SNIA ESF | ETHERNET STORAGE

EVERYTHING YOU ALWAYS WANTED TO KNOW ABOUT STORAGE, BUT WERE TOO PROUD TO ASK

Part Rosé

The iSCSI Pod

March 2, 2017
10:00 am PT

Today's Presenters

Loy Evans
Cisco

Rob Davis
Mellanox

Alex McDonald
NetApp

J Metz
Cisco

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

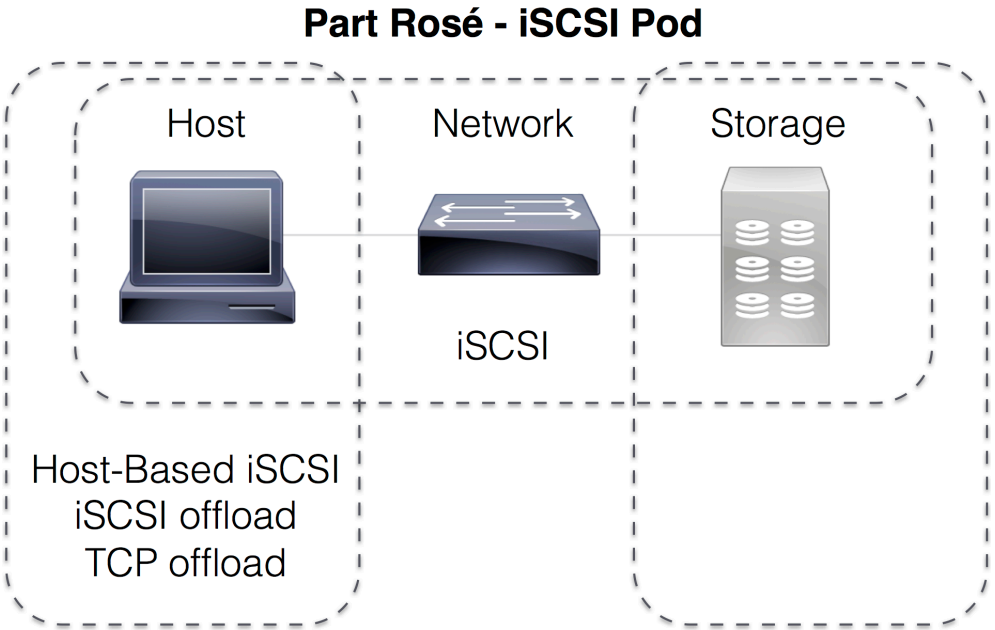
SNIA Legal Notice

- 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.

Today's Agenda

- Brief iSCSI Overview
- Host-based iSCSI
- iSCSI & TCP Offload



SNIA ESF | ETHERNET STORAGE

A Brief iSCSI Overview

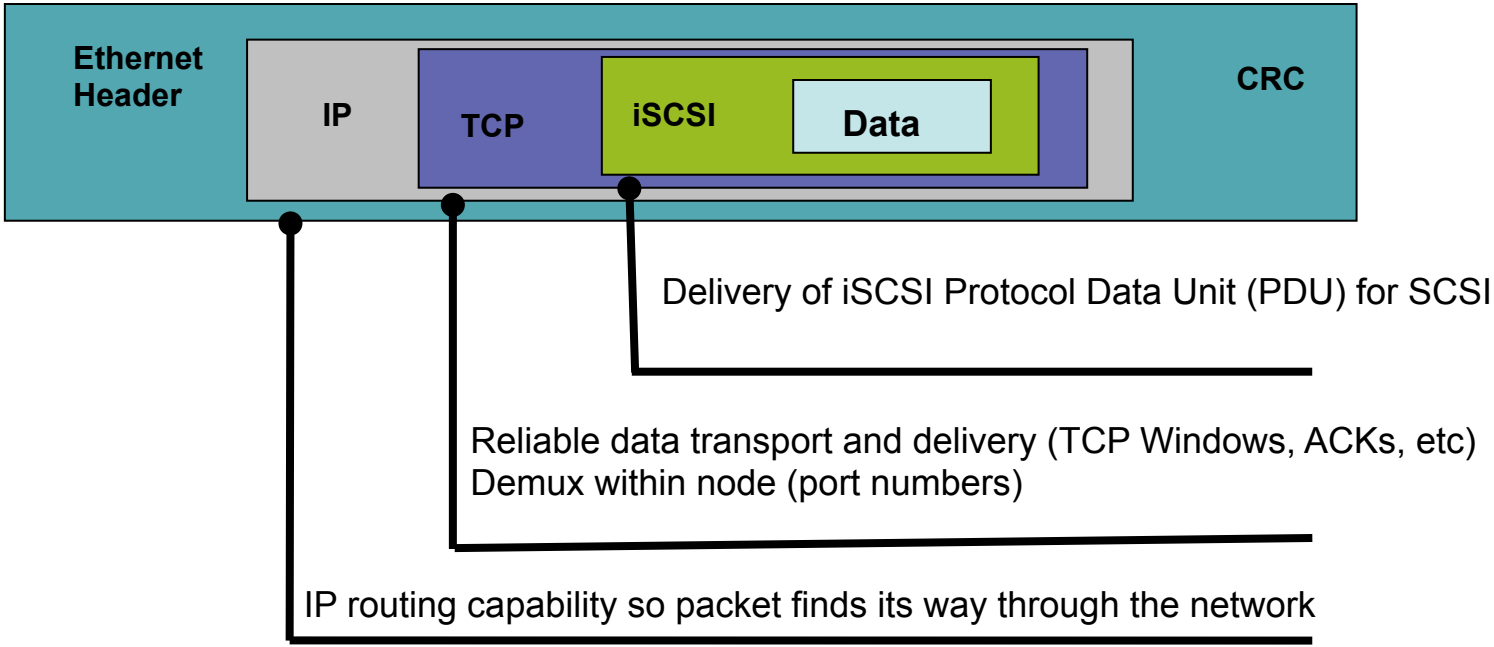
Alex McDonald

iSCSI History

- SCSI (Small Computer System Interface)
  - Originally designed for communication with various devices
  - Now exclusively for disk-like devices
- Fibre Channel (FC) drove separation of device vs. transport
  - Commands for devices vs. how the commands are transmitted
- iSCSI approach: Use TCP/IP as basis for transport
  - Only mainstream SCSI transport that does no hardware definition
- Defines block storage over a standard (TCP/IP) network

