SNIA NSF | NETWORKING STORAGE

NVMe/TCP: Performance, Deployment, and Automation

Live Webinar

July 19, 2023

10:00 am PT / 1:00 pm ET

Today's Presenters

Erik Smith
Distinguished Engineer
Dell Technologies

Christine McMonigal
Director of Hyperconverged Marketing
Intel

The SNIA Community

200 Corporations, universities, startups, and individuals	**2,500** Active contributing members	**50,000** Worldwide IT end users and professionals

Technologies We Cover

Ethernet, Fibre Channel, InfiniBand®

iSCSI, NVMe-oF™, NFS, SMB

Virtualized, HCI, Software-defined Storage

Storage Protocols (block, file, object)

Securing Data

SNIA NSF | NETWORKING STORAGE

SNIA NSF | NETWORKING STORAGE

SNIA Legal Notice

- The material contained in this presentation is copyrighted by SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Today's Agenda

- Overview
- Performance
- Deployment Considerations
- Automation
- Q&A

Overview

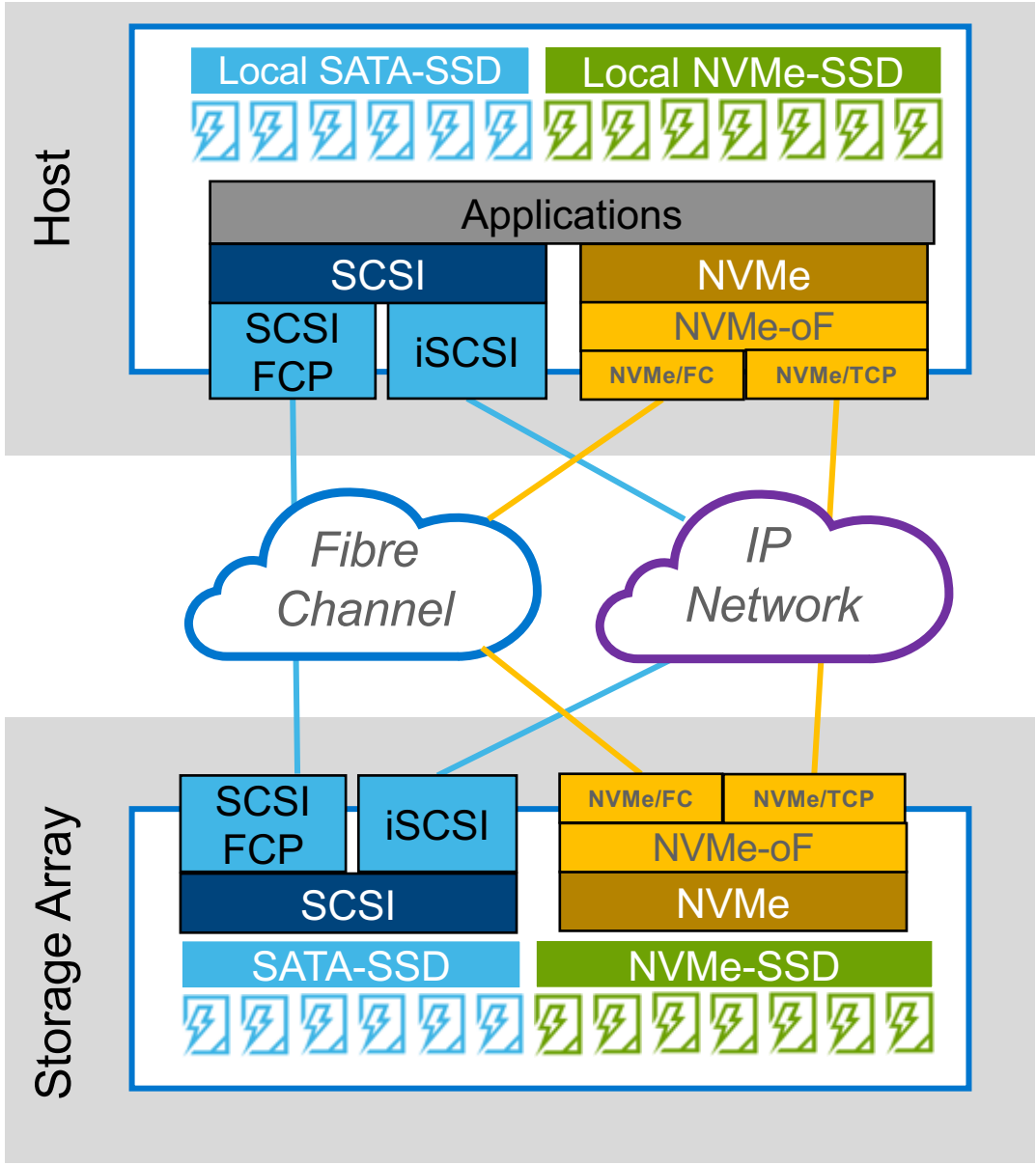
Additional Information about this Presentation

- This presentation provides practical deployment considerations for NVMe™ /TCP and will not attempt to cover all aspects of NVMe/TCP.
- If a detailed description of the protocol is required, additional resources are available:
  - What NVMe™/TCP Means for Networked Storage – Posted 4 years ago and is still 100% accurate. This video, presented by Sagi Grimberg and J Metz, is the BEST way to get an overall understanding of the NVMe/TCP protocol.
  - SNIA NSF – Discovery Automation for NVMe IP-Based SANs – A high level overview of Discovery Automation related enhancements that have been added to NVMe/TCP since it was first introduced
  - SNIA SDC – Discovery Protocol deep dive – A more detailed look at Discovery Automation protocols.
  - Security of Data on NVMe over Fabrics, The Armored Truck Way – A detailed overview of NVMe/TCP security considerations.

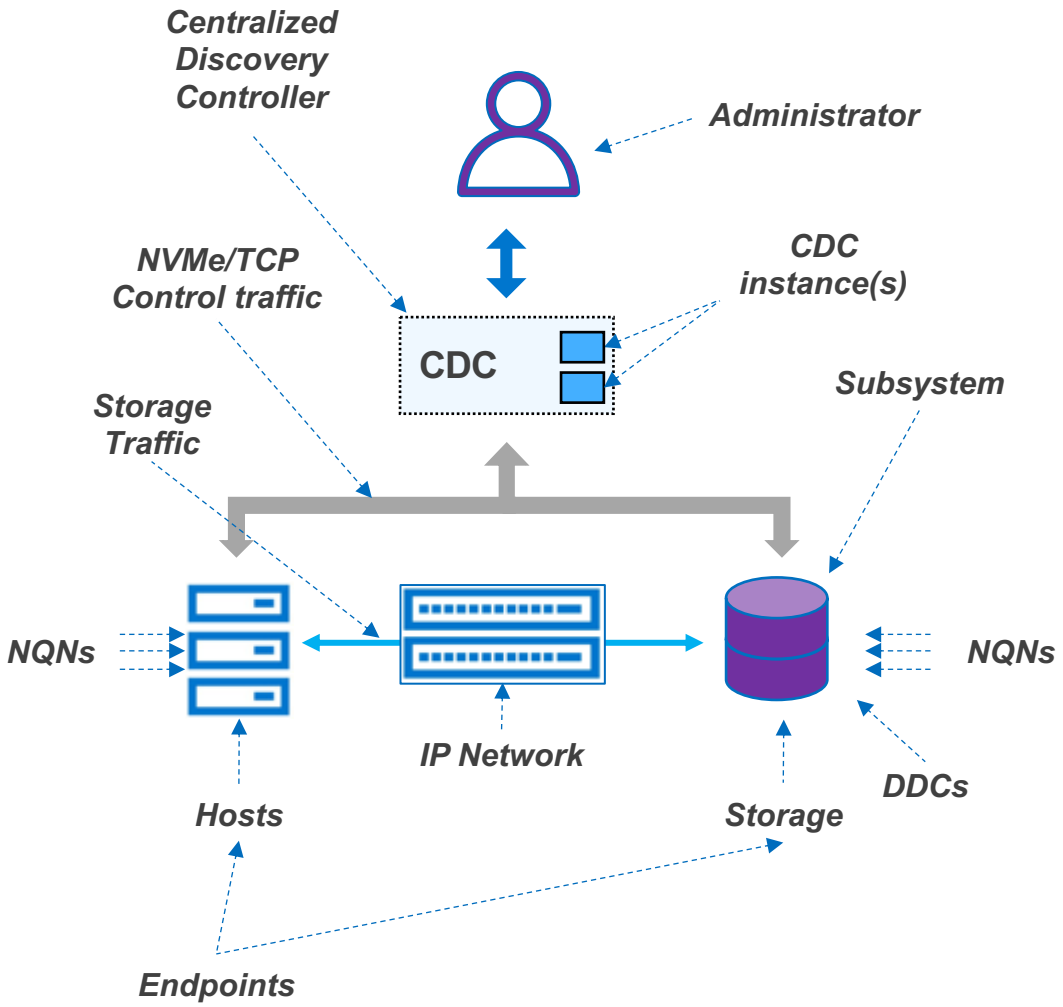
SAN Evolution

From SCSI to NVMe

- Applications running on a host that are accessing external array-based storage via either FC or iSCSI.
- NVMe Drives were first introduced on the host in 2015 and were used mainly for caching and boot drives
- NVMe-SSDs improve storage array performance locally but using the SCSI protocol can add significant latency.
- NVMe-oF™ can run over either Fibre Channel or Ethernet.



NVMe IP-Based SAN terminology



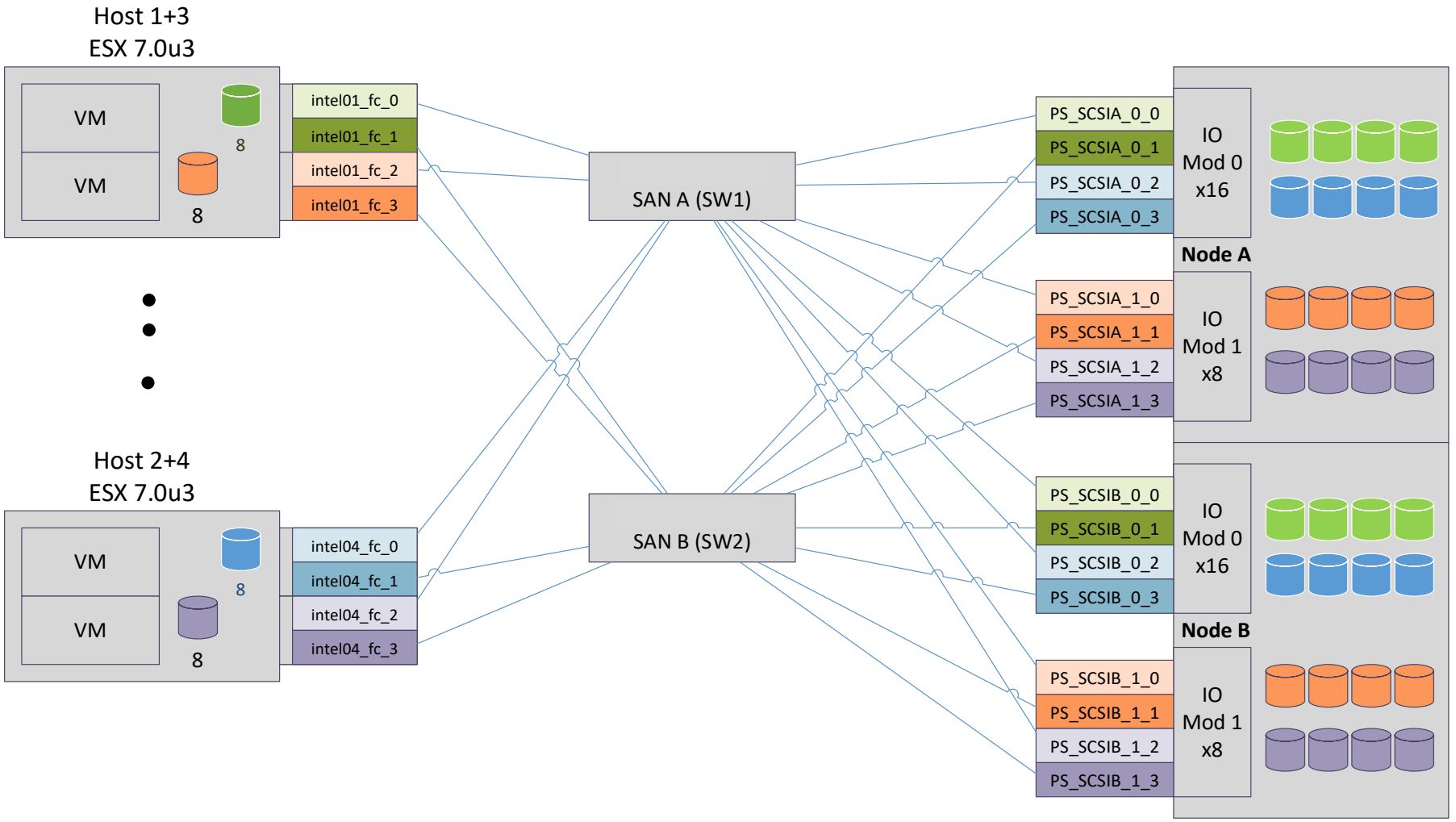
- **Endpoints: On hosts and storage systems**
  - Identified by NVMe Qualified Name (NQN) and IP Address
- **IP Network:**
  - Most modern switches (e.g., 25GbE capable and above) will work.
- **Subsystem**: Storage array, analogous to SCSI target
  - Identified by NVMe Qualified Names (NQN). NQN has a similar function to WWNN (Worldwide unique Node Name) in FC, and IQN in iSCSI.
- **CDC**: Centralized Discovery Controller Instances
  - Each CDC instance provides a Discovery controller for Endpoints that are taking part in a particular NVMe IP-based SAN instance.
- **DDC**: Direct Discovery Controllers
  - An NVMe Discovery controller that resides on Subsystems
  - Hosts could connect directly to storage via the DDC, but would lose the advantages of Centralization

Performance

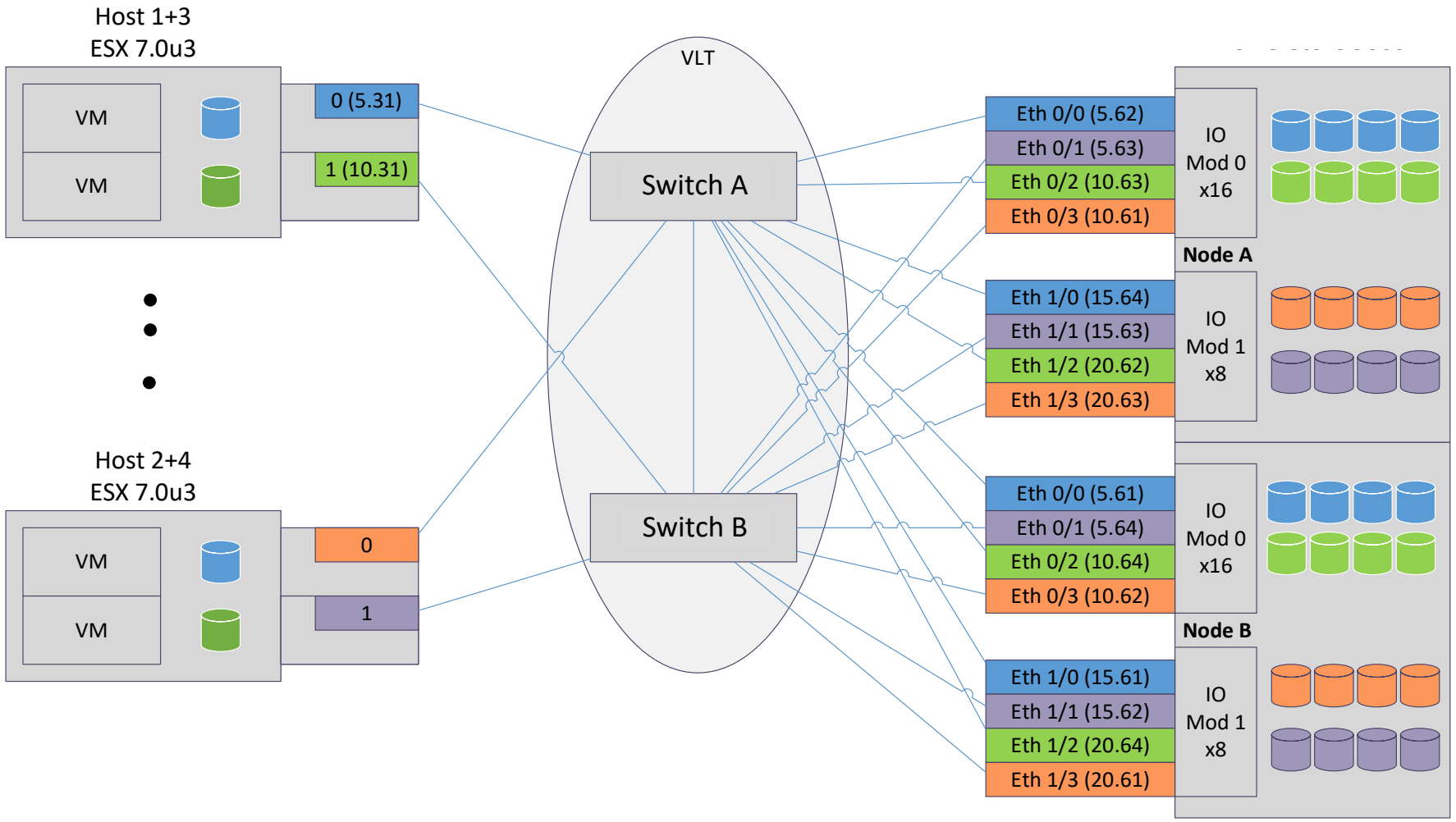
Performance Testing Notes

- NVMe/FC @ 32GFC provided the highest IOPS, lowest latency and lowest CPU utilization.
- The performance related metrics were captured during testing at Dell.
- Additional information about the testing and how you can reproduce our results are available in the NVMe Transport Performance Comparison whitepaper.

