SNIA ESF | ETHERNET STORAGE

Everything You Wanted to Know about Storage, but Were too Proud to Ask

Part Sepia

Getting from Here to There Pod

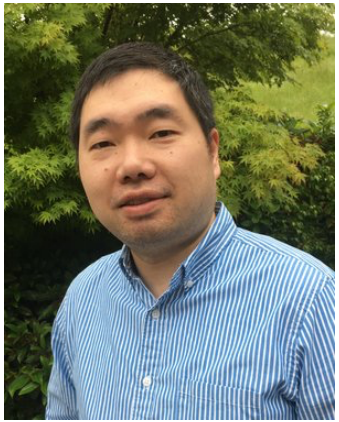
May 9, 2017
10:00 am PT

Today's Presenters

Chad Hintz
Cisco

John Kim
Mellanox

Alex McDonald
NetApp



Fred Zhang
Intel

SNIA at a glance

SNIA ESF | ETHERNET STORAGE

160
unique member companies

3,500
active contributing members

50,000
IT end users & storage pros worldwide

Learn more: **snia.org/technical**

@SNIA

SNIA Legal Notice

SNIA ESF | ETHERNET STORAGE

- The material contained in this presentation is copyrighted by the 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
  - The 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 the 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.

Agenda

- Encapsulation vs. Tunneling
- IOPS vs. Latency vs. Jitter
- QoS

Encapsulation vs. Tunneling

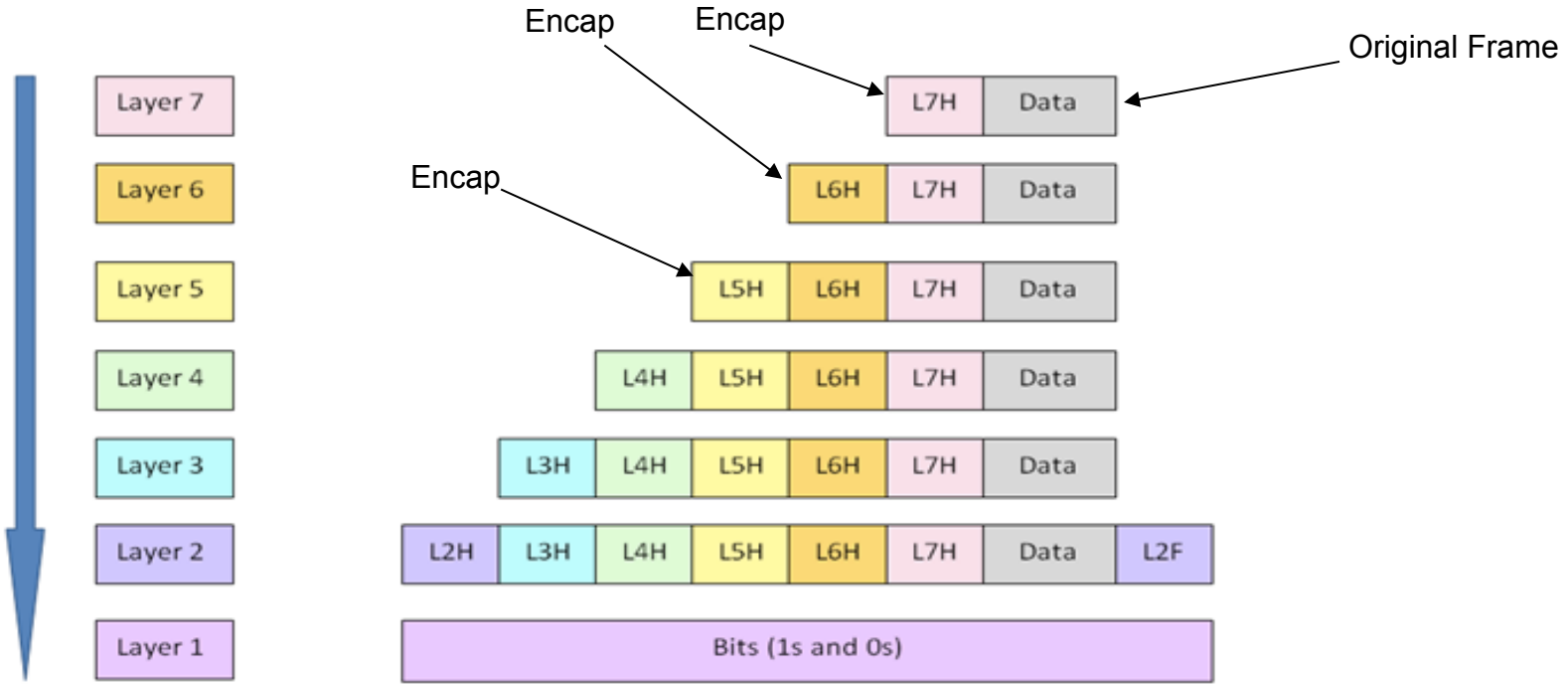
Definitions to start

- **Encapsulation:** Encapsulation is the process of encapsulating the payload with an additional header so that it can be sent (tunneled) through the intermediate network correctly. After the transmission, the encapsulated payload needs to be de-encapsulated at the routing end point and can be forwarded to the final destination
- **Tunneling:** Tunneling is a method used to transfer a payload of one protocol using an internetwork transportation medium of another protocol.

Well that explains it right?

Clear as Mud?