How iSCSI works

- Encapsulates SCSI commands in TCP/IP packet

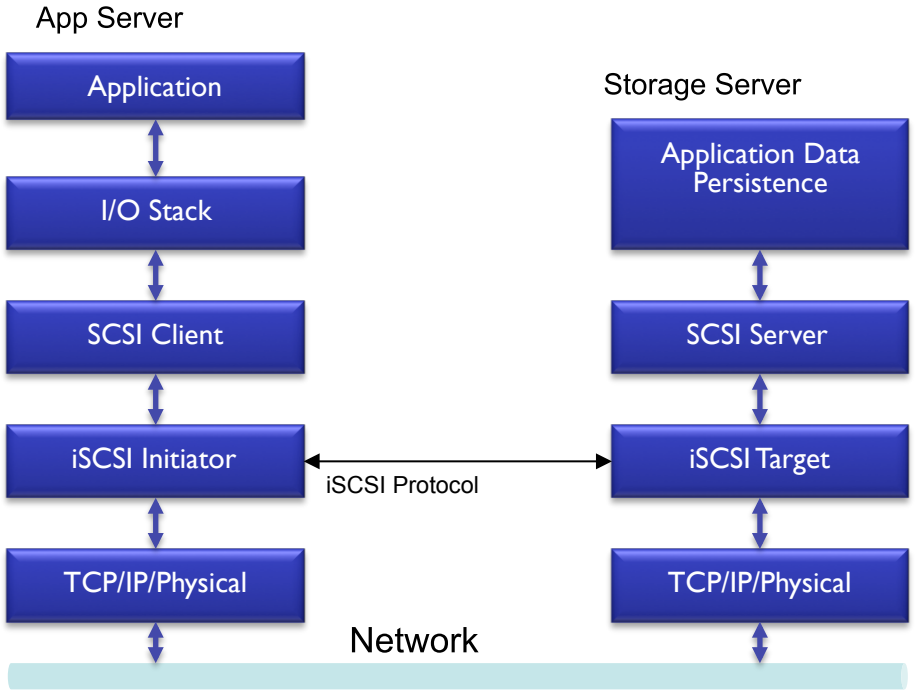


SCSI Concepts

- Request / Response protocol
  - There can be no response until there is a request
  - INITIATORS are where requests are created
  - TARGETS are where requests are serviced and responses created
- SCSI INITIATORS are usually hosts
  - Compute equipment, like servers or workstations
  - But hosts can also be targets
  - iSCSI initiators can maintain multiple parallel connections to multiple targets

iSCSI?

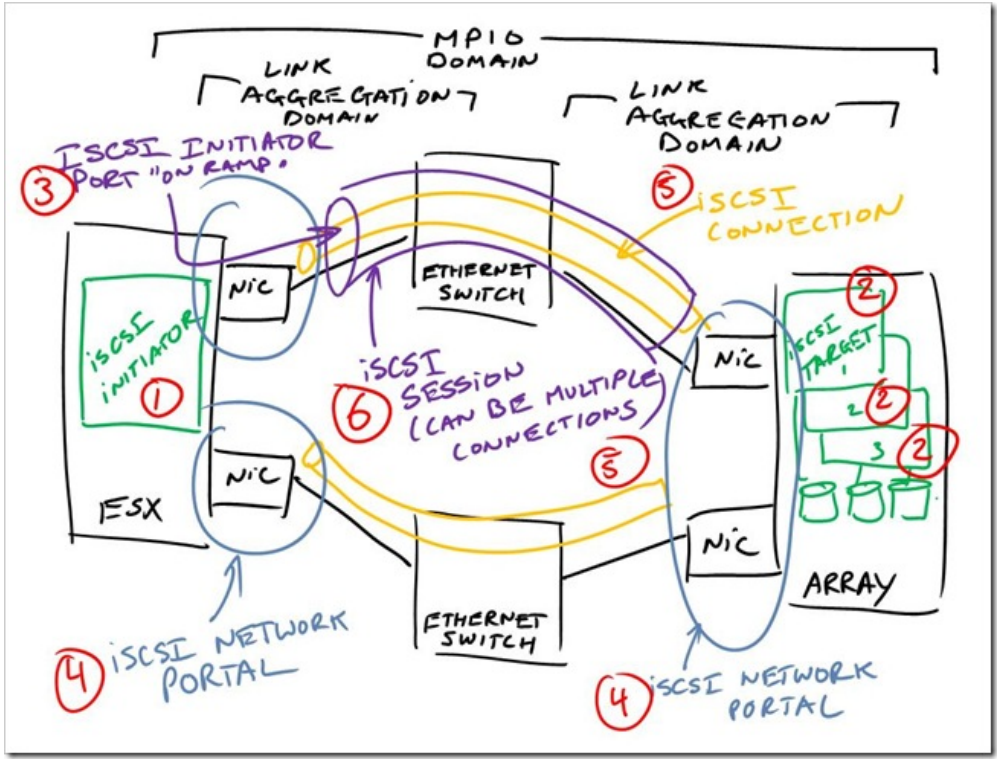
- iSCSI is a client-server SCSI transport protocol
  - iSCSI can run on any physical network that TCP/IP can run on – Ethernet, InfiniBand,..
- Any type of SCSI device can be accessed over iSCSI
  - Block Storage is the most typical (and the only type supported on Windows Server)
- Original protocol spec is RFC 3720
  - RFC 5048 corrects/clarifies the original
  - RFC 7143 replaces the original



Sessions, Connections and Target Portals

SNIA ESF | ETHERNET STORAGE

1. Initiator
2. Target
3. Initiator Port
4. iSCSI Network Portal
5. iSCSI Session
6. iSCSI Connection



Advanced: other SNIA resources

- Some topics we won't cover today
  - iSER: iSCSI Extensions for RDMA (Remote Direct Memory Access)
  - SR-IOV & MR-IOV: Single & Multi Root I/O Virtualization (PCIe bus specification for virtualization support)
  - RoCE, iWARP and other networking technologies
- Huge amount of advanced topic SNIA material available
  - Search on Google for keywords using `site:snia.org`
    - `site:snia.org iscsi rdma`

SNIA ESF | ETHERNET STORAGE

Host-based iSCSI

Rob Davis

What is Host-Based iSCSI?

SNIA ESF | ETHERNET STORAGE

- Host-based iSCSI uses software to implement iSCSI.

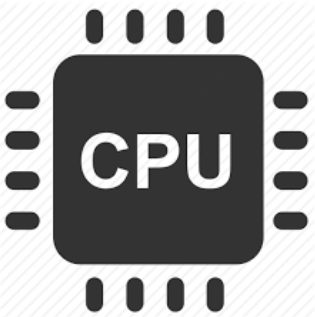
  Typically, this happens in a kernel-resident device driver that uses the existing network card (NIC) and network stack to emulate SCSI devices for a computer by speaking the iSCSI protocol. Software initiators are part of all popular operating systems and are the most common method of deploying iSCSI.

iSCSI Protocol Stack

Components of iSCSI stack that can be “host based” or “offloaded”