Performance Test Configuration - FC Based

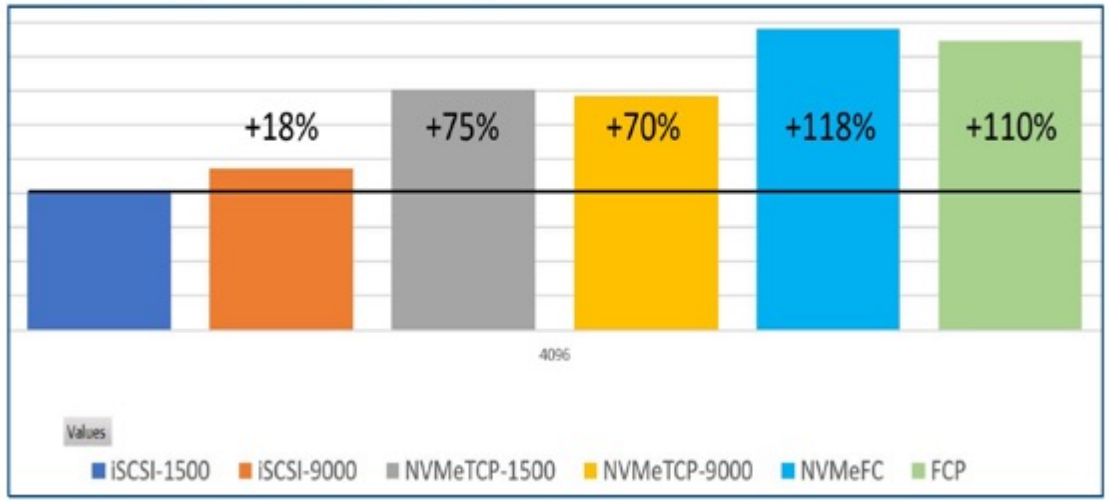


Performance Test Configuration - Ethernet Based

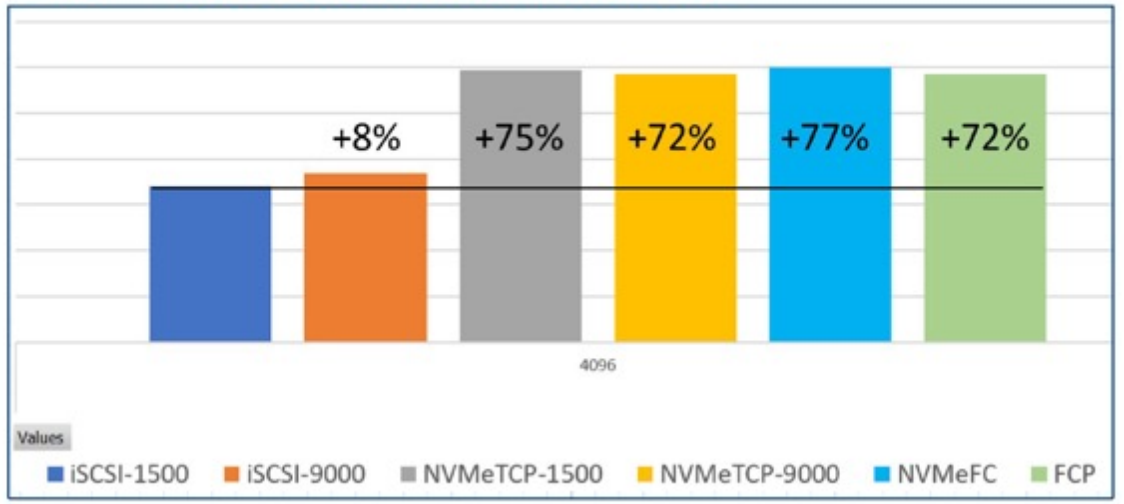


Performance Comparison – Total IOPS

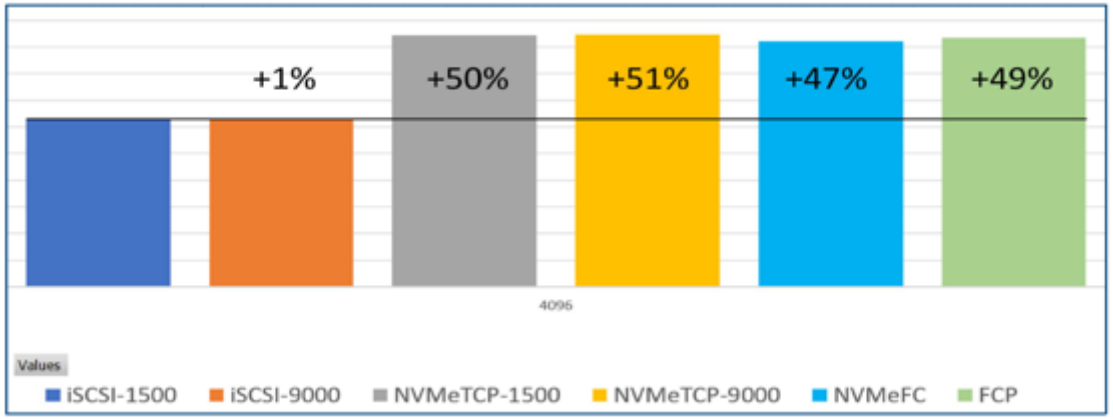
3.1.1 IOPS – 4K - 100% READ



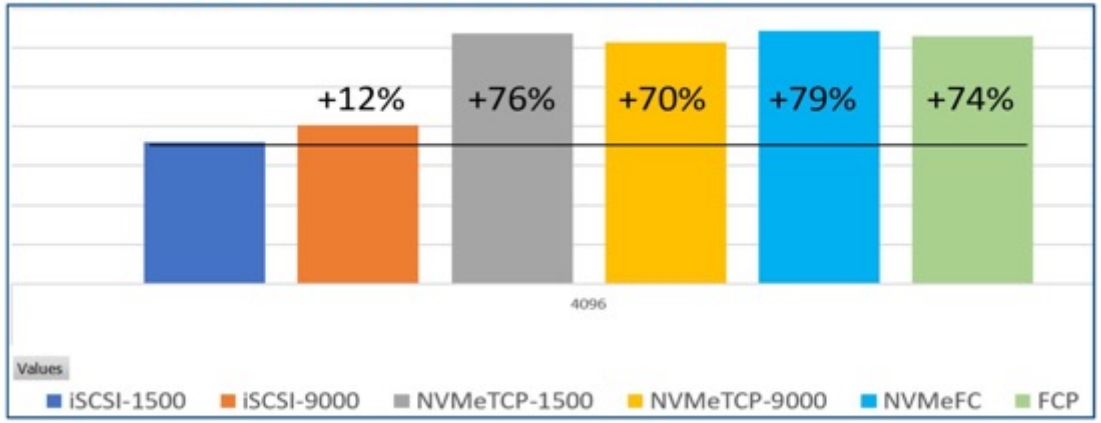
3.1.3 IOPS – 4K - 50% READ / 50% WRITE



3.1.2 IOPS – 4K - 100% WRITE

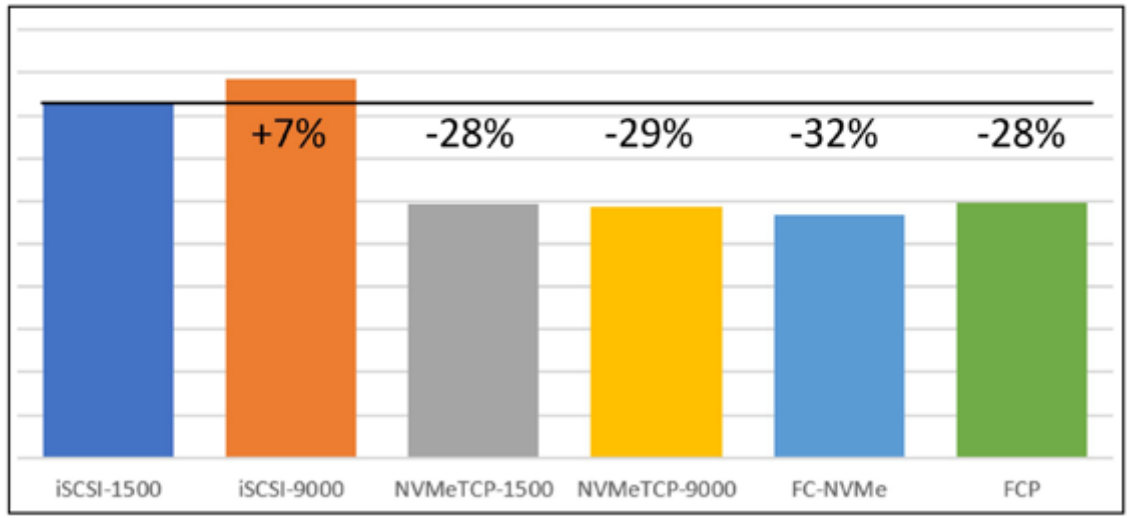


3.1.4 IOPS – 4K - 70% READ / 30% WRITE

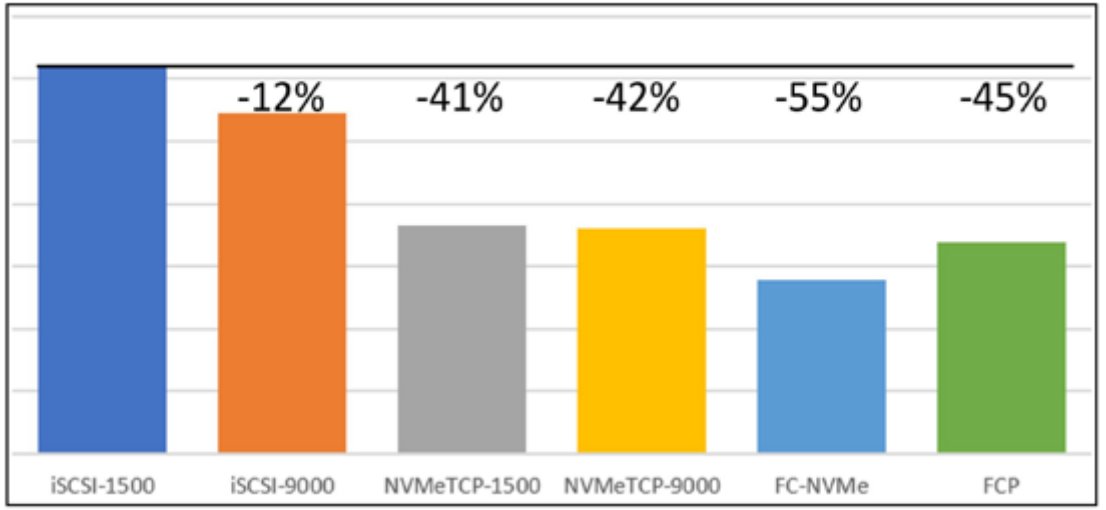


Performance Comparison – Overall CPU Utilization

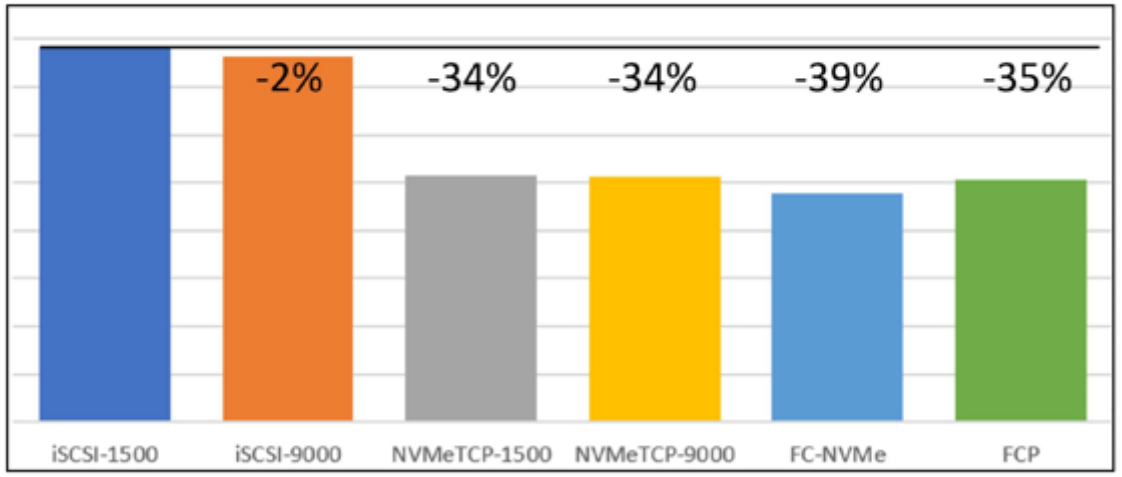
3.3.1 CPU Utilization – 4K - 100% Write



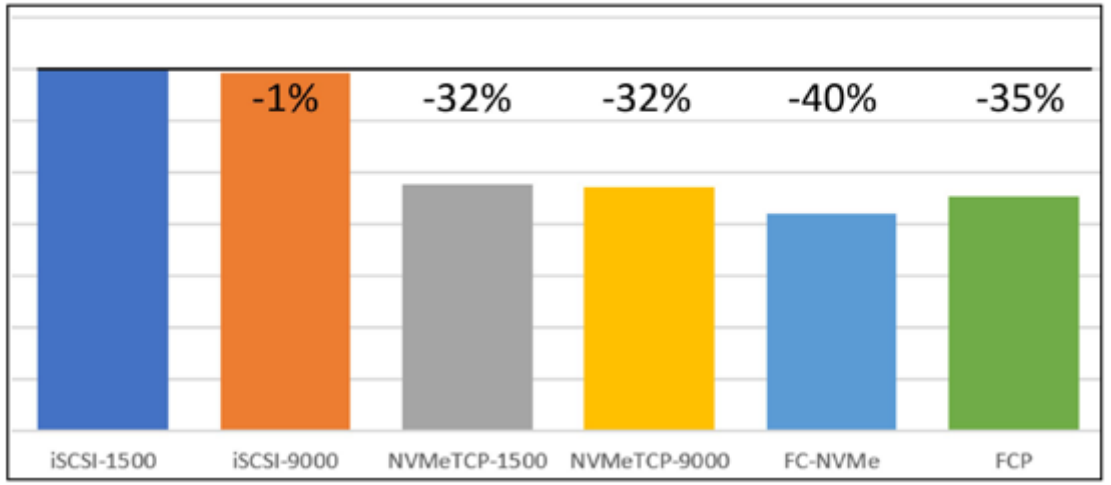
3.3.2 CPU Utilization – 4K - 100% Read



3.3.3 CPU Utilization – 4K - 50% READ / 50% WRITE



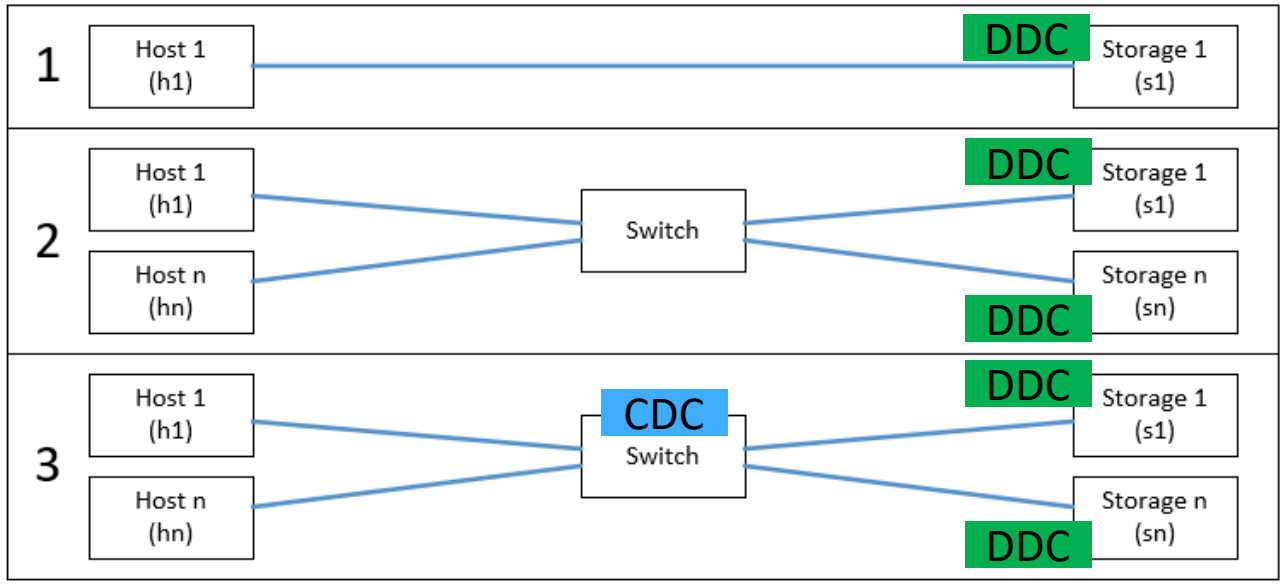
3.3.4 CPU Utilization – 4K - 70% READ / 30% WRITE



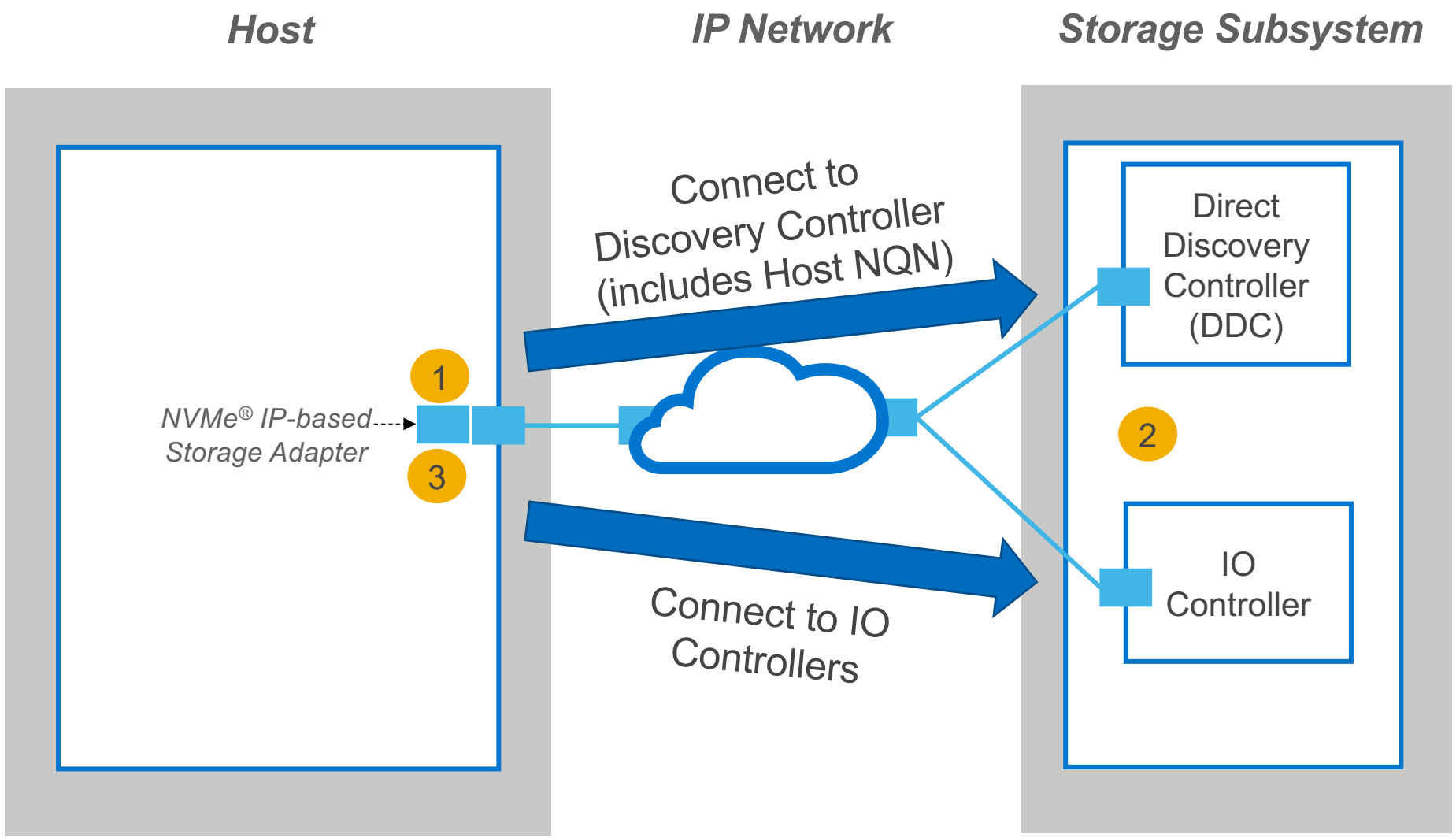
Deployment Considerations

Supported Deployment Types

- We (NVM Express - FMDS WG) defined a standardized solution to allow for the discovery of Discovery Controllers
