0x5FBF79822FA30000;

Descriptor tag = NVMe Connection Identifier;
Descriptor length = 8  Bytes;
NVMe Connection Identifier = 0x5FBF79822FA30000;

CRC = 0x268BD28B    (Correct);
EOF = EOFt;
```

Accept

NVMe Association ID

NVMe Connection ID

NVMe-FC (Connect)

```
SOF = SOFi3;
RCtl = FC4Cmd; D_Id = 0xAE00C3;
CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00E0;
Type = SCSI FCP; F_Ctl [Exchange Context = Originator; First_Sequence; End_Sequence; Sequence Initiative = Transfer Sequence Initiative];
SEQ_Id = 0x01; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
OX_Id = 0x020E; RX_Id = 0xFFFF;
PARA = 0x00000000;
Differentiator = FC-NVMe Cmd IU; CMD IU Length = 24  Words;
Flags [Write = ->Data];
NVMe Connection Identifier = 0xE5B420ADBB500000;

Command Sequence Number = 0x00000001;
Data Length = 0x00000400;
Opcode = Fabrics Cmd; Reserved = 0x40  (Unexpected Value Found); CID = 0x0000;
Fabrics Cmd = Connect;



SGL Entry 1 [

Length = 0x00000400  Bytes;
SGL Descriptor Type = Transport SGL Data Block descriptor; SGL Descriptor SubType = 0xA Reserved  (Unexpected Value Found)];
Record Format = NVMe 1.2.1; Queue ID = 0x0000;
Subm Queue Size = 32; Connect Attributes [Priority Class = Urgent];
Keep Alive Timeout = 0  ms;
```

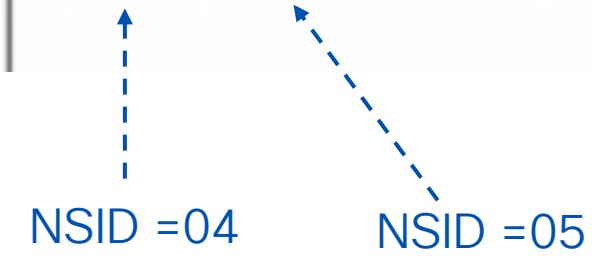
Fabric Command = Connect

default queue
size = 32

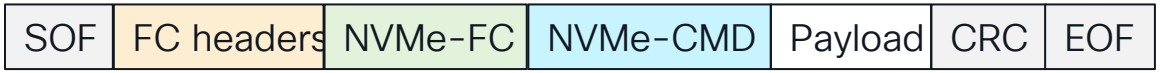
Queue ID = 0 (Admin)

NVMe-FC (Reply Identify Active Name Space List)

```
SOF = SOFi3;
RCtl = FC4SData; D_Id = 0xAE00E0;
CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00C3;
Type = SCSI FCP; F_Ctl [Exchange Context = Responder; RO];
SEQ_Id = 0x81; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
OX_Id = 0x0239; RX_Id = 0x0353;
PARA = 0x00000000; Pld bytes = 0x0800;
Pld = 04000000 05000000 00000000 00000000 00000000 00000000 00000000 00000000...;
```



NVMe-FC (Read command)



Index	Hex	Interpretation
SOF 000000	FB B5 56 56	SOF = SOFi3;
FCH 000000	06 AE 00 C3	RCtl = FC4Cmd; D_Id = 0xAE00C3;
FCH 000001	00 AE 00 E0	CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00E0;
FCH 000002	08 29 00 00	Type = SCSI FCP; F_Ctl [Exchange Context = Originator; First_Sequence; End_Sequence; Sequence Initiative = Transfer Sequence Initiative];
FCH 000003	01 00 00 00	SEQ_Id = 0x01; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
FCH 000004	02 3F FF FF	OX_Id = 0x023F; RX_Id = 0xFFFF;
FCH 000005	00 00 00 00	PARA = 0x00000000;
FCP 000000	FD 28 00 18	Differentiator = FC-NVMe Cmd IU; CMD IU Length = 24 Words;
FCP 000001	00 00 00 02	Flags [Read = <-Data];
FCP 000002	5F BF 79 82	NVMe Connection Identifier = 0x5FBF79822FA30002;
FCP 000003	2F A3 00 02	
FCP 000004	00 00 00 02	Command Sequence Number = 0x00000002;
FCP 000005	00 00 10 00	Data Length = 0x00001000;
NVMe 00000	02 40 51 00	Opcode = Read; PRP or SGL = SGL; CID = 0x0051;
NVMe 00001	04 00 00 00	NSID = 0x00000004;
NVMe 00002	00 00 00 00	
NVMe 00003	00 00 00 00	
NVMe 00004	00 00 00 00	Metadata SGL Segment Pointer = 0x00000000;
NVMe 00005	00 00 00 00	
NVMe 00006	00 00 00 00	SGL Entry 1 [
NVMe 00007	00 00 00 00	
NVMe 00008	00 10 00 00	Length = 0x00001000 Bytes;
NVMe 00009	00 00 00 5A	SGL Descriptor Type = Transport SGL Data Block descriptor; SGL Descriptor SubType = 0xA Reserved (Unexpected Value Found)];
NVMe 00010	00 00 00 00	Starting LBA = 0x00000000;
NVMe 00011	00 00 00 00	
NVMe 00012	07 00 00 00	Number of Logical Blocks = 0x08; PRInfoAction = Pass;
NVMe 00013	00 00 00 00	Dataset Management [Access Latency = None; Access Frequency = Unknown];
NVMe 00014	00 00 00 00	Expected Initial Block Ref Tag = 0x00000000;
NVMe 00015	00 00 00 00	Expected Block App Tag = 0x0000; Expected Block App Tag Mask = 0x0000;
FCP 000000	00 00 00 00	
FCP 000001	00 00 00 00	
End 000000	6E 0E 7A 10	CRC = 0x6E0E7A10 (Correct);
End 000001	95 75 75 FD	EOF = EOFt;

NVMe-CMD “Read”

Connection-ID

NSID

SLB

NLB

NVMe-FC (Read NSID)