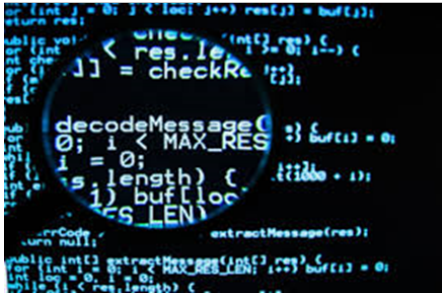
NIC

Host Based iSCSI = Onload



Onload

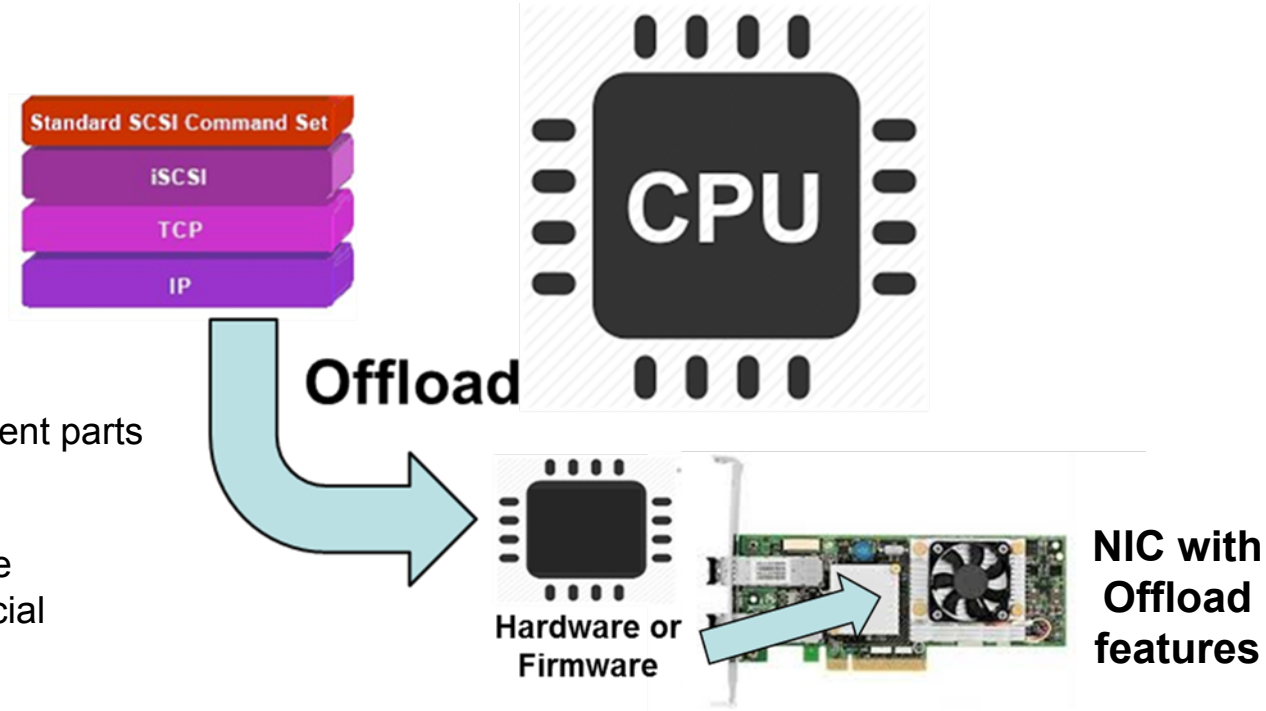
Software



With “host based” iSCSI the CPU executes the iSCSI stack software

Any NIC

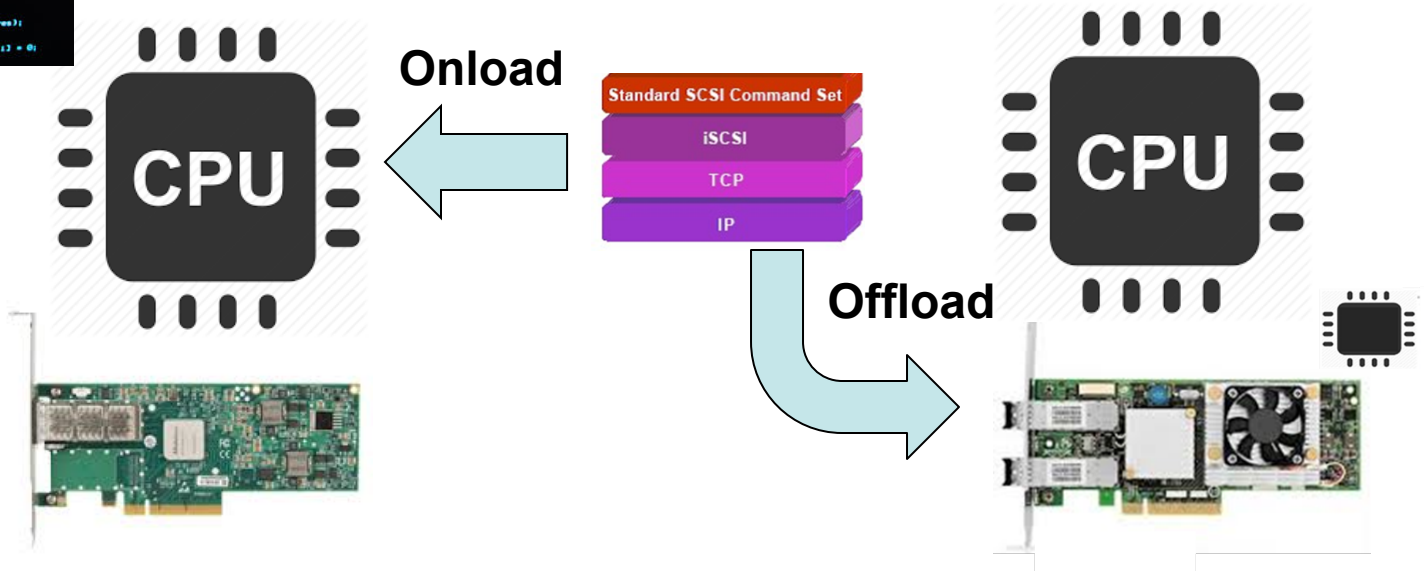
Offload



With “offload” different parts of the iSCSI stack, depending on the implementation, are executed on a special purpose NIC