  - Automated – No explicit configuration or “seed” information required
  - Dynamic - Discovery Controllers will come and go
  - Capable of being Highly Available
- Solution must scale to support:
  1. Direct Connect
     Example: test/dev environments
  2. Single broadcast domain
     Example: Single switch deployments
  3. Centralized DC **(NEW)**
     Example: Multi-switch configurations

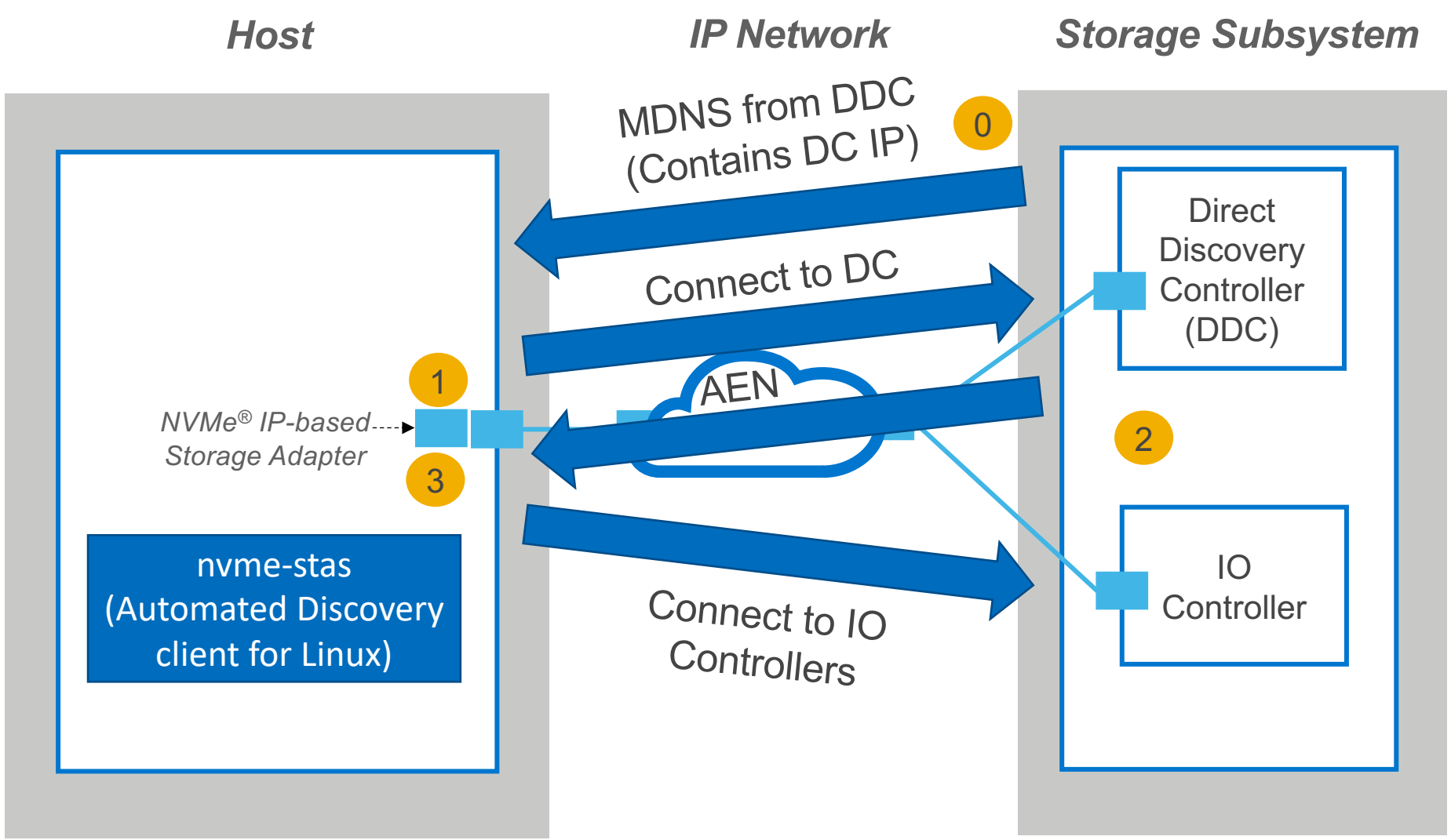


Configuration Steps without Automated Discovery (Pre 8009/8010)



1. Host Admin configures the host to connect to a Discovery Controller on the subsystem at a particular IP Address.
2. Storage admin provisions Namespaces (Storage) to the Host NQN
3. Host Admin can now Discover and connect to IO Controllers on this subsystem
4. Repeat 1-3 on all Hosts for each subsystem

Configuration Steps with Automated Discovery of DDCs (TP8009)



0 Host and subsystems automatically discover each other via mDNS,

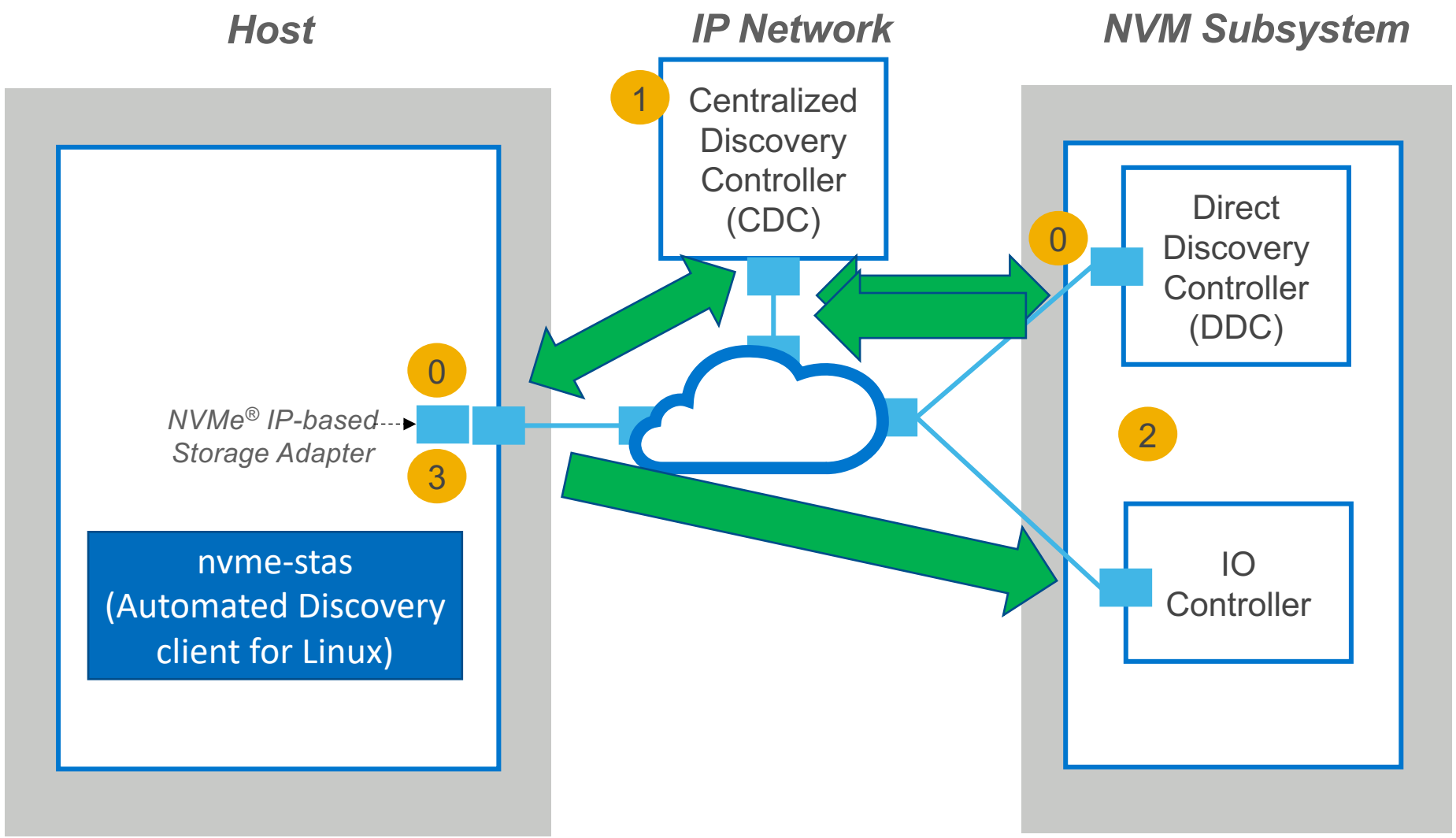
1 Host automatically sends connect to DDC at IP Address Discovered via mDNS