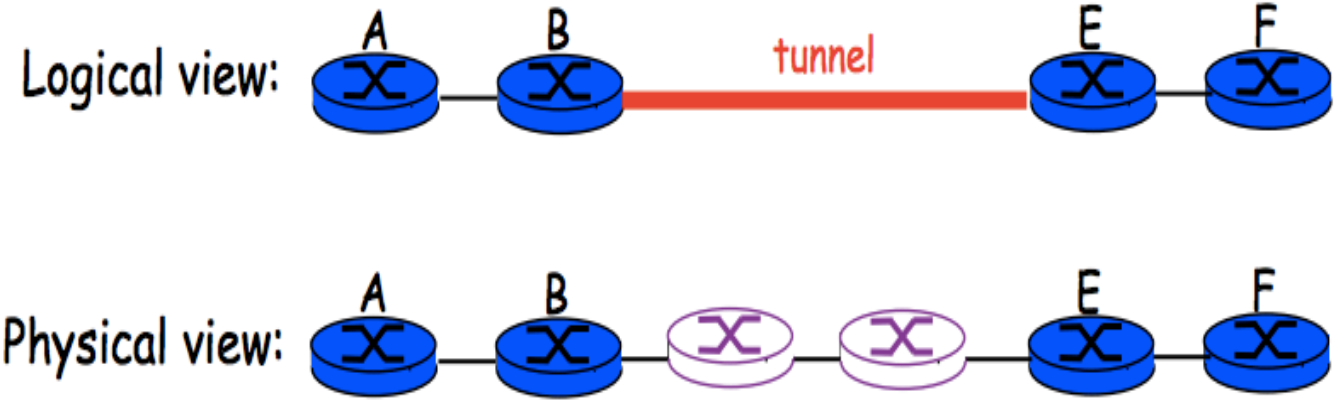
Deeper Dive-Encapsulation



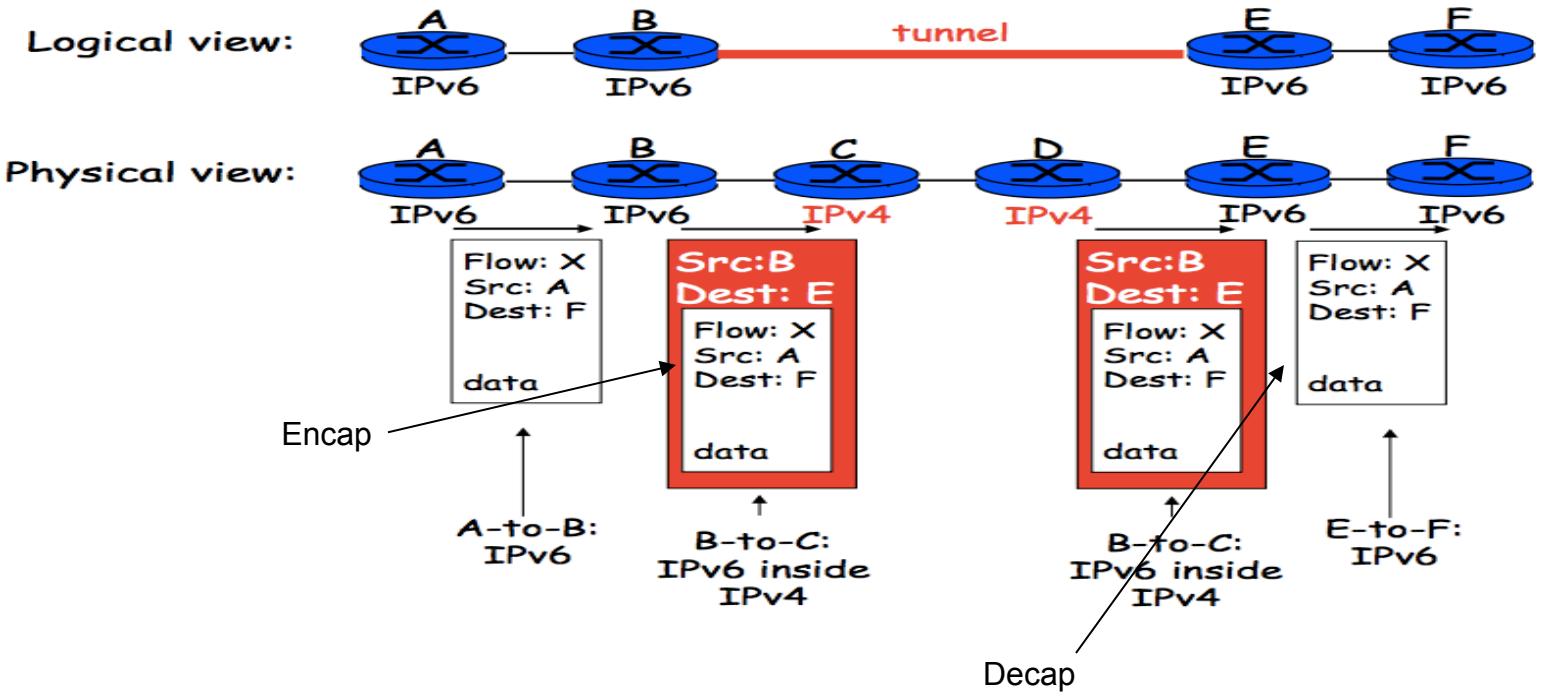
Encapsulation Summary

Taking a original data unit (unchanged)
then wrapping / encapsulating it in another frame.

Deeper Dive-Tunneling



Putting them together

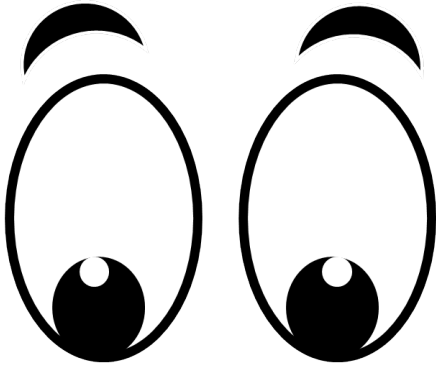


Summary of Encapsulation vs. Tunneling

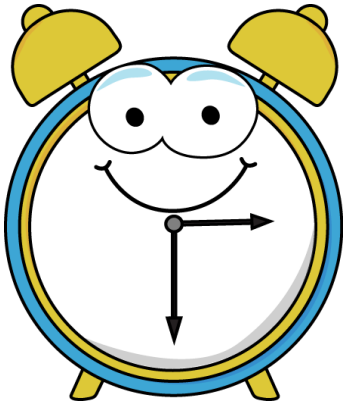
Tunneling is a method used to transfer a payload of one protocol using an internetwork infrastructure of another protocol. Encapsulation is the process of encapsulating the frame with an additional header so that it can be sent (tunneled) through the intermediate network correctly. Tunneling is referred to the whole process of encapsulation, transmission and de-encapsulation, while encapsulation is only a step within this entire process

IOPS vs. Latency vs. Jitter

IOPS <*eye-ops*>—Definition



- Input/output Operations per Second
  - Completed storage transactions
  - Read, write, or both; Block or file
  - Different ways to measure different workloads
- Who Wants Lotsa' IOPS?
  - Random workloads, usually small block
  - Databases, Virtual machines (I/O blender)
  - Some technical computing apps



What Affects IOPS?