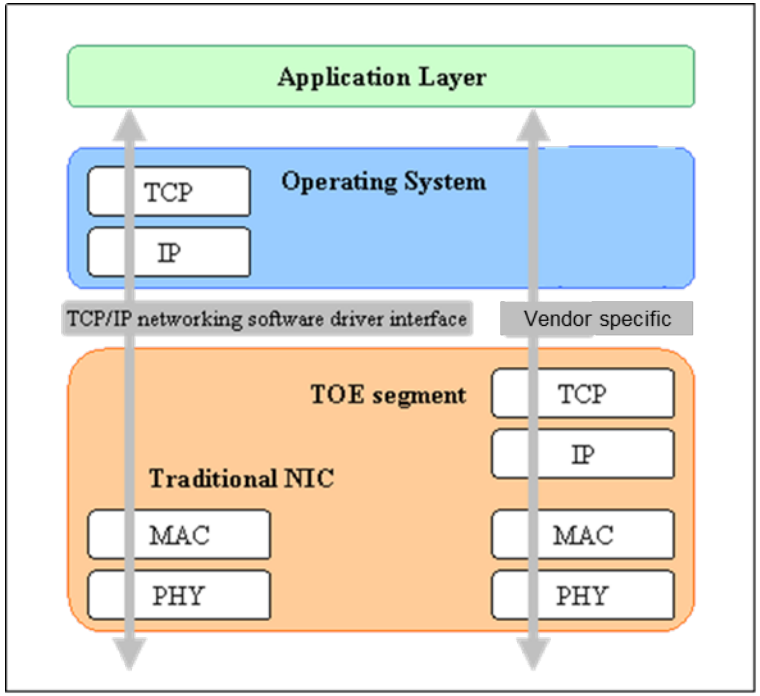
Onload/Offload

Side by side comparison, both approaches are successful in the storage market



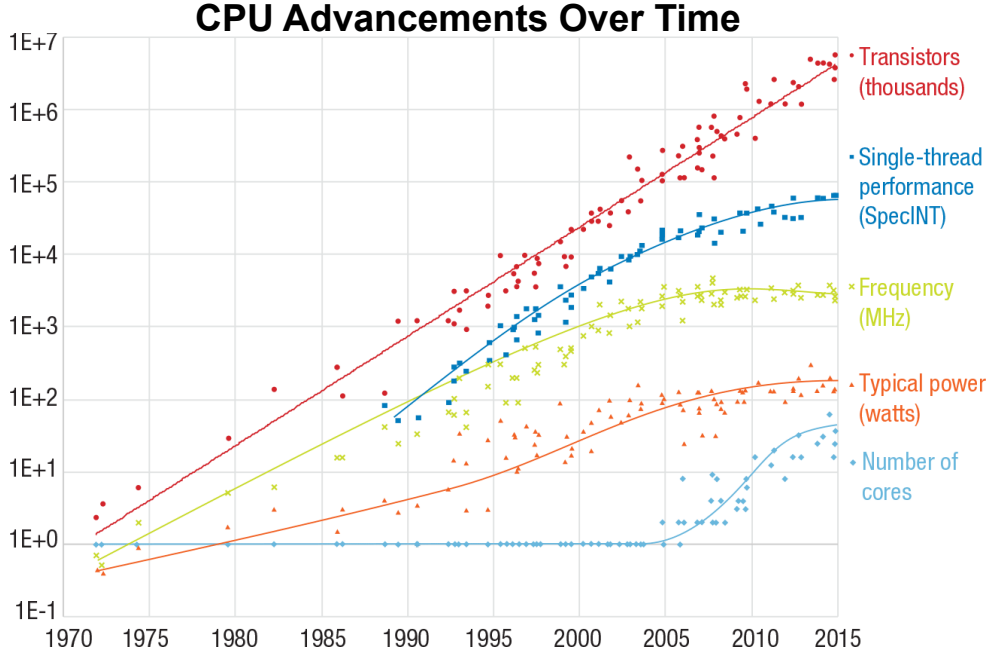
Host Based iSCSI Highlights

- Cost
  - Free
- OS based software
  - Ecosystem wide validation
  - Updates at all levels of stack
  - Innovation comes automatically
- No vendor lock in

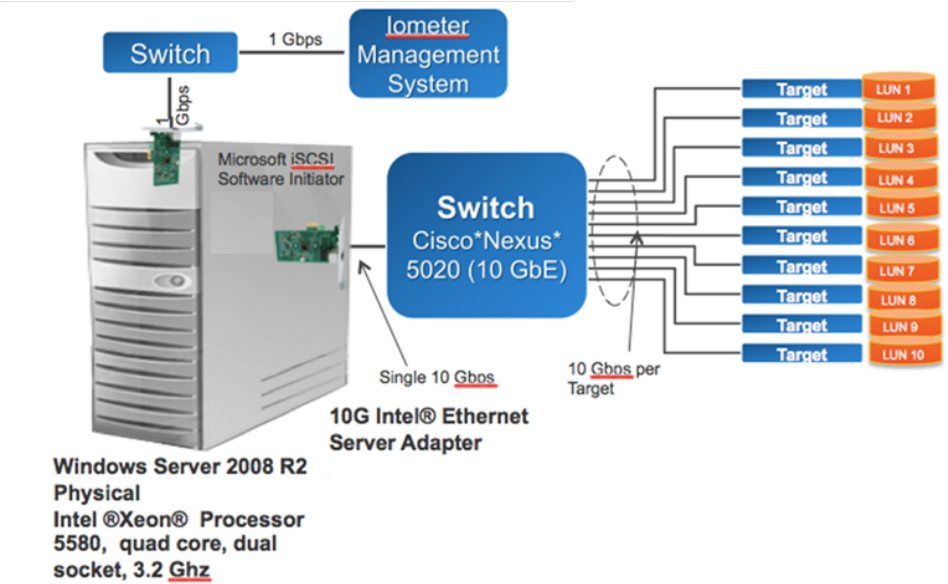


Host Based iSCSI CPU Needs Follow Advancements

- CPU resource usage
  - HP ProLiant DL380 Gen9
  - Xeon CPU E5-2680 V3 @ 2.8GHz, 12 cores, 3Q14
  - MTU 1500b
  - 128KB block size
- 100GbE – 25%
- 25GbE – 10%