2 Storage admin provisions Namespaces (Storage) to the Host NQN

3 Host receives AEN from DDCs, retrieves the list of IO Controllers on that subsystem and connects to them.

4 Repeat 1-3 on all Hosts for each subsystem

Configuration Steps with Automated Discovery of CDC (TP8009/8010)



0 Host and subsystems automatically discover the CDC, connect to it and Register Discovery info

1 Zoning performed on CDC (optional)

2 Storage admin provisions namespaces to the Host NQN. Storage may send zoning info to CDC

3 After zoning, Host receives AEN, uses get log page, and connects to each IO Controller

4 Repeat 1-2 for each Hosts on each subsystem

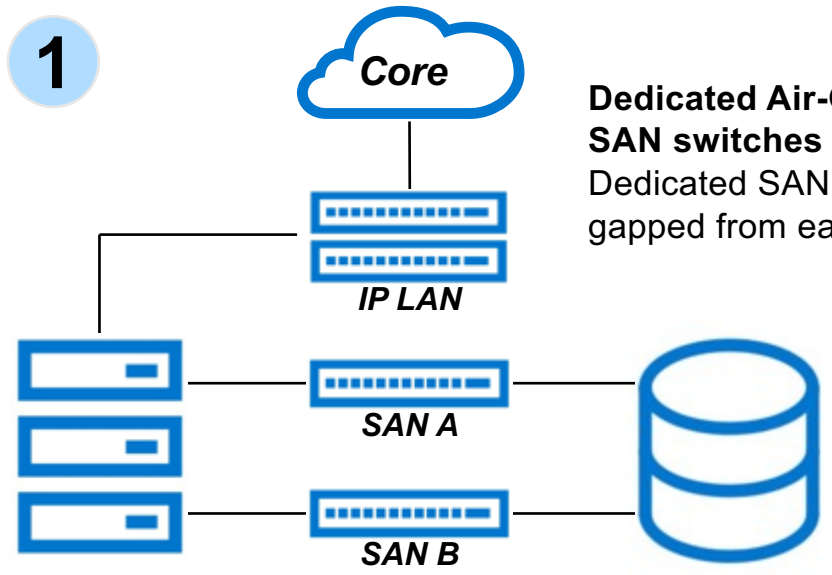
nvme-stas – Automated Storage Discovery for Linux

- libnvme used by both nvme cli and nvme-stas
- use nvme cli for manual (one-shot) configuration
- use nvme-stas for dynamic/automated discovery of storage

Features - Comparing nvme-stas with nvme-cli

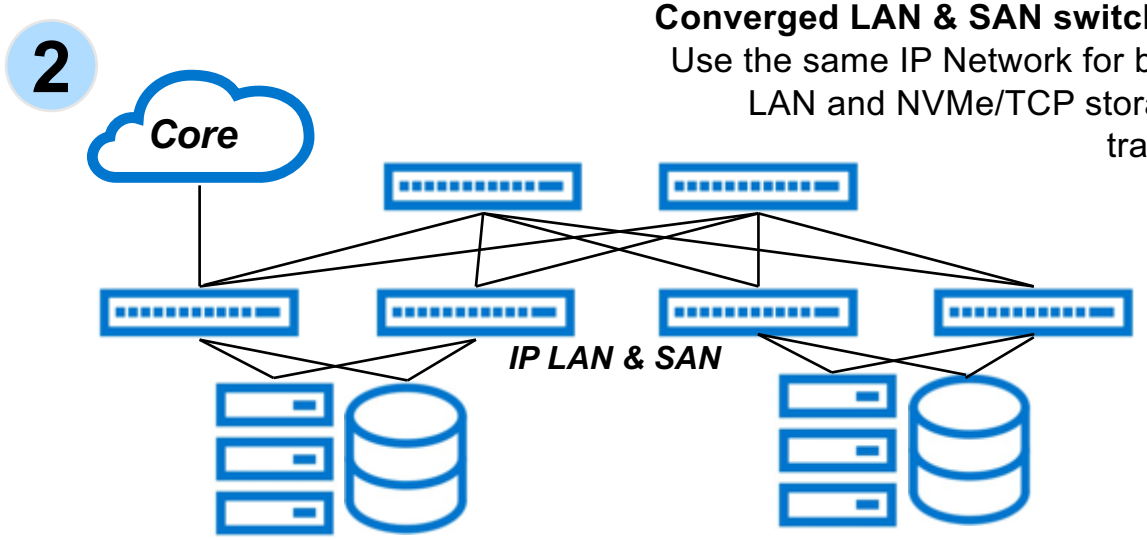
Feature	nvme-stas	nvme-cli
IP address family filter	**Yes** – /etc/stas/*.conf: **ip-family**=[ipv4, ipv6, ipv4+ipv6]	**No**
Automatic DIM registration with Central Discovery Controller (CDC) per TP8010	**Yes**	**No** – Manual only: nvme dim
Automatic (zeroconf) discovery of Direct/Central Discovery Controller (DDC/CDC)	**Yes** – stas registers with the Avahi daemon to be notified when CDCs or DDCs are detected by mDNS service discovery and automatically connects to them.	**No**
Manual Discovery Controller (DC) config with explicit include/exclude	**Yes** – /etc/stas/stafd.conf: **controller=, exclude=** Exclusion is needed to eliminate unwanted mDNS-discovered DCs.	**Partial** – No way to exclude DCs (*moot point since mDNS is not supported*). Use /etc/nvme/discovery.conf to include controllers.
Manual I/O Controller (IOC) config with explicit include/exclude	**Yes** – /etc/stas/stacd.conf: **controller=, exclude=** Exclusion is needed to eliminate unwanted IOCs from log pages *(although this should really be done by properly defining the zones at the DC)*	**Partial** – With the help of JSON config files, but no possibility to exclude IOCs.
AEN monitoring + Auto Connection/Disconnect for Fabric Zoning support	**Yes** – React to Fabric Zoning changes (configurable: /etc/stas/stacd.conf) Connect and Disconnect with retries.	**Partial** – React to Fabric Zoning changes Connect-only without retries (one-shot udev rule)
Use PLEO bit to get only Port Local Entries when retrieving log pages	**Yes**	**No**
Layer 3 connectivity w/o static routes	**Yes** – Automatic/Configurable (/etc/stas/*.conf: ignore-iface=)	**Yes** – Manual (--host-iface)
Explicit exclude of specific interfaces used for discovery	**Yes** – /etc/stas/*.conf: **exclude** = host-iface=<interface>	**No**
AVE client support	**Planned** – Under design	**No**
Human-friendly "nvme list" command	**No** – stafctl and stacctl only display data in JSON format (for now). *Not needed since "nvme list -v" does the job so well.*	**Yes** – nvme list -v

Dedicated vs. Converged Topology



Dedicated Air-Gapped Dual SAN switches
Dedicated SAN switches, air-gapped from each other.

- Familiar to FC experienced teams
- Switches and ports dedicated to NVMe/TCP SAN traffic
- More secure: Isolated from attack vectors such as Man-in-the-Middle and impersonation
- Spine-Leaf topology to scale up each SAN

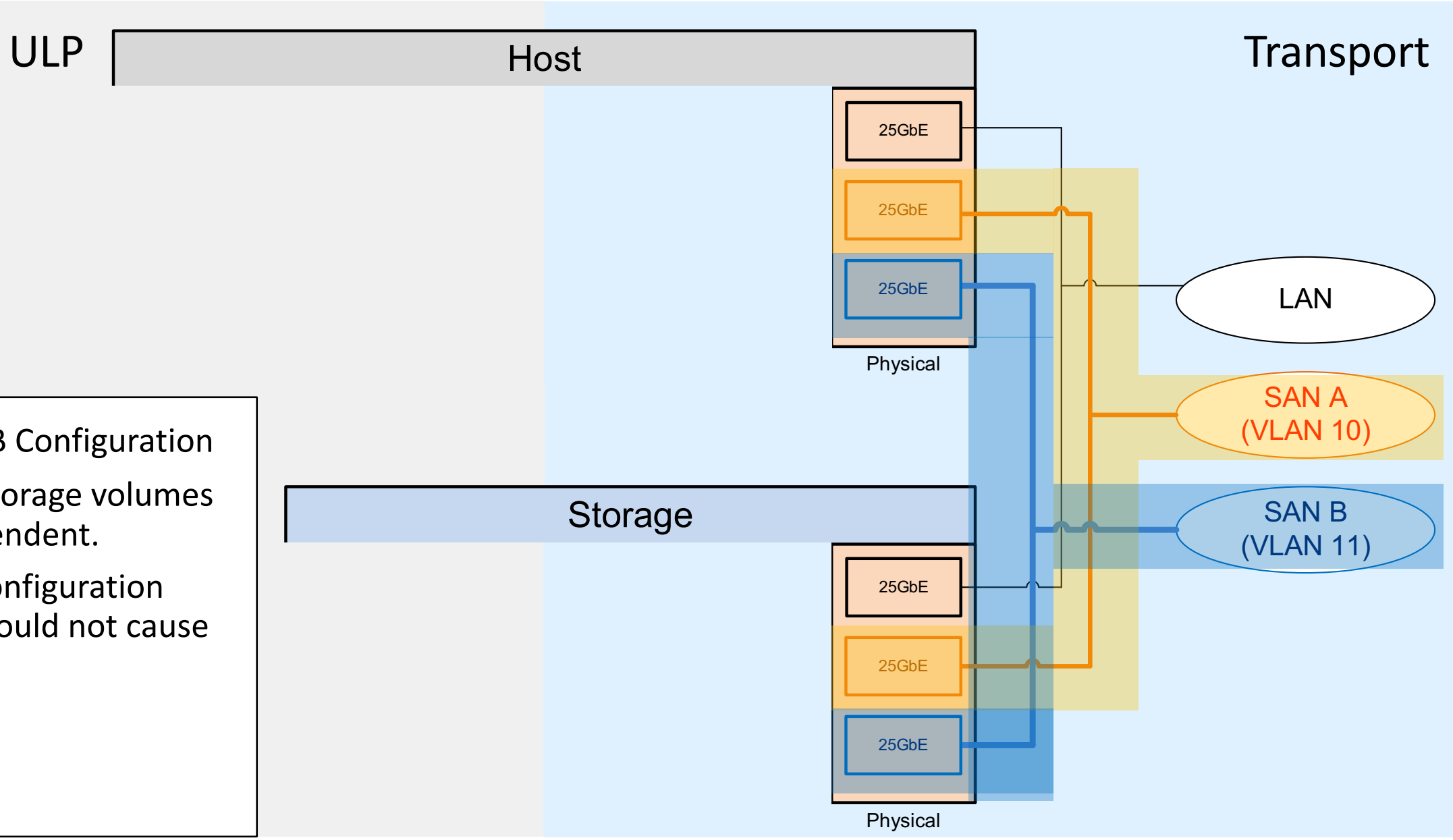


Converged LAN & SAN switches
Use the same IP Network for both LAN and NVMe/TCP storage traffic.

- Familiar to IP Networking teams
- Can reuse existing switches
- If NVMe/TCP traffic is sharing uplinks, congestion monitoring and avoidance tools are recommended
- Spine-Leaf topology to scale up

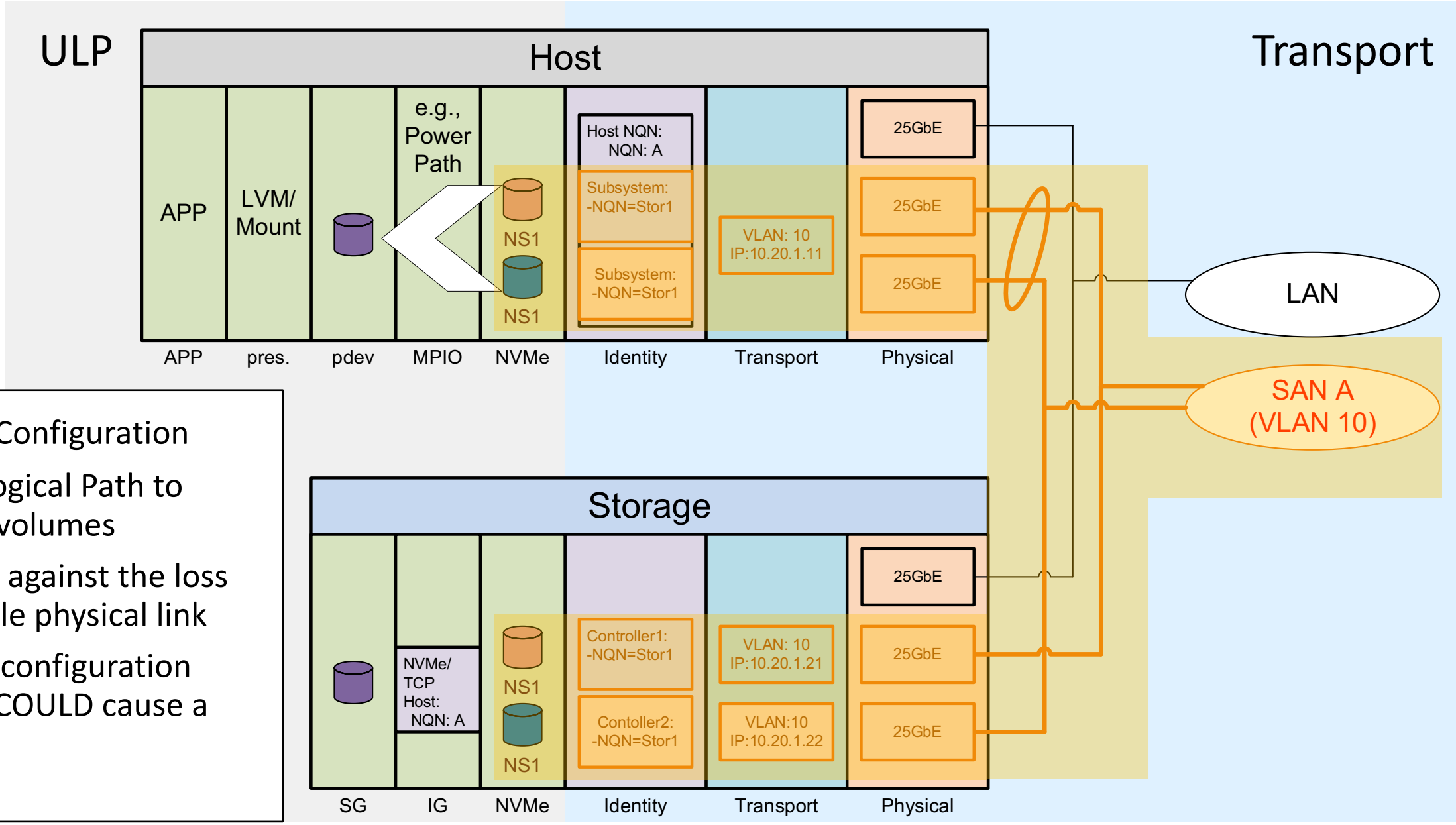
Resiliency – Link Aggregation versus Storage Multipathing

Link Aggregation	Storage Multipathing
Multiple network interfaces act as a single logical link	Utilizes (expects) multiple network interfaces to ensure isolation of each path
Network configuration is required	Network configuration is not required
Enables link level resiliency	Enables end-to-end resiliency
Does not protect against Upper Layer Protocol (ULP) related configuration issues causing a DU (Data Unavailability) event. (e.g., zoning, masking, etc)	Helps prevent DU due to ULP misconfiguration



SAN A / SAN B Configuration

- Paths to storage volumes are independent.
- A single configuration activity should not cause a DU.