- Performance of media (HDD, SSD, DRAM, etc.)
  - Different devices have different IOPS limits
- Latency of the data path
  - Combined latency of controller, network, media, and software
- How to increase IOPS
  - Increase IOPS capacity of controller, network, media
  - Reduce latency—more efficient software, caching, RDMA
  - Parallelism: Increase threads/jobs or queue depth

Latency—Definition

- Time to complete (or start) one transaction
  - Akin to storage reaction time
  - Lower is better → higher IOPS
  - Measure average or 99.xxx percentile
- Who needs low latency?
  - Database, VMs, technical computing
  - Typically small, random, workloads
  - Little effect on large or sequential I/O

High-latency transactions at the DMV

What Affects Latency?

- Total latency of the I/O path
  - Combined latency of client, controller, network, media
- How to lower latency
  - Reduce latency of client, controller, media, or network
  - Better software, caching, RDMA
- Is there a tradeoff with IOPS?
  - Generally lower latency = higher IOPS, but…
  - More parallelism can increase booth IOPS and latency

Jitter—Definition

- Variation or unpredictability in latency
  - Not average latency; Affected by “long-tail”
  - Often measured as 99.xxx percentile
  - Static vs. transient jitter
- Why is jitter undesirable?
  - High jitter makes performance unpredictable
  - Potential poor audio/video/web experience
  - Some apps cannot tolerate jitter

High jitter annoys network and storage administrators

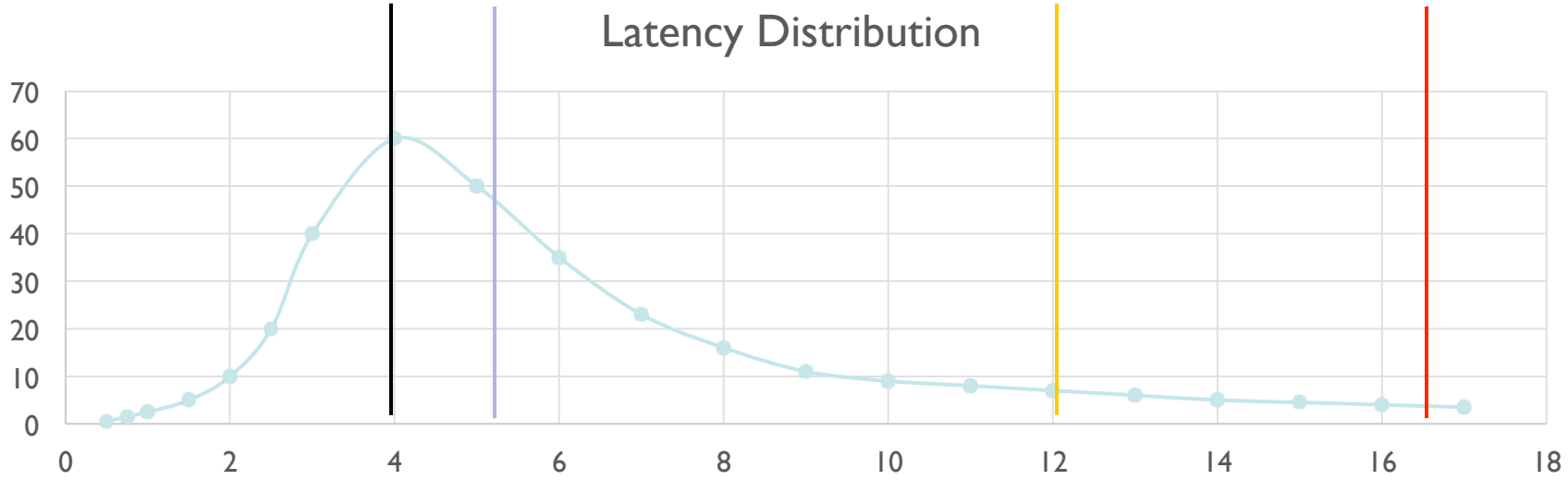
Long Tail Latency

Median Latency = 4ms

Mean (average) Latency = 5.2ms

99th Percentile Latency = 12ms

99.9th Percentile Latency = 16.5ms



What Affects Jitter?