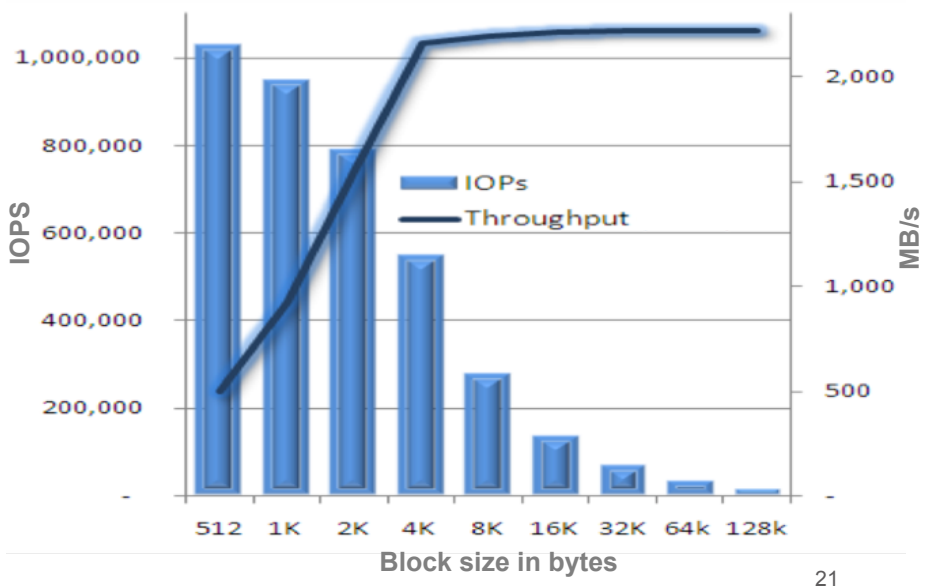


Host Based iSCSI Performance Advances with the CPU - 2010



2010 CPU Technology

1M IOPs with Host Based iSCSI

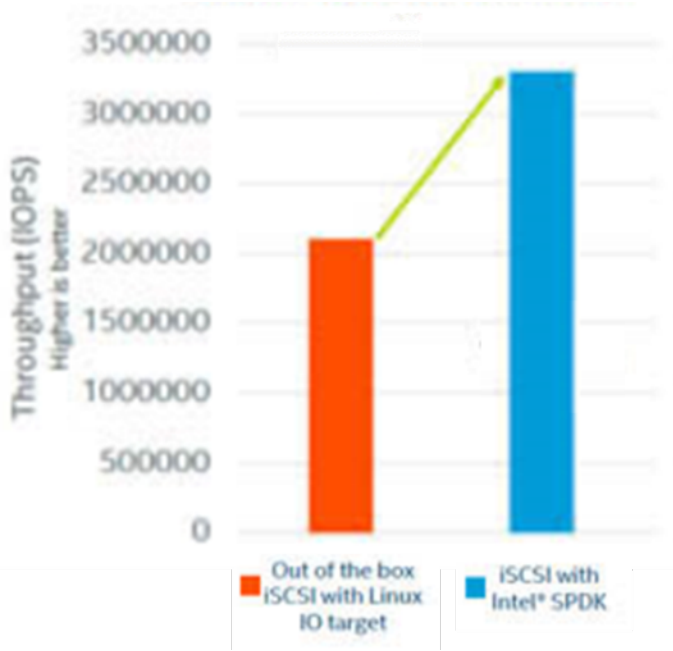


Host-Based iSCSI Performance Advances with the CPU - 2016

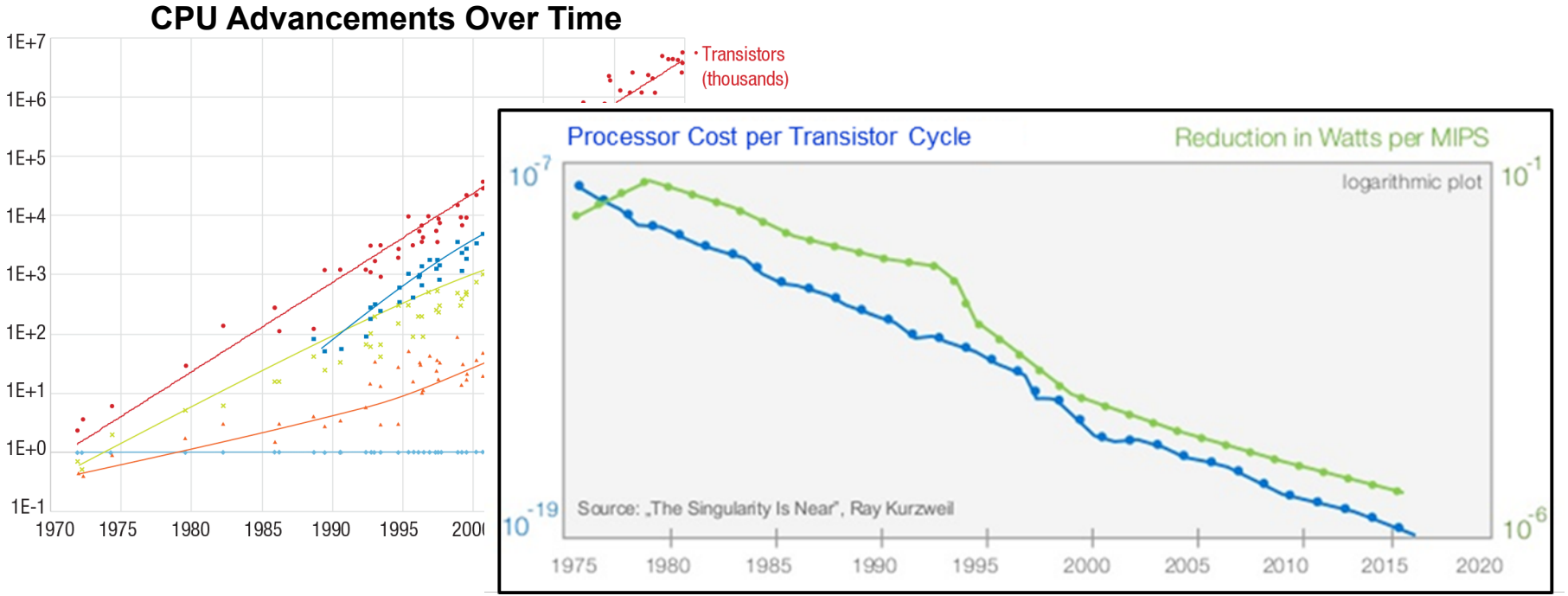
SNIA ESF | ETHERNET STORAGE

- 2016 CPU Technology
- Intel ES-2600 V4
- Host-Based Linux LIO Target
  - Over 2M IOPs
- Host-Based iSCSI with SPDK Target
  - Over 3M IOPs