Single SAN Configuration

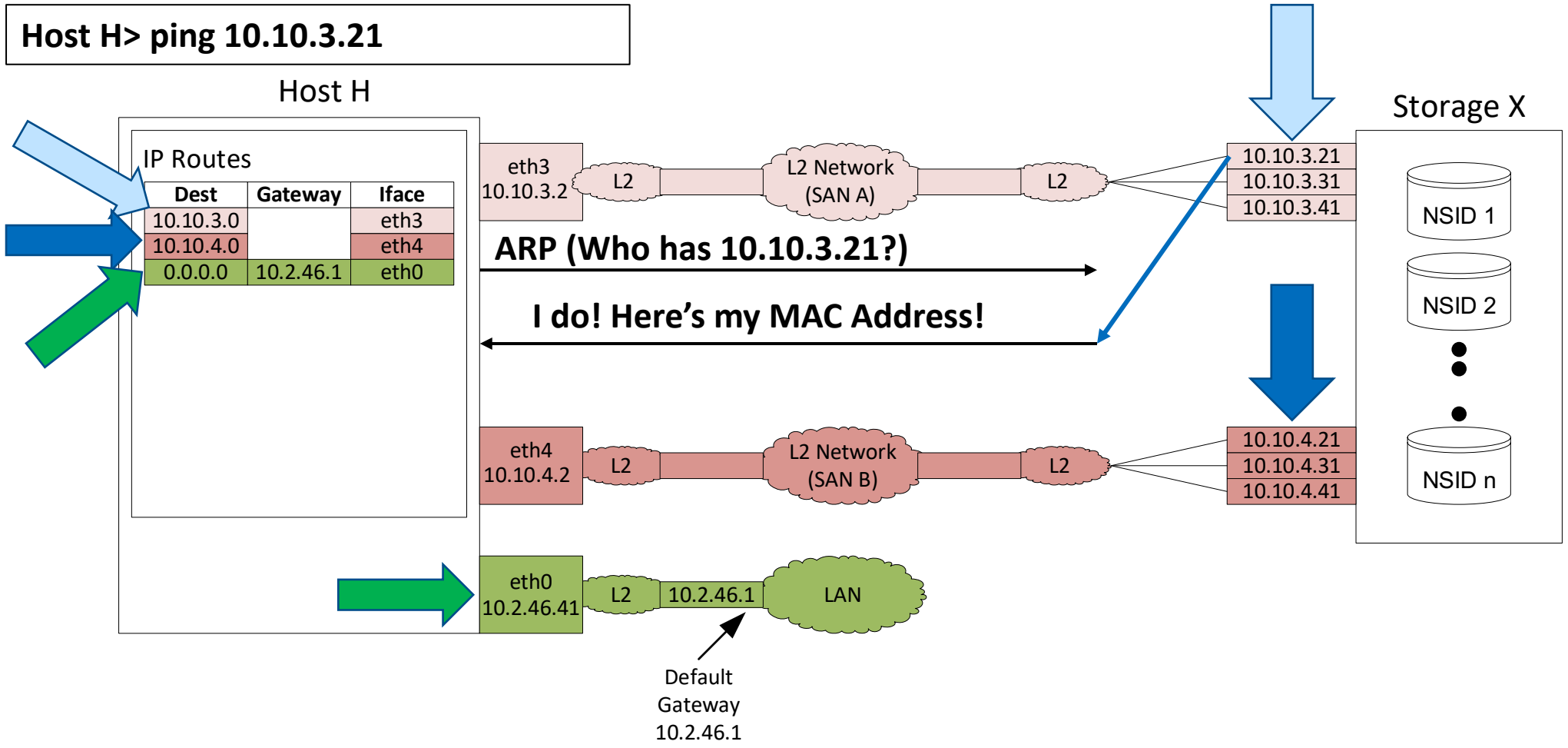
- Single Logical Path to storage volumes
- Protects against the loss of a single physical link
- A single configuration activity COULD cause a DU.

Accessing Storage over L2 and L3 networks

- The network topology of an IP Based SAN has an impact on a host's ability to access external storage (e.g., an array).
- **An L2 IP Based SAN** = Host interfaces accessing external storage interfaces that are on the same subnet.
  - This is very straight-forward.
- **An L3 IP Based SAN** = Host interfaces accessing external storage interfaces that are NOT on the same subnet.
  - This requires additional configuration when using Linux or Windows

Accessing Storage over L2 and L3

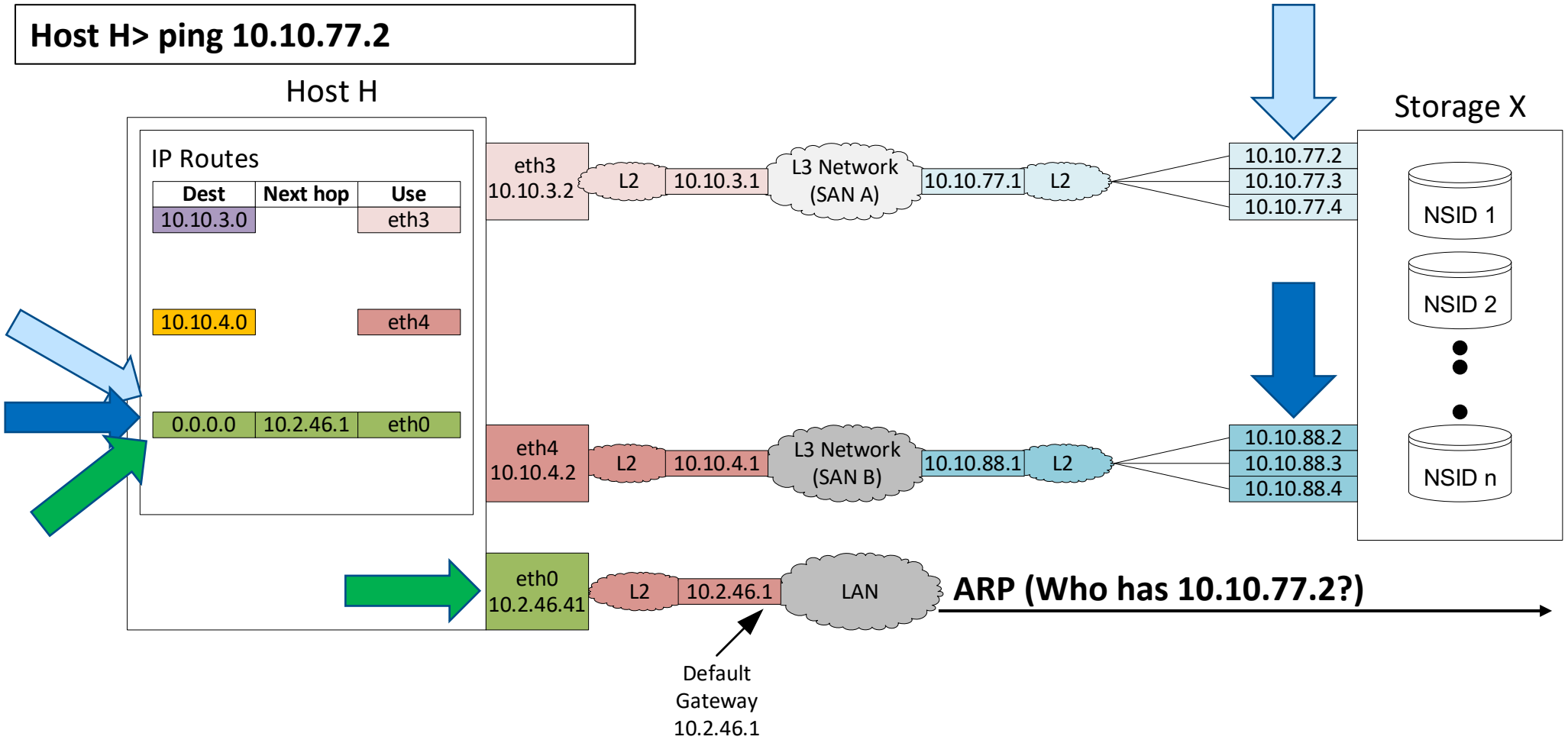
Host and Storage interfaces are on the same L2 subnet



This example assumes subnets are /24

Accessing Storage over L2 and L3

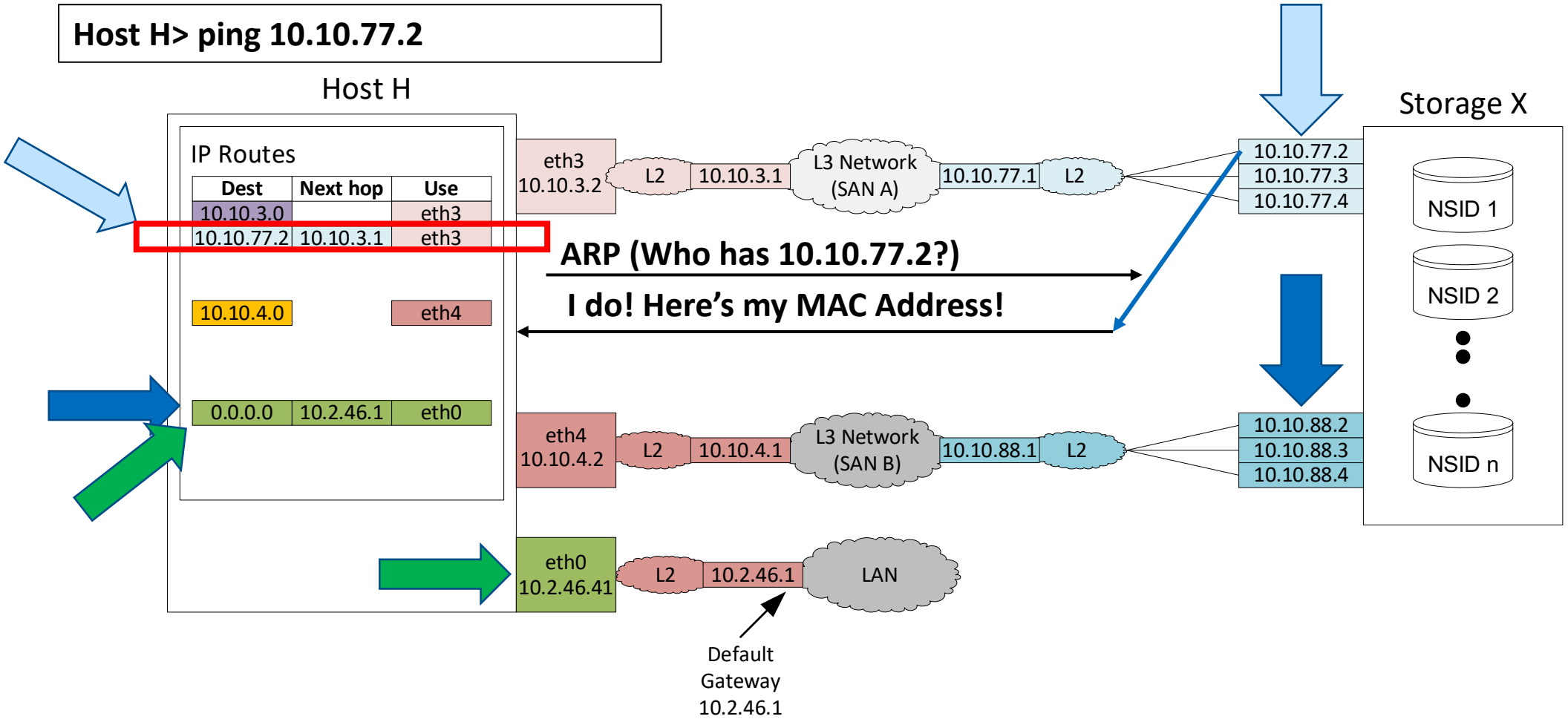
Host and Storage interfaces are on **different** L2 subnet



This example assumes subnets are /24

Accessing Storage over L2 and L3

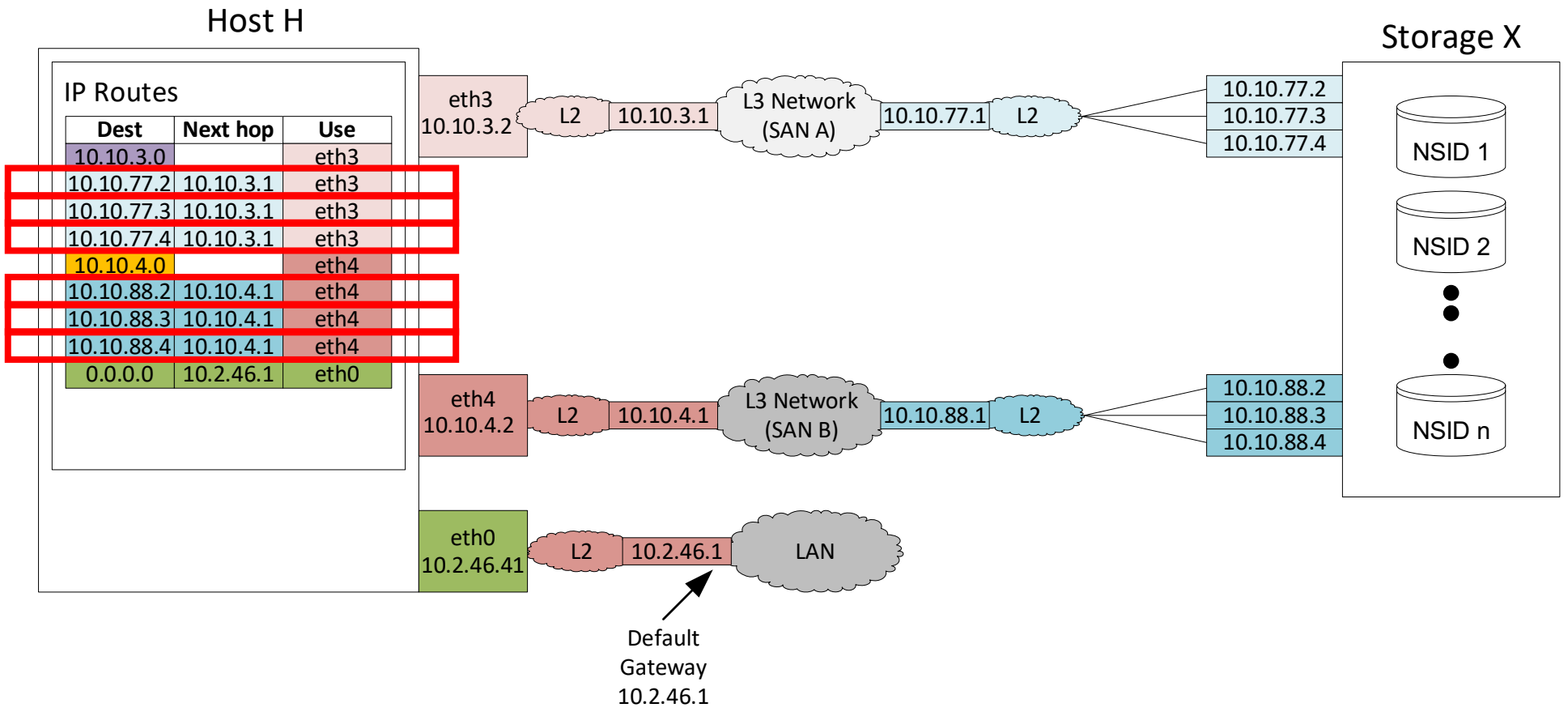
Host and Storage interfaces are on **different** L2 subnet – Solution 1 – Manually update the Route table.