- On the storage controller
  - Checkpoints, snapshots, replication, other housekeeping tasks
  - CPU too busy, Context switching, NUMA affinity
- On the media/drives
  - Recalibration, write coalescing, garbage collection
  - Write errors/retries, RAID rebuilds, media scrubbing
- On the network
  - Congestion, Packet loss, multi-pathing, flow control

How Do We Reduce Jitter?

- Storage and Media
  - Smarter software
  - More intelligent or integrated media management
  - Don’t overload the controllers
- Networks—Aim for deterministic latency
  - Higher bandwidth reduces congestion and packet loss
  - Lossless networks, congestion control, and/or QoS
  - Buffering reduces jitter but might increase average latency

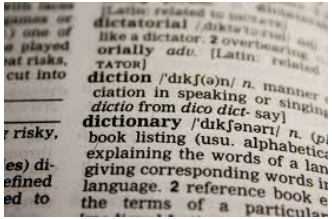
IOPS, Latency, Jitter—Summary

- IOPS measures storage transactions/second
  - Important for small I/O workloads
- Lower latency allows more IOPS
  - High latency bad for some workloads
- Jitter is variability in latency
  - Matters if you need deterministic latency

QoS

QoS Definitions

- Terms
  - Quality – normally meaning “fit for purpose”
  - Service – the data we requested
  - Rate limit or throttling – caps on quantity delivered per unit time
  - Burst rate – above average quantities demanded over brief periods
    - No definition of “above average” or “brief periods”
  - Noisy Neighbor – anyone you consider to be getting more than their fair share

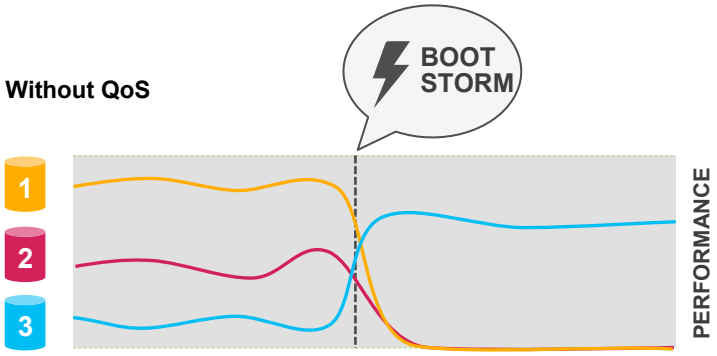


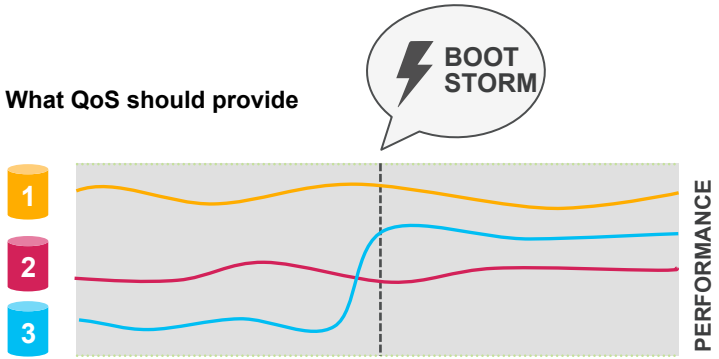
Storage Quality of Service

- QoS as a measurable characteristic of storage
  - Performance specifically related to bandwidth & latency
  - IOPS (I/O per second) often used as a proxy
  - Not covered here in this narrow definition
    - "End to end"; this discussion focused on storage element
    - Uptime, recovery time, meantime to fix (MTTF), availability, etc…
- Meeting QoS commitments
  - Ability to satisfy needs of the end-user
  - Application (or application type) as a proxy for end-user

Quality of Service

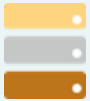
QoS	Methodology	Narrative
	No QoS • Try to provide enough IOPS for any combination of application workload	• No guarantees for multiple workload environments • No protection from “noisy neighbor” applications • Overcommit of resources can be expensive