```
SOF = SOFi3;
RCtl = FC4Cmd; D_Id = 0xAE00C3;
CS_CTL = 0x00 [DSCP = 0x00]; S_Id = 0xAE00E0;
Type = SCSI FCP; F_Ctl [Exchange Context = Originator; First_Sequence; End_Sequence; Sequence Initiative = Transfer Sequence Initiative];
SEQ_Id = 0x01; DF_Ctl = 0x00; SEQ_Cnt = 0x0000;
OX_Id = 0x0240; RX_Id = 0xFFFF;
PARA = 0x00000000;
Differentiator = FC-NVMe Cmd IU; CMD IU Length = 24  Words;
Flags [Read = <-Data];
NVMe Connection Identifier = 0x5FBF79822FA30002;

Command Sequence Number = 0x00000003;
Data Length = 0x00001000;
Opcode = Read; PRP or SGL = SGL; CID = 0x0052;
NSID = 0x00000004;


Metadata SGL Segment Pointer = 0x00000000;

SGL Entry 1 [

Length = 0x00001000  Bytes;
SGL Descriptor Type = Transport SGL Data Block descriptor; SGL Descriptor SubType = 0xA Reserved  (Unexpected Value Found)];
Starting LBA = 0x00000008;

Number of Logical Blocks = 0x08; PRInfoAction = Pass;
Dataset Management [Access Latency = None; Access Frequency = Unknown];
Expected Initial Block Ref Tag = 0x00000000;
Expected Block App Tag = 0x0000; Expected Block App Tag Mask = 0x0000;


CRC = 0x263126B3   (Correct);
EOF = EOFt;
```

Read = SGL

NSID

Starting LBA

Number of Logical Blocks

Agenda

1-Why NVMe?

2-NVMe Architecture (PCIe)

3-NVMe Transport Options (FC, TCP, RoCEv2)

4-NVMe Datacenter Design

5-Additional Information

- -NVMe Upcoming Features
- -NVMe Additional Information
- -NVMe Flow Traces

Check Out Other Storage Related Sessions

Session ID	Title
BRKDCN-3812	Tues 2.30pm Level 2, Lagoon H -Kamal Bakshi Dos and Don'ts of Deploying NVMe Over Fabrics (this session)
BRKDCN-3645	Wed 10.30am Level 2, Lagoon H -Paresh Gupta SAN Insights - Real-time and always-on NVMe visibility at scale
PSODCN-2355	Wed 2.00pm Level 3, South Seas H -Kiran Ranabhor Real-time NVMe and SCSI visibility using Cisco SAN Analytics
BRKDCN-2489	Wed 4:00pm Level 3, South Seas D -Nemanja Kamenica IP Fabric for Storage Networks Best Practice and Design
BRKDCN-3241	Thurs 8.00am Level 2, Lagoon B -Paresh Gupta Detecting, Alerting, Identifying and Proactively Preventing SAN Congestion

Technical Session Surveys

- Attendees who fill out a minimum of four session surveys and the overall event survey will get Cisco Live branded socks!
- Attendees will also earn 100 points in the Cisco Live Game for every survey completed.
- These points help you get on the leaderboard and increase your chances of winning daily and grand prizes.

Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. www.cisco.com/go/certs

Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.

Learn

Cisco U.
IT learning hub that guides teams and learners toward their goals

Cisco Digital Learning
Subscription-based product, technology, and certification training

Cisco Modeling Labs
Network simulation platform for design, testing, and troubleshooting

Cisco Learning Network
Resource community portal for certifications and learning

Train

Cisco Training Bootcamps
Intensive team & individual automation and technology training programs

Cisco Learning Partner Program
Authorized training partners supporting Cisco technology and career certifications

Cisco Instructor-led and Virtual Instructor-led training
Accelerated curriculum of product, technology, and certification courses

Certify

Cisco Certifications and Specialist Certifications
Award-winning certification program empowers students and IT Professionals to advance their technical careers

Cisco Guided Study Groups
180-day certification prep program with learning and support

Cisco Continuing Education Program
Recertification training options for Cisco certified individuals

Here at the event? Visit us at **The Learning and Certifications lounge at the World of Solutions**

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

cisco Live!


cisco
The bridge to possible
Thank you
cisco Live!


CISCO Live!
ALL IN