3.3M IOPs with Host-Based iSCSI



Host-Based iSCSI Takes Advantage of Other CPU Advancements



SNIA ESF | ETHERNET STORAGE

iSCSI & TCP Offload

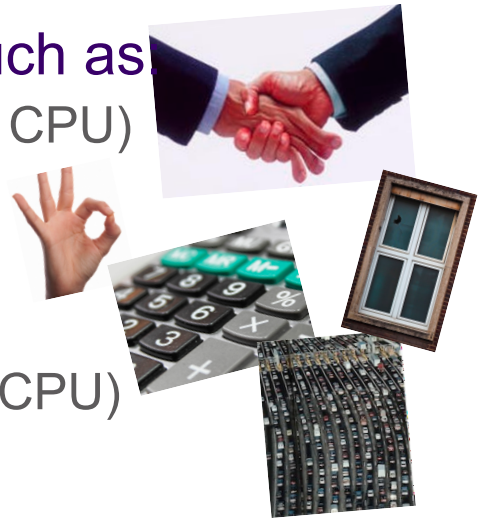
Loy Evans

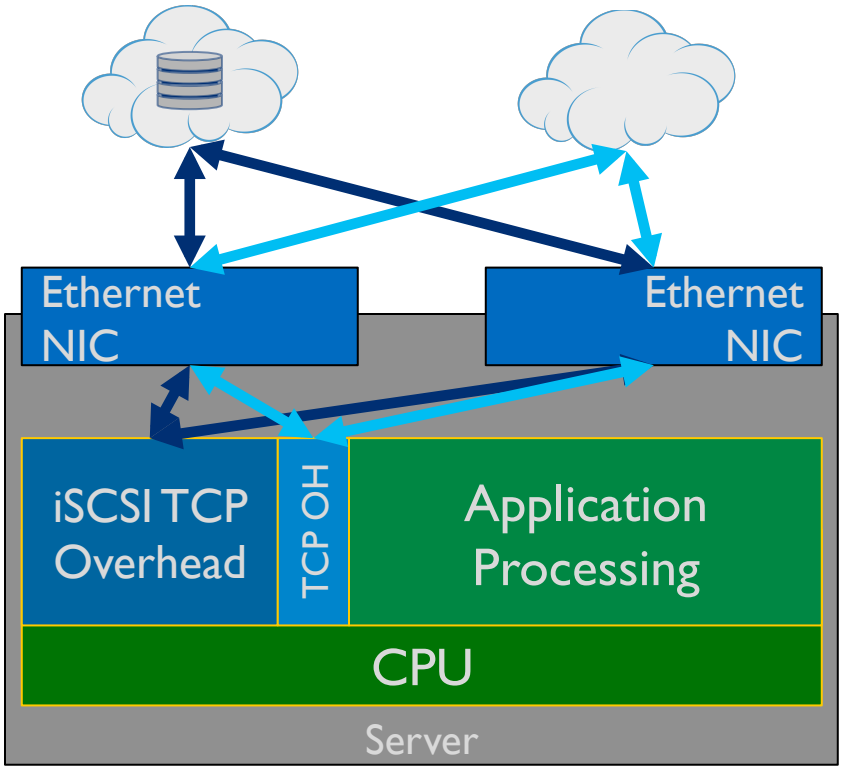
What is TCP Offload?

- **TCP offload** engine is a function used in network interface cards (NIC) to offload processing of the entire TCP/IP stack to the network controller
- By moving some or all of the processing to dedicated hardware, a TCP offload engine **reduces the load on main system CPU** for other tasks

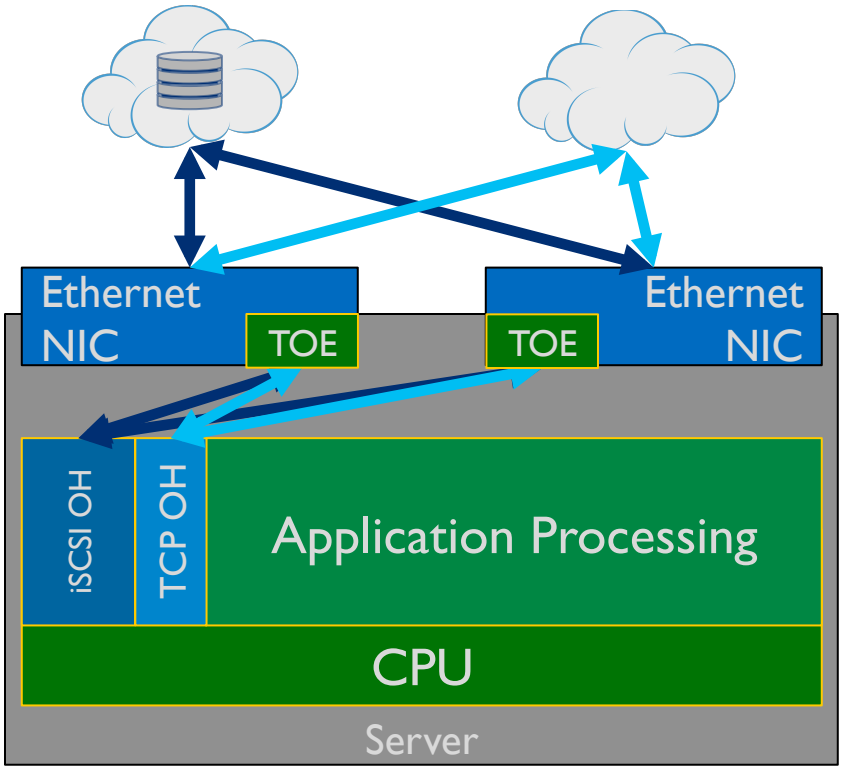
What is the problem here?

- TCP developed when connectivity was on **unreliable links**
- TCP designed to manage the connection & delivery to **compensate** for unreliable links
- TCP Management requires CPU **overhead**, such as:
  - Connection Establishment: 3-way handshake (low CPU)
  - On-going Acknowledgement (low CPU)
  - Window calculation (low CPU)
  - Checksum calculation (moderate to high CPU)
  - Congestion management & windowing (moderate CPU)





- Running an onload software iSCSI stack puts the burden of the extra TCP overhead on the OS and CPU
- The Overhead processed by the CPU depends on the type of workload, %age of iSCSI data to application CPU load
- An example: 10G NICs for iSCSI traffic, 1G NICs for Application traffic – if each interface were to run 100%, iSCSI interface would require 10x overhead processing



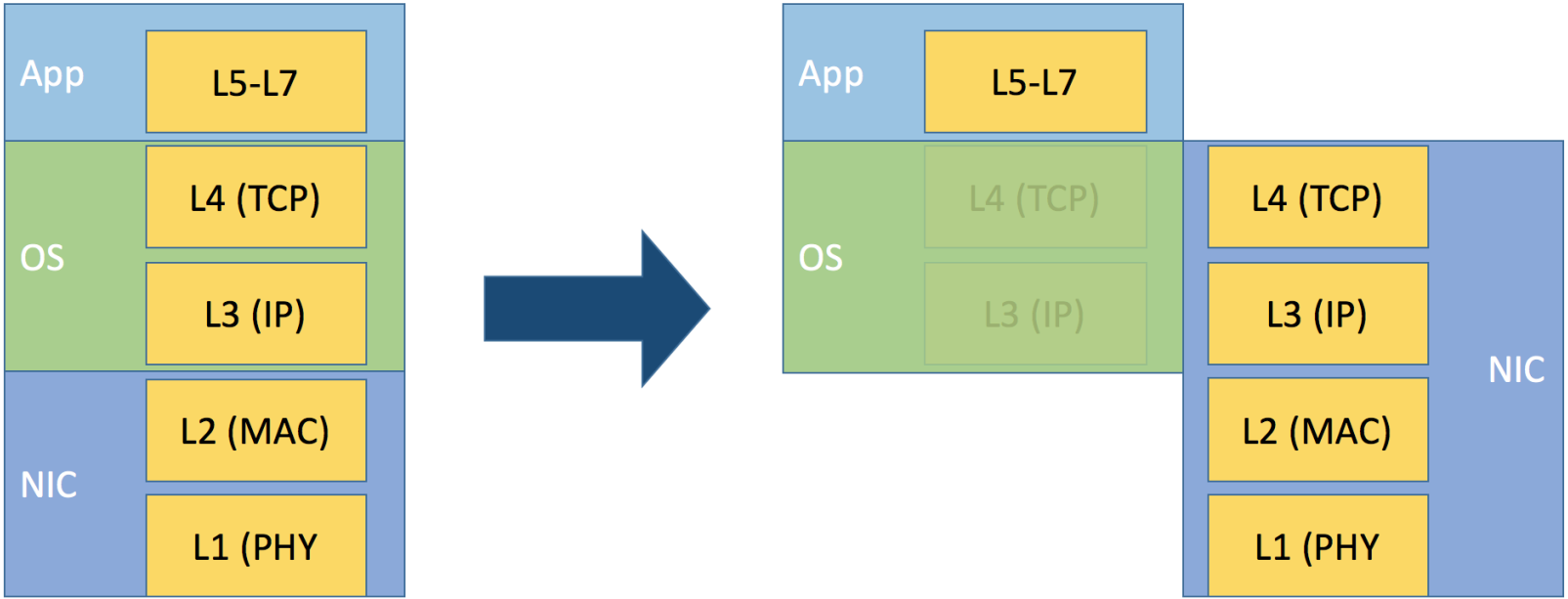
- With TCP Offload, the CPU only really has to deal with I/O interrupts
- NIC handles some % TCP overhead
- Depends on workload types
- Depends on traffic patterns
- Depends on what offload feature(s) used
- Savings: Your mileage may vary