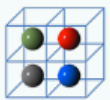
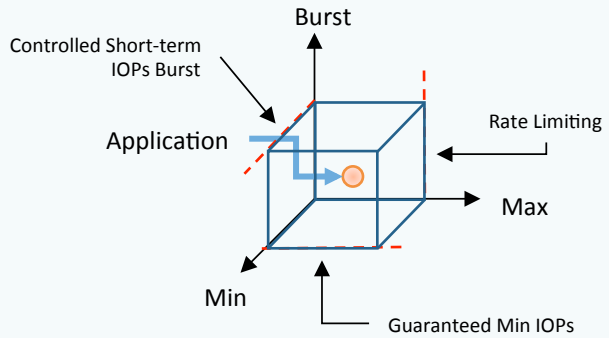


Different QoS techniques

SNIA ESF | ETHERNET STORAGE

QoS	Methodology	Narrative
	**Storage** Tiering (aka Gold, Silver, Bronze) • Combine different **storage types** to create different performance and capacity tiers	• Workload performance varies greatly as algorithm moves data between storage types (fast to slow, etc.) • "Noisy neighbor" applications steal performance • No control over individual applications • Manual placement & migration of data often superior but tedious
	Prioritization of **applications** (aka class of service) • Ranking of **applications** into tiers such as "mission critical," "moderate," and "low"	• Lack of control over any single application getting the performance needed • Performance is based on arbitrary levels • "Noisy neighbors" get worse if prioritized as "mission critical"

Different QoS techniques

QoS	Methodology	Narrative
	**I/O** rate limiting by **application** • **I/O limits** are applied to individual **application** performance	• Limits amount of performance a “noisy neighbor” application can access • No minimum performance guarantee • Done at application/hypervisor or somewhere in the storage layer
	**Minimum, maximum & burst** I/O by **application** • Individually controlled QoS levels • “Box” application into performance behavior	

Summary: Other QoS Techniques

- Caching
  - Very similar considerations as for storage tiering
  - Storage normally unaware of application/client caching
- Network bandwidth management & traffic shaping
  - Usually a function of the network
  - Whole end-to-end network management difficult for storage system to provide; part of an overall QoS solution
- Compression
  - Storage largely unaware of application/client & network compression

Other Storage Terms Got Your Pride? This is a Series!

SNIA ESF | ETHERNET STORAGE

Don't miss our next one – July 6, 2017

Everything You Wanted To Know About Storage But Were Too Proud To Ask – Vermillion

(What-If-Programming-and-Networking-Had-A-Baby)

- Storage APIs and POSIX
- Block, File, and Object Storage
- Byte Addressable and Logical Block Addressing
- Log Structures and Journaling Systems

Register at https://www.brighttalk.com/webcast/663/260321

Other Storage Terms Got Your Pride? This is a Series!

SNIA ESF | ETHERNET STORAGE

- Check out previously recorded webcasts:
- Teal – Buffers, Queues and Caches https://www.brighttalk.com/webcast/663/241275
- Rosé - All things iSCSI https://www.brighttalk.com/webcast/663/244049
- Chartreuse - The Basics: Initiator, Target, Storage Controller, RAID, Volume Manager and more https://www.brighttalk.com/webcast/663/215131
- Mauve Architecture: Channel vs. Bus, Control Plane vs. Data Plane, Fabric vs. Network https://www.brighttalk.com/webcast/663/225777
- More to come follow us @SNIAESF for announcements

Speaking of Series…Check out Storage Performance Benchmarking

SNIA ESF | ETHERNET STORAGE

- Storage Performance Benchmarking:
  1. Introduction and Fundamentals
  2. Solution under Test
  3. Block Components
  4. File Components

Watch them all on-demand at:
http://www.snia.org/forums/esf/knowledge/webcasts-topics

After This Webcast

SNIA ESF | ETHERNET STORAGE

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Ethernet Storage Forum (ESF) website and available on-demand at www.snia.org/forums/esf/knowledge/webcasts
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-ESF blog: sniaesfblog.org
- Follow us on Twitter @SNIAESF
- Need help with all these terms? Download the SNIA Dictionary http://www.snia.org/education/dictionary