Evolution of the System z Channel

Howard Johnson – hjohnson@Brocade.com
Patty Driever – pgd@us.ibm.com

8 August 2011 (4:30pm – 5:30pm)
Session 9934
Room Europe 7

Legal Stuff

- Notice
  - IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing to: IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.
  - Any references in this information to non-IBM Web sites are provied for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
- Trademarks
  - The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: FICON® IBM® RedbooksTM System z10TM z/OS® zSeries® z10TM
  - Other Company, product, or service names may be trademarks or service marks of others.

Abstract

- This session examines the evolution of the FICON channel from the birth of the industry to the looming converged infrastructure. The speakers will discuss the designs and modifications that allowed System z solutions to move from ESCON to FICON and beyond. Come see where the FICON channel has been and where it's going.

Agenda

- Yesterday
  - Review of the origins of the channel and its early evolution
    - Byte channel
    - ESCON channel
    - FICON bridge
- Today
  - Examine the current channel technology
    - Native FICON
    - High Performance FICON
- Tomorrow
  - Explore the future of channel technology
    - 16G / 32G FICON
    - Fibre Channel over Ethernet

Yesterday

The origins of the channel and its early evolution

The Early Years

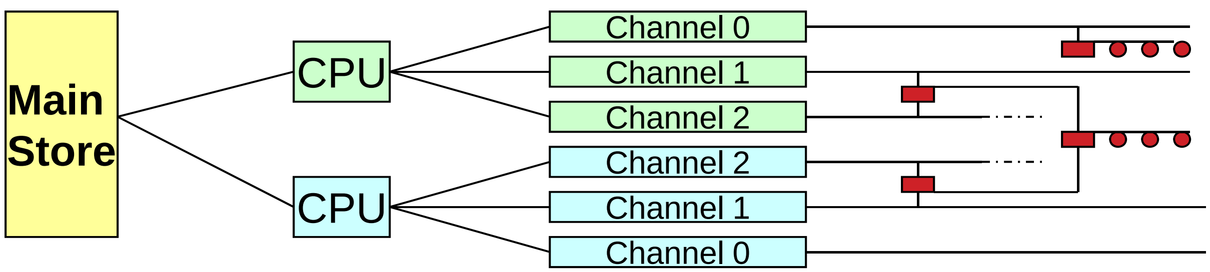
S/360 & S/370 I/O ARCHITECTURE

S/360

- First real computer ***architecture***
  - Provided for ability to deliver a range of models with increasing price/performance characteristics
    - Fastest models used hard-wired logic, while microcode technology used to deliver wide range of performance within the S/360 family
  - Designed to provide foundation for application programs that could migrate forward across models/system
    - Application level compatibility maintained through today on System z processors (with few restrictions)
  - Channels were specialized processors driven by a special instruction set that optimized the transfer of data between peripheral devices and system main memory
    - Channel architecture defined mechanism to transfer data, but was independent of device architecture
  - S/360 systems had one **byte** channel (channel 0) and one to six **selector** channels
  - New family of I/O devices developed for S/360 that used new standardized 'bus & tag' interfaces

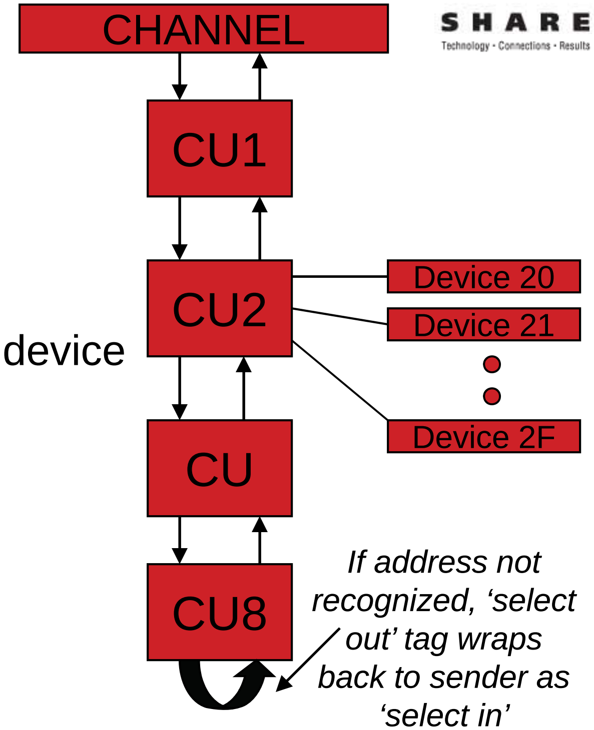
S/360 & S/370 I/O Architecture

- Designed to provide value in key mainframe focus areas: security, resiliency & performance
  - Security/Integrity
    - Host-based configuration definition methodology introduced in S/370
      - *Security controls based on host definitions (IOCDS)*
      - *4 digit device number = Channel + Unit Address*
    - CU enforced atomicity
      - *Reserve/release*
      - *Extent checking*
  - Resiliency
    - Channel set switching (S/370)
  - Performance
    - Bi-directional data transfers (reads/writes) and transfer of non-contiguous portions of the disk in a single I/O operation
      - *Command chaining*
      - *Data chaining*
      - *Indirect Data Address Words (S/370)*
    - Asynchronous notification for unsolicited events
      - *Busy/no longer busy status*



Parallel Channels

- Introduced with S/360 in 1964
  - Circuit connected (multi-drop CUs & devices)
  - 1 device at a time (max 256 devices per channel)
  - No dynamic switching
  - Initially