TCP Offload – push the Work to the Network

(hardware)

SNIA ESF | ETHERNET STORAGE

- Full offload or Full stack offload
  - NIC runs full TCP/IP stack



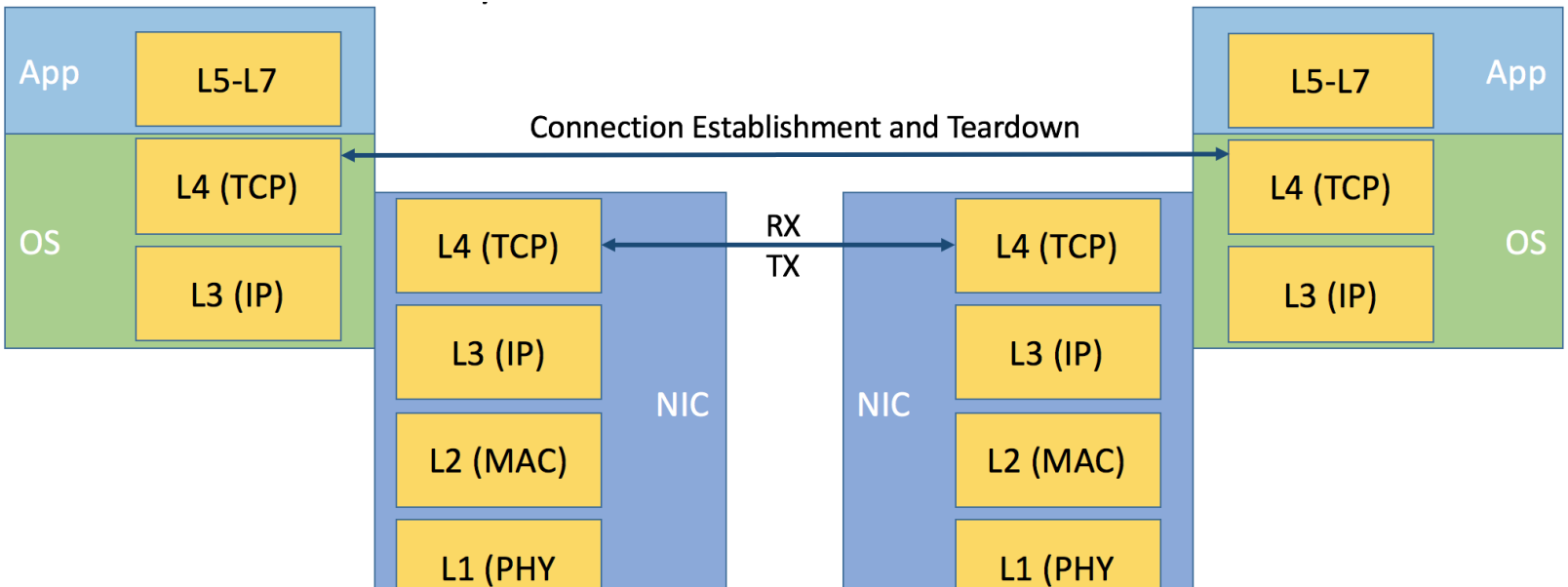
TCP Offload – push the Work to the Network

(hardware)

SNIA ESF | ETHERNET STORAGE

TCP Chimney Offload

- OS owns connection establishment/security & tear down
- Passes state to NIC, NIC handles all TX/RX

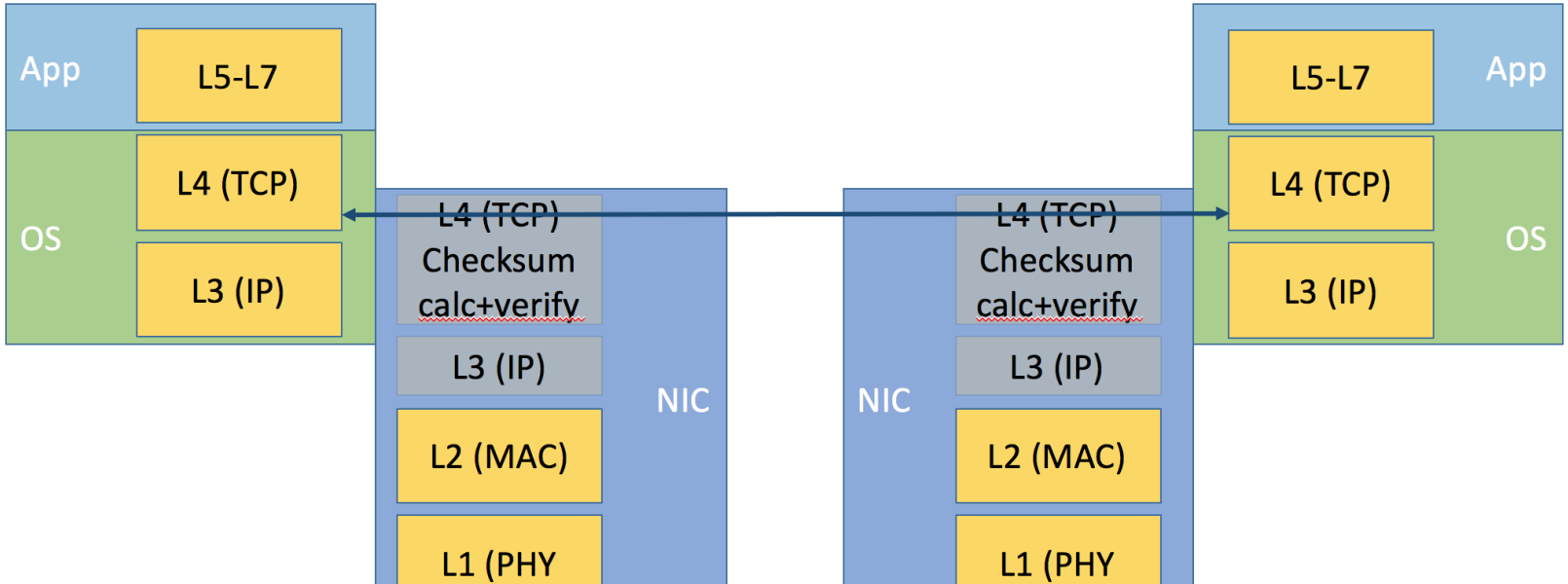


TCP Offload – push the Work to the Network
(hardware)