This example assumes subnets are /24

Accessing Storage over L2 and L3

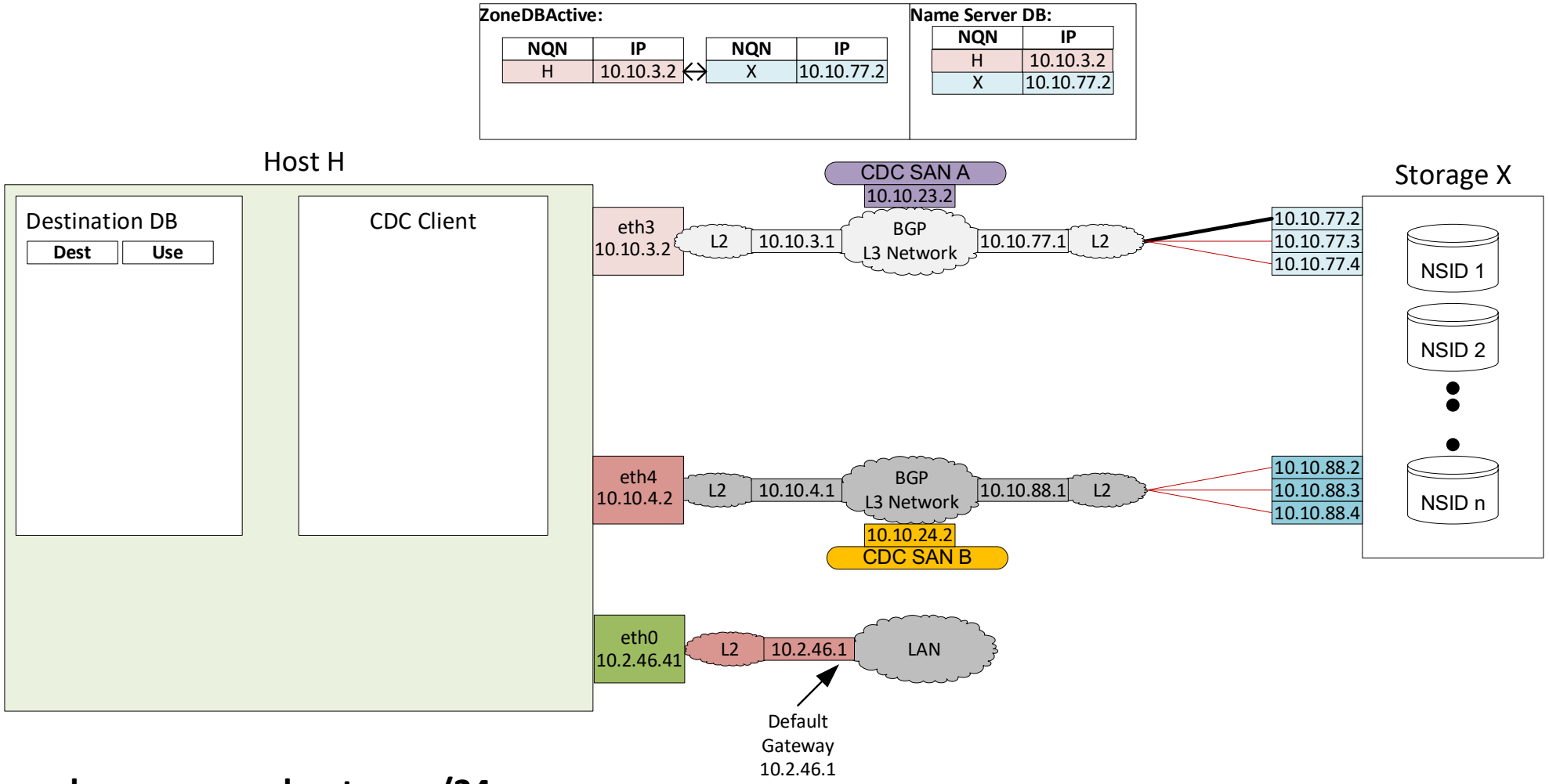
Host and Storage interfaces are on **different** L2 subnet – Solution 1 – Manually update the Route table.



This example assumes subnets are /24

Accessing Storage over L2 and L3

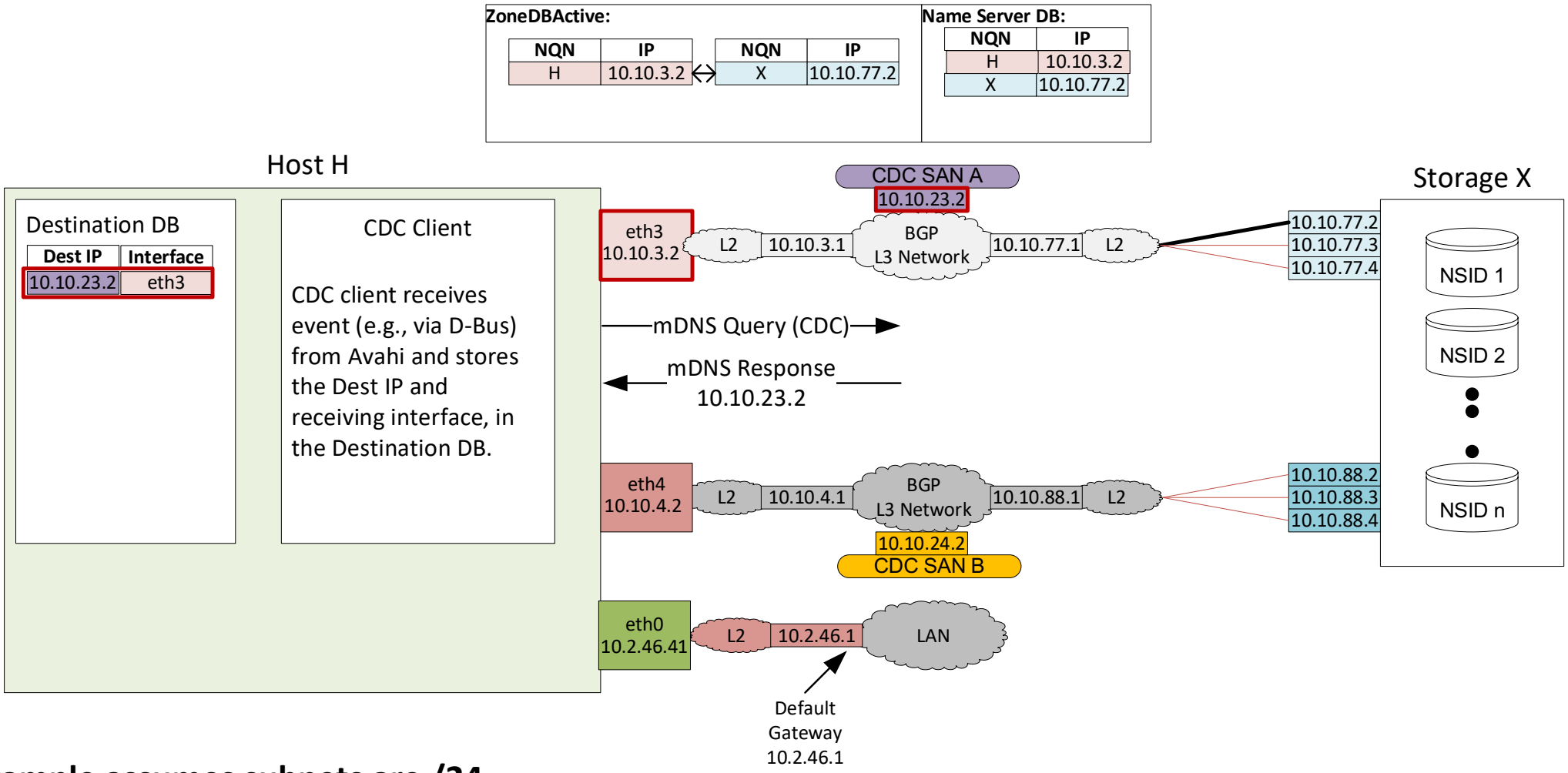
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

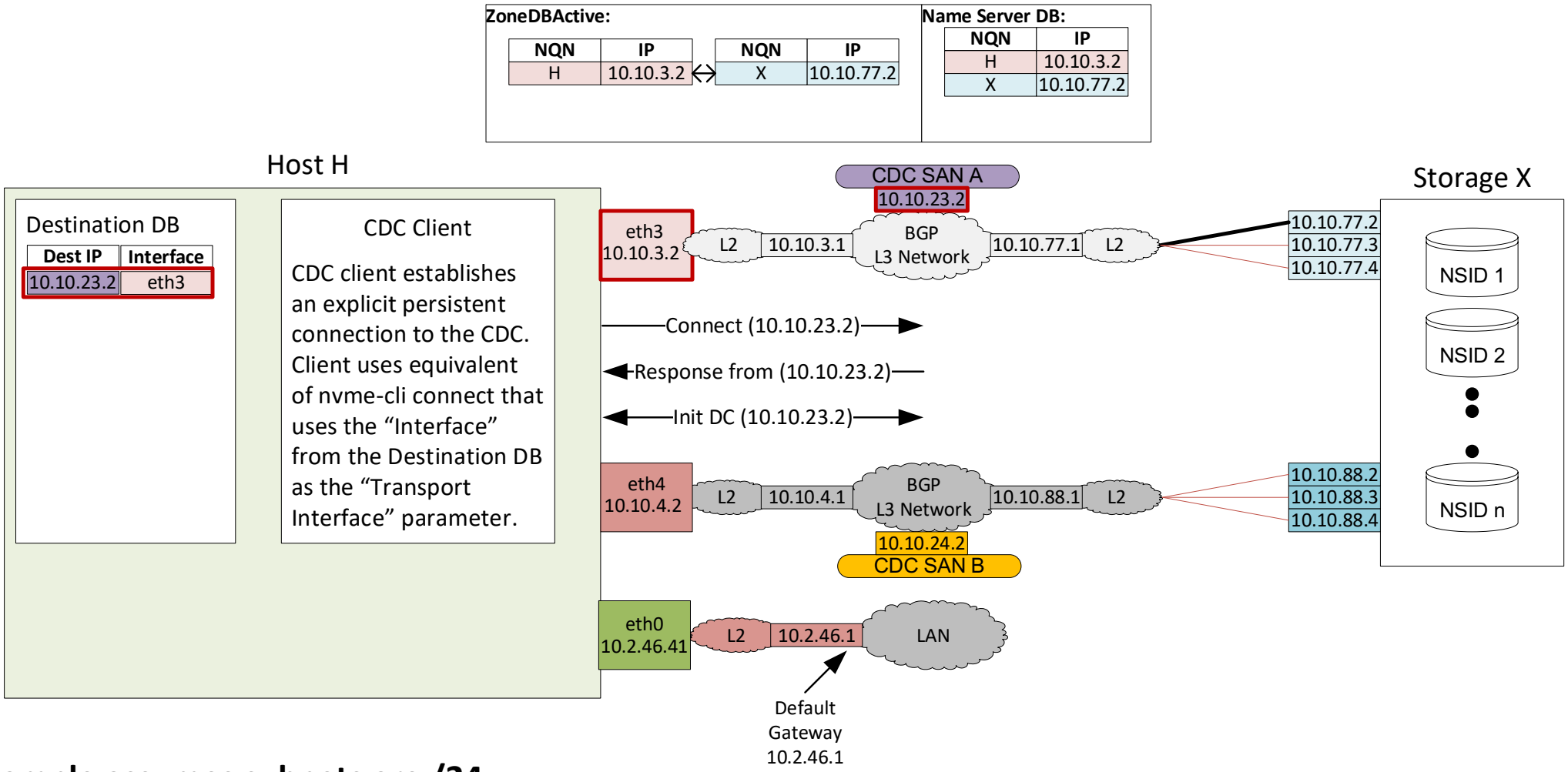
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

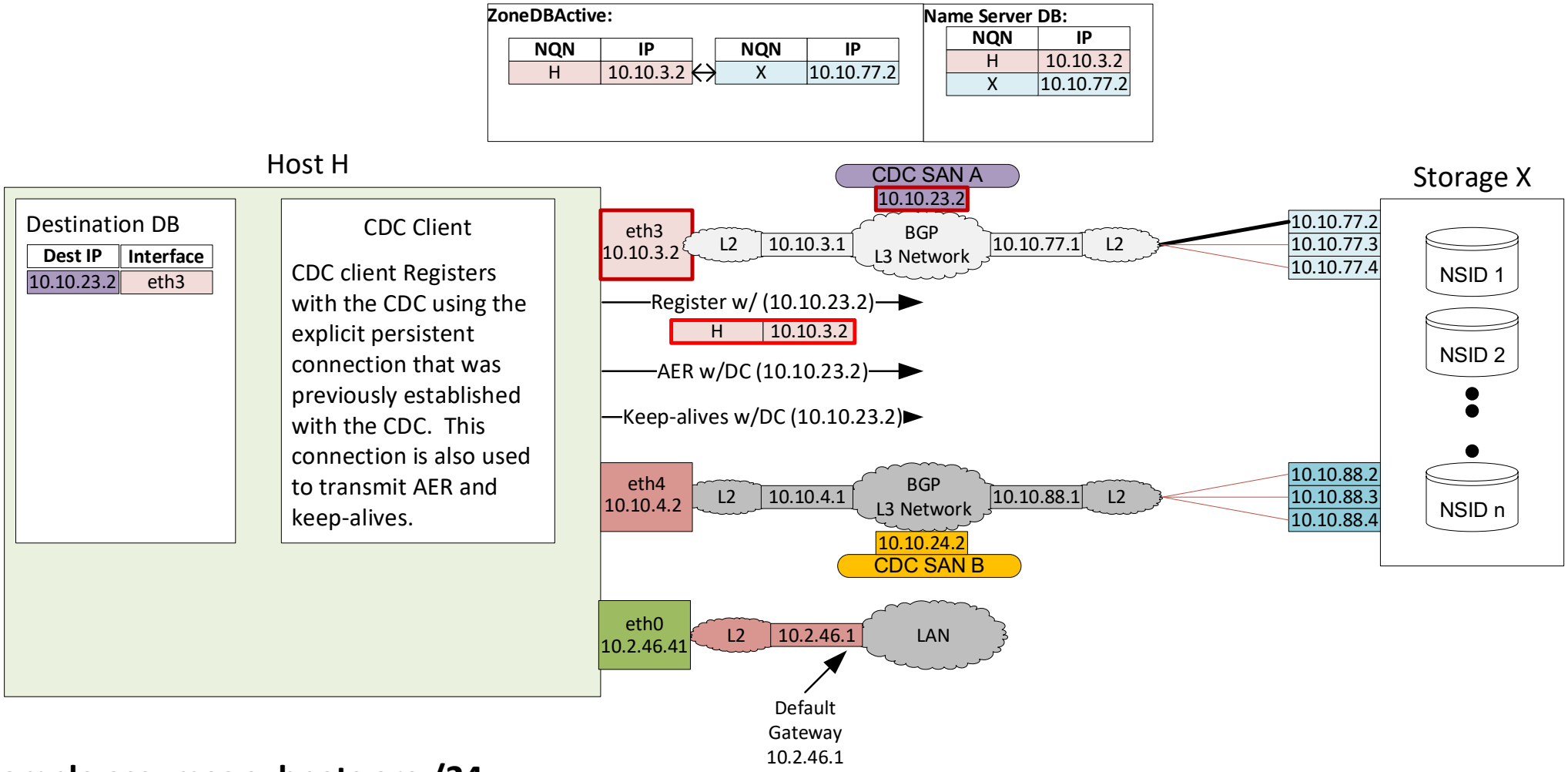
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

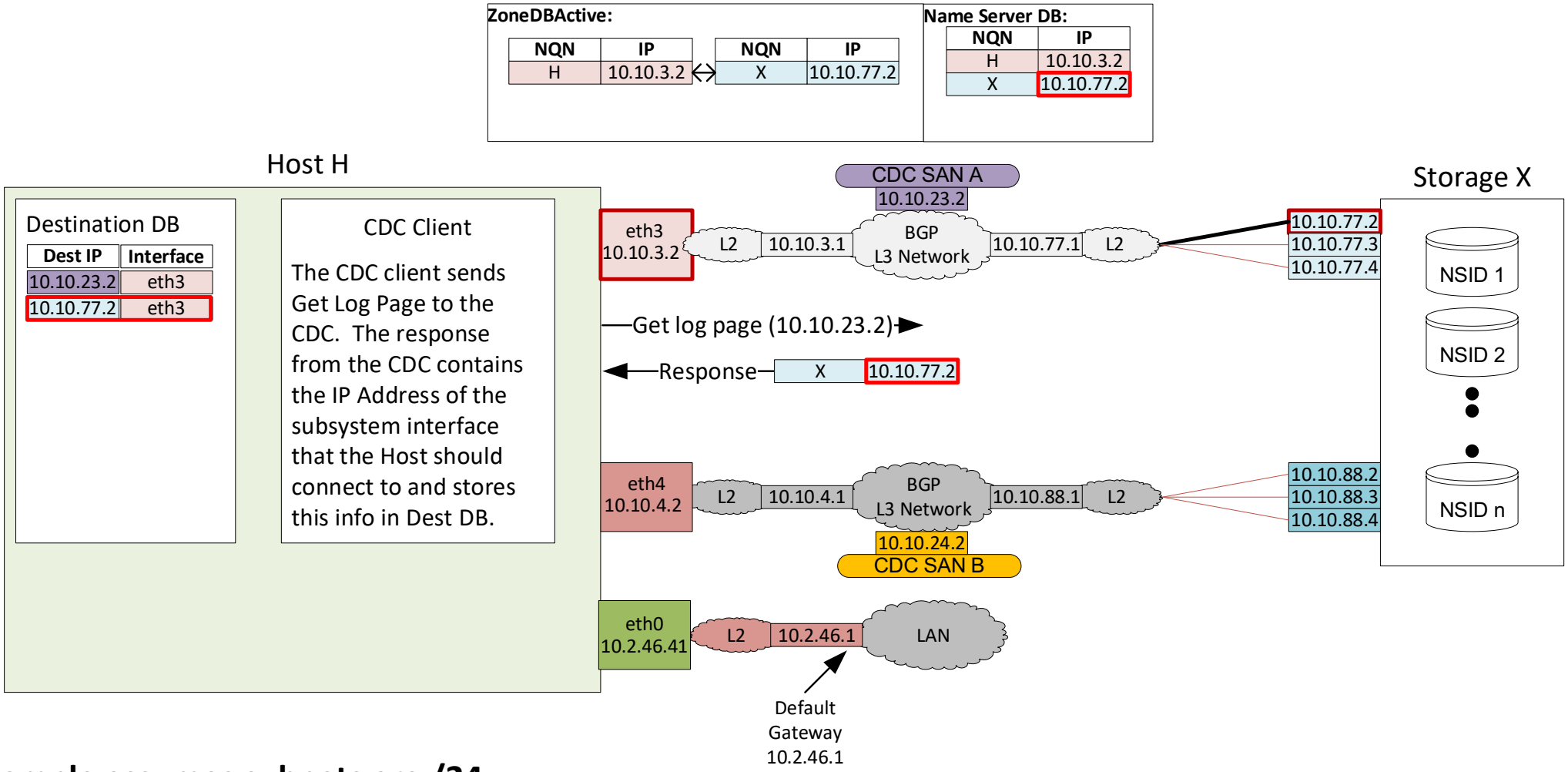
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

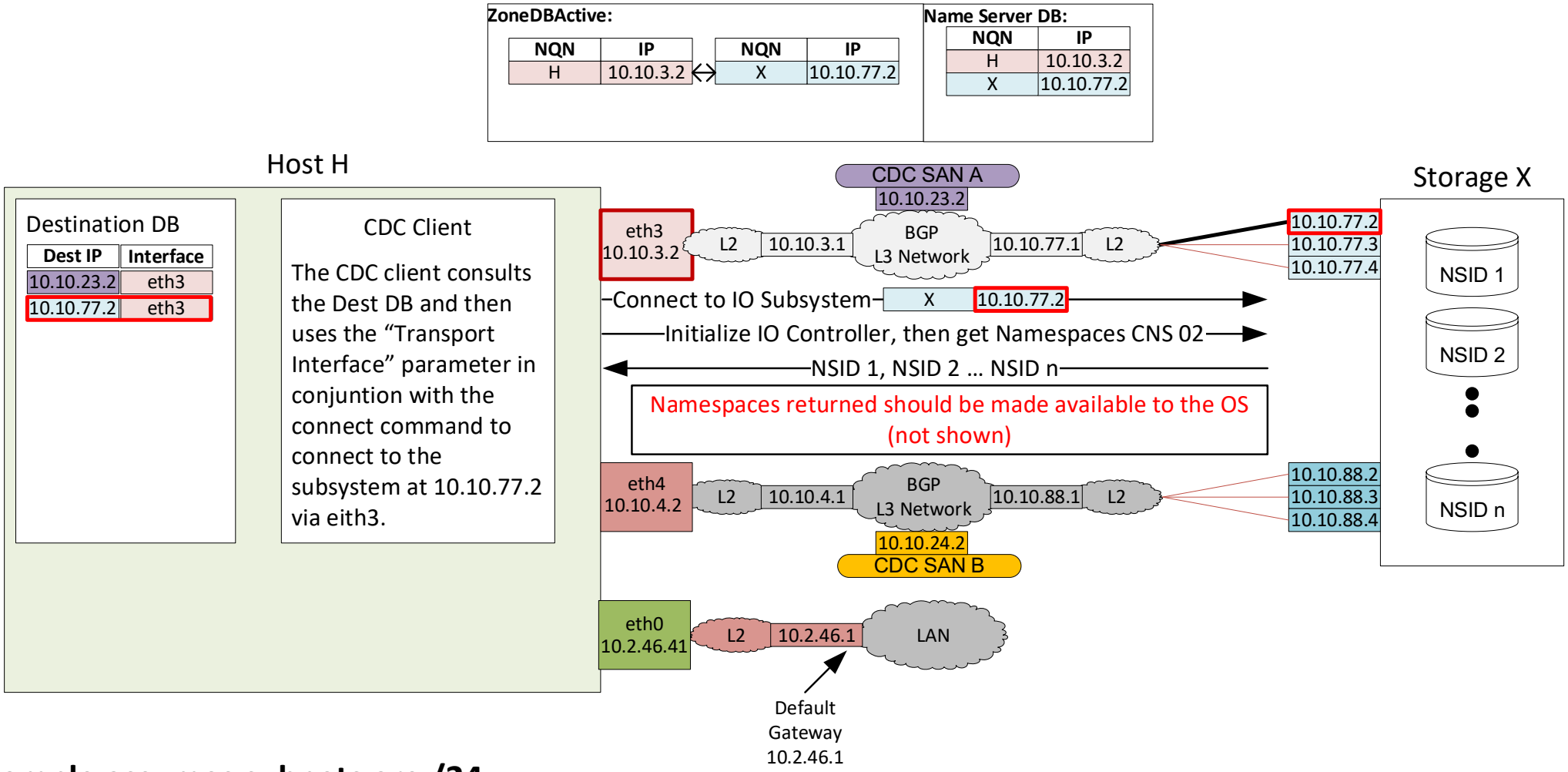
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

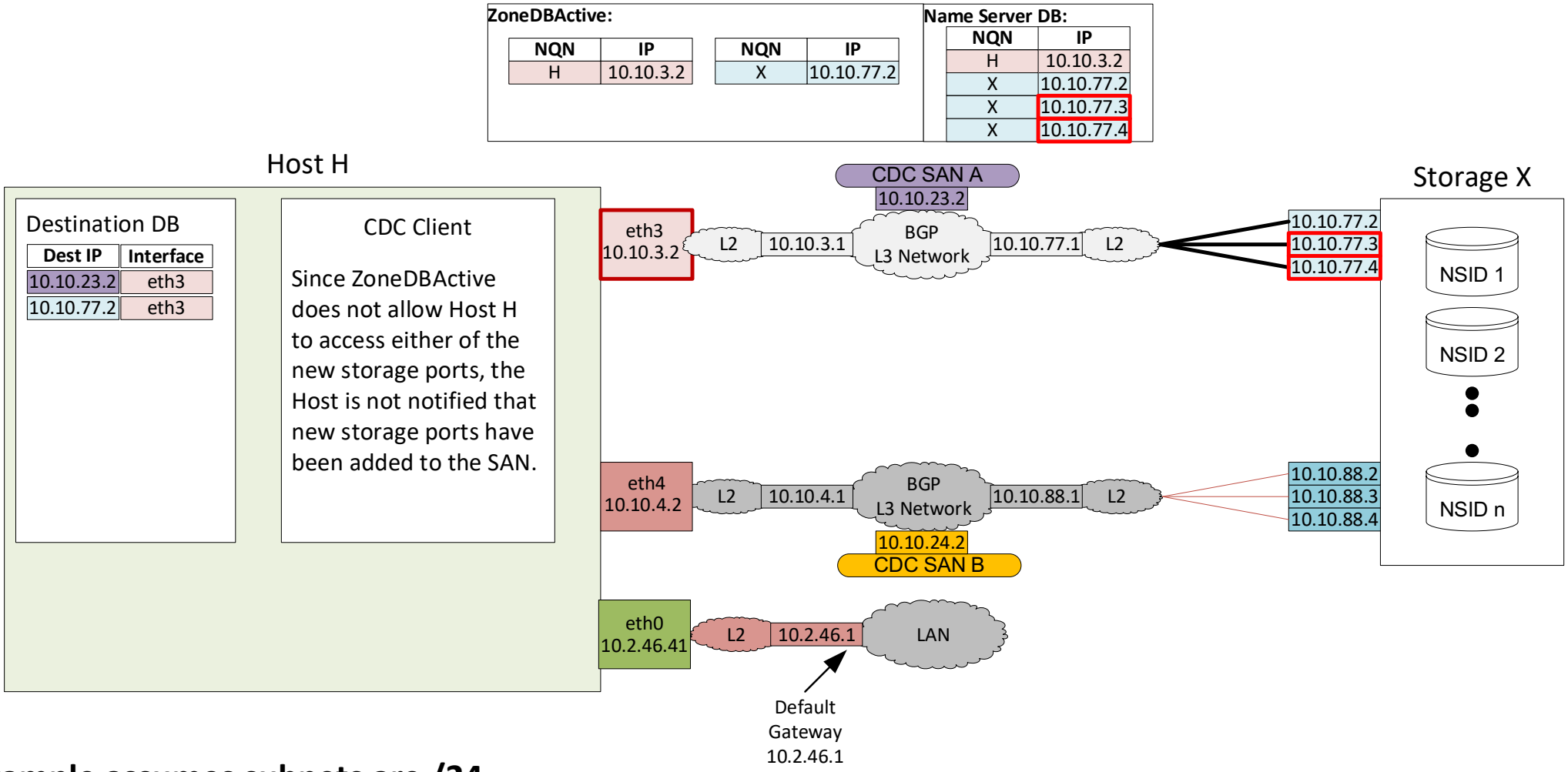
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

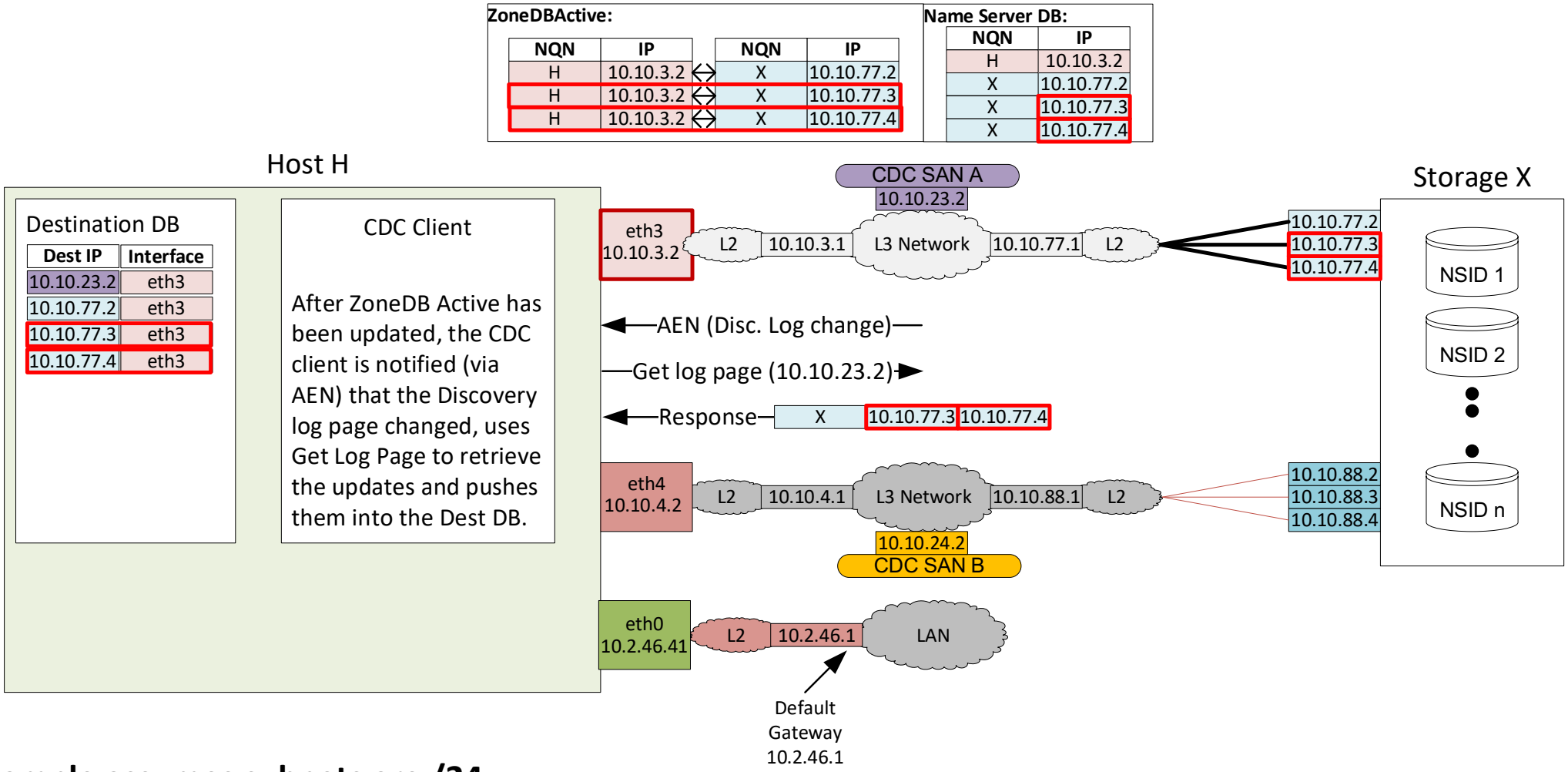
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Accessing Storage over L2 and L3

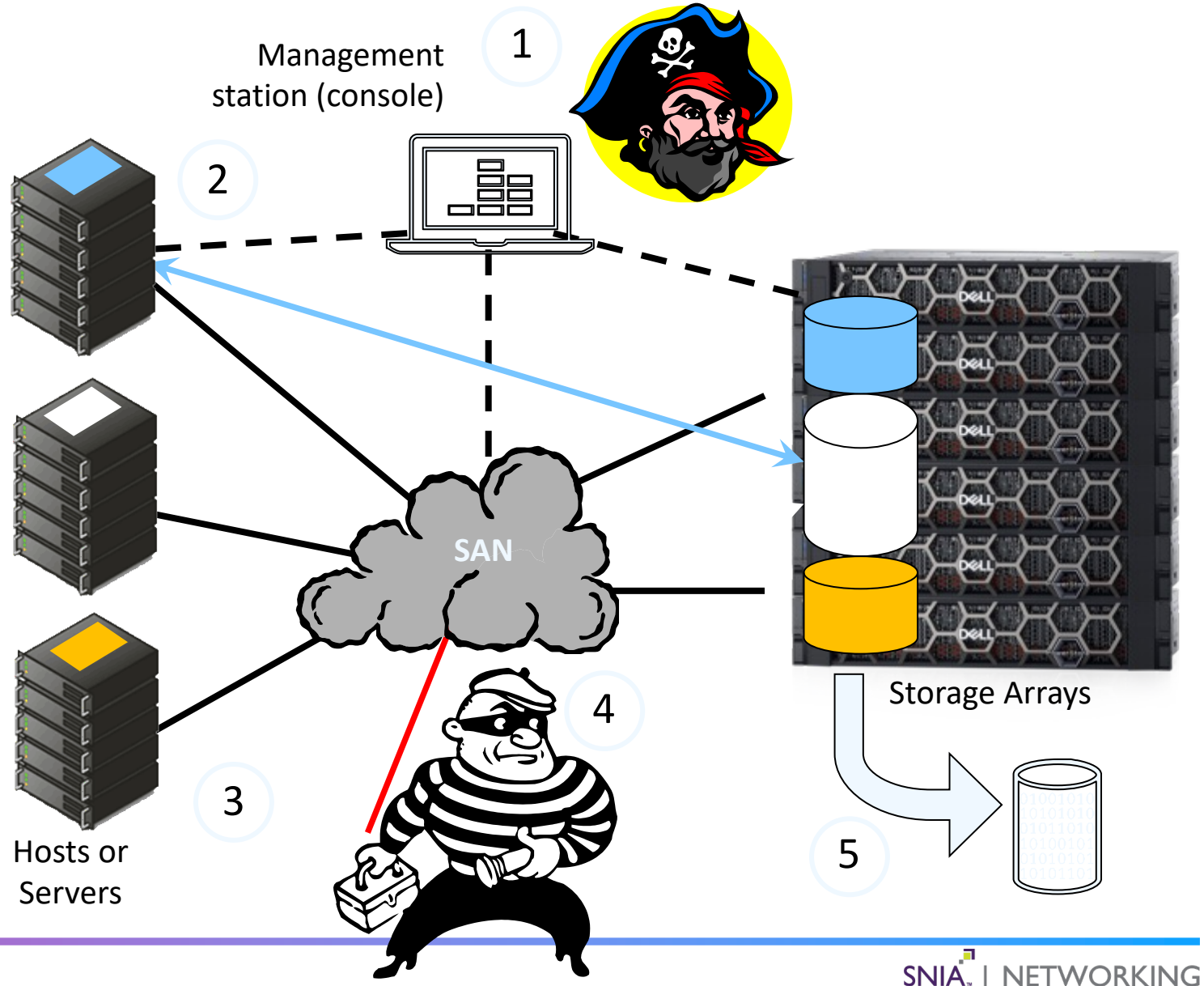
Host and Storage interfaces are on **different** L2 subnet – Solution 2 – Automation!



This example assumes subnets are /24

Storage Area Networking Security Threats

1. **Management & System Integrity**
2. **Uncontrolled Storage Access**
3. **Impersonation (Spoofing)**
4. **Communication Access**
   - Eavesdrop
   - Inject/Modify
5. **External Access**
   - Media theft
   - Other access



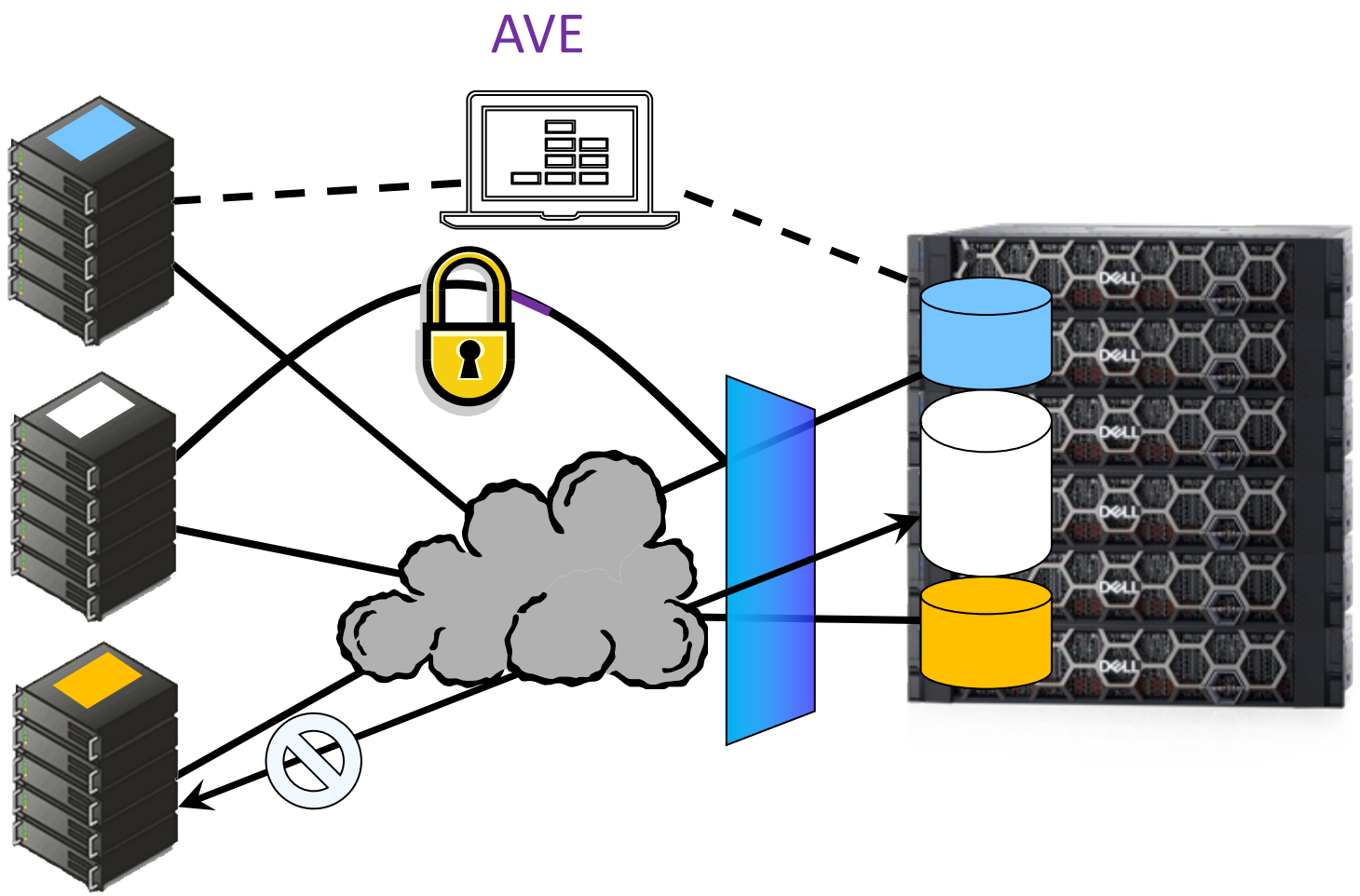
Security Threats and Countermeasures

Risks / threats	Countermeasures
Management & System Integrity	Authentication & Authorization Logging and Anomaly Detection Secure Channels Hardware/software/firmware integrity checks Hardware root of trust
Uncontrolled Storage Access	NVMe namespaces mapping NVMe-oF Zoning
Impersonation (Spoofing)	Authentication (Proof of identity) DH-HMAC-CHAP
Communication Access • Eavesdrop • Inject/Modify	Secure Channel (data in flight) Confidentiality Cryptographic Integrity TLS 1.3
External Access • Media theft • Other access	NVMe-oF: Data in flight encryption Data at rest encryption

Focus of NVMe/TCP Security initiatives

Authentication Verification Entity (AVE)

- A centralized way to perform authentication verification
- A server replying to authentication verification requests
  - Maintains a centralized database of {identity (NQN), key(s)} records
  - Performs authentication verification computations on behalf of its clients
  - Only authenticated authorized entities allowed access to the AVE
  - Similar function of a RADIUS server
  - Will be integrated with SFSS



Authentication Based Zoning (ABZ)

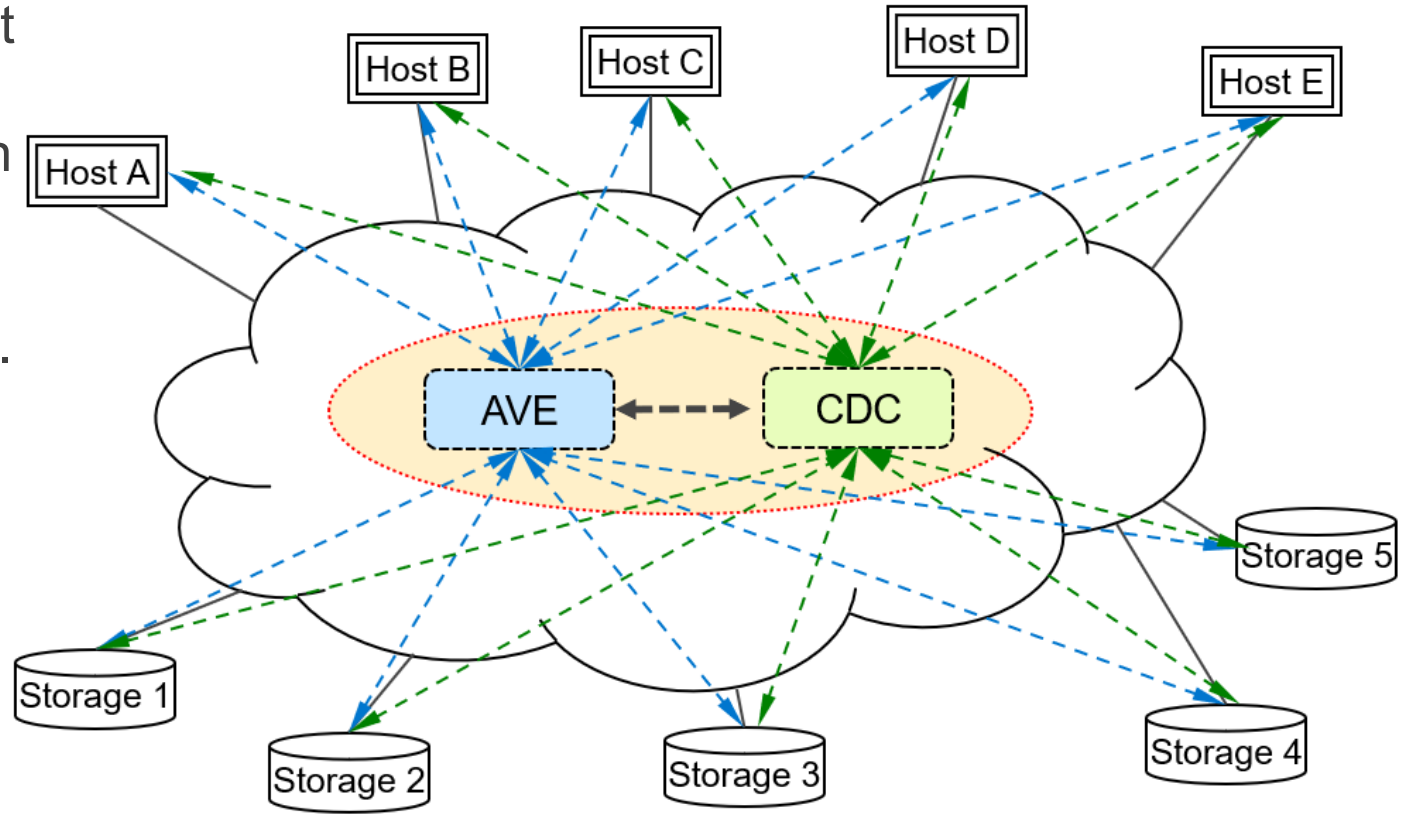
“Hard” Zoning for IP-Based SANs

- Centralized Discovery Controller (CDC)
  - Enables hosts to discover storage resources
  - Filters information returned to hosts and subsystems based on access control rules (e.g., zoning)
- Authentication Verification Entity (AVE):
  - A centralized authentication verification server
    - Offloads hosts and controller from authentication verification
    - Makes key management scalable (just one key per host/controller)
- They can work together as a part of a single solution

Authentication Based Zoning (ABZ)

“Hard” Zoning for IP-Based SANs

- The AVE can act as a Zoning enforcement agent
- The AVE can use the Zoning configuration from the CDC to determine what connections between hosts and subsystems are allowed and what are not.
  - If a connection between a host and a subsystem is allowed by Zoning, then the AVE performs authentication verification
  - If a connection between a host and a subsystem is not allowed by Zoning, then the AVE returns an authentication verification failure, without the need to perform authentication verification



Automation

SANdbox – Overview and Demo

- SANdbox is a Github repo that contains:
  - Scripts that demonstrate how to provision NVMe/TCP storage to a host
    - Powershell and Python scripts are available.
  - Documentation
  - SFSS (Dell's CDC implementation) that can be downloaded and used for evaluation purposes.
- For more information see the following:
  - Let's play in an AWS based SANdbox!
  - Let's play in an AWS based SANdbox – Part 2 Setting up your AWS VPC and networks

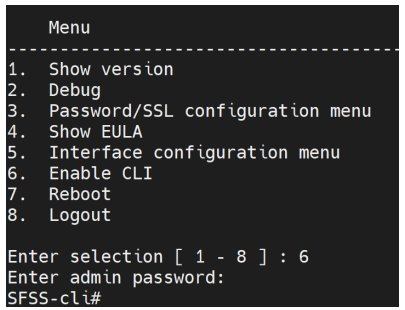
Alternative SFSS Configuration Methods

ANSIBLE

- https://galaxy.ansible.com/dellemc/sfss
- https://github.com/ansible-collections/dellemc.sfss

{RESTful API}

- https://developer.dell.com/apis/13238/



SFSS-CLI: Commands are in the SFSS User Guide

- https://www.dell.com/support/manuals/en-us/dell-emc-smartfabric-storage-software-trial/sfss-120-user-guide/sfss-command-line-interface?guid=guid-01ffe54f-02ab-458e-99d3-301288babc31&lang=en-us

Resources

Deployment Resources

- SFSS Interactive Demo
- SmartFabric Storage Software Deployment Guide
- SmartFabric Storage Software Trial Download (login required)
- Nvme-stas (Dell maintained open source discovery client for Linux)

Support Matrices

- eLabs Navigator: NVMe/TCP Switch Interoperability
- SmartFabric Storage Software (SFSS) Interoperability Matrix

Other Resources

- NVMe IP SAN Knowledge Center
- BLOG: The Future of Software-defined Networking for Storage Connectivity
- NVMe, NVMe/TCP, and Dell SmartFabric Storage Software Overview - IP SAN Solution Primer
- NVMe-oF Looking beyond performance hero numbers
- SNIA NSF – Discovery Automation for NVMe IP-Based SANs
- SNIA SDC – Discovery Protocol deep dive
- SANdbox Developer enablement portal for NVMe-oF

Q&A

After this Webinar

- Please rate this webinar and provide us with your feedback
- This webinar and a copy of the slides are available at the SNIA Educational Library https://www.snia.org/educational-library
- A Q&A from this webinar, including answers to questions we couldn't get to today, will be posted on our blog at https://sniansfblog.org/
- Follow us on Twitter @SNIANSF

Thank You