SNIA ESF | ETHERNET STORAGE

TCP Checksum Offload

- NIC calculates and verifies the checksum of each packet

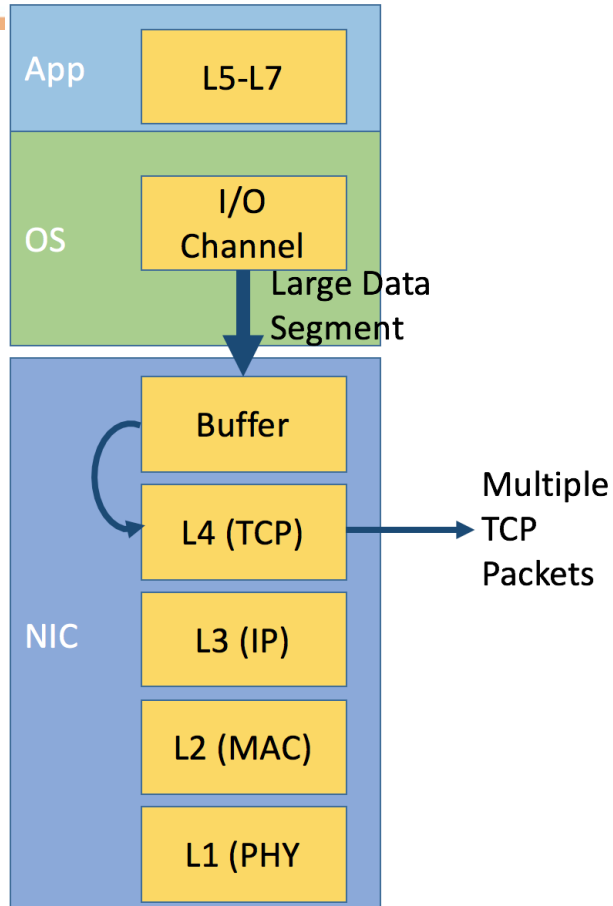


TCP Offload – push the Work to the Network

(hardware)

SNIA ESF | ETHERNET STORAGE

- Large Segment Offload (LSO)
  - AKA TCP Segmentation Offload (TSO)
  - Large Segments must be **broken down** into many smaller TCP/IP size packets
  - Without:
    - **CPU** calculates data segment boundaries
    - CPU breaks into multiple small segments
    - Passes small segments to NIC
  - With:
    - Large data segment passed to NIC
    - **NIC** breaks down to TCP/IP packet
    - Very little CPU overhead
  - Increases **outbound** throughput

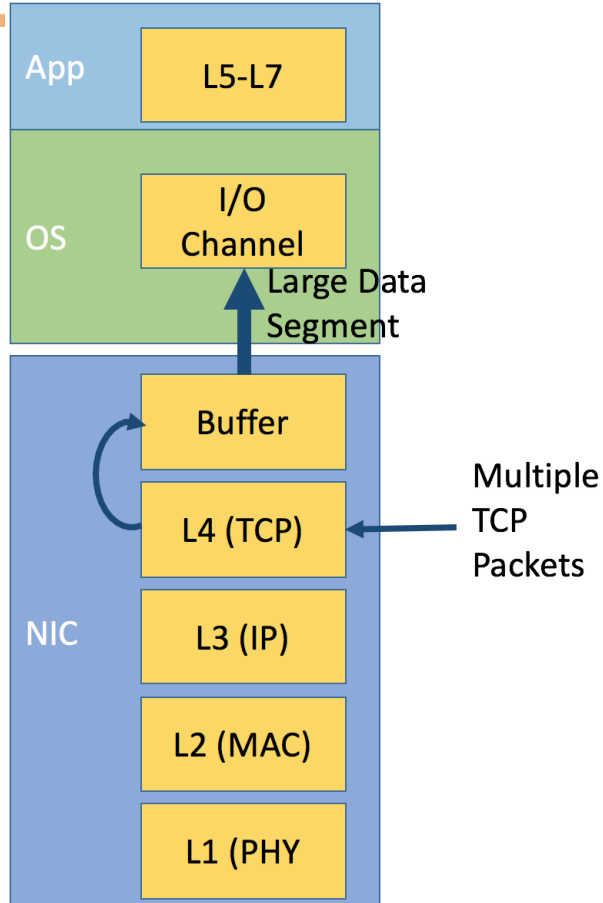


TCP Offload – push the Work to the Network

(hardware)

SNIA ESF | ETHERNET STORAGE

- Large Receive Offload (LRO)
  - Many small TCP packets must be **reassembled** into Large segments on the host
  - Without:
    - Host has to hold TCP connection open during transfer
    - Utilize **Host RAM** for buffer
    - Utilize **Host CPU** to reassemble
  - With:
    - **NIC** Aggregates multiple incoming TCP packets into a buffer
    - Pass larger payloads to the IO channel with a single interrupt
  - Increases **inbound** throughput



Speaking of Hardware

- TCP Offload Engine (TOE)
  - Chip on NIC that allows offloading of TCP stack functions from OS to hardware on NIC
  - **Full or partial** stack offload
  - One to many features in partial mode
    - Each feature is configurable through BIOS (card or board)
  - Can be used to offload TCP for more than just iSCSI
- iSCSI HBA/NIC
  - Specialized card specifically designed for iSCSI offload
  - Might not allow offload of non-iSCSI workloads

Conclusions

- Host Based iSCSI uses the system CPU to manage the iSCSI Software stack in OS using a standard Ethernet NIC
- Offloading uses hardware available on specialized NICs which can reduce the traffic management load on the CPU
- There advantages and disadvantages to both approaches
- In either case, iSCSI provides reliable and flexible Block Storage over existing Ethernet infrastructure with enterprise-grade performance

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

SNIA ESF | ETHERNET STORAGE

- Check out previously recorded webcasts:
- Evolution of iSCSI https://www.brighttalk.com/webcast/663/197361
- Chartreuse https://www.brighttalk.com/webcast/663/215131
  - The Basics: Initiator, Target, Storage Controller, RAID. Volume Manager and more
- Mauve https://www.brighttalk.com/webcast/663/225777
  - Architecture: Channel vs. Bus, Control Plane vs. Data Plane, Fabric vs. Network
- Teal https://www.brighttalk.com/webcast/663/241275
  - Buffers, queues and caches

Other Storage Terms Got Your Pride? Future Webcasts

- Future Topics/Colors (in no particular order):
- Vermillion (What-If-Programming-and-Networking-Had-A-Baby Pod)
  - Coherence/Cache Coherence, Storage APIs, Block, File, Object, Byte Addressable, Logical Block Addressing
- Turquoise (Where-Does-My-Data-Go Pod)
  - Volatile v. Non-Volatile v Persistent Memory, NVDIMM v. RAM v. DRAM v. SLC v. MLC v. TLC v. NAND v. 3D NAND v. Flash v SSDs v. NVMe, NVMe (the protocol)
- **More**…

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
- 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 2016 SNIA Dictionary http://www.snia.org/education/dictionary

SNIA ESF | ETHERNET STORAGE

Thank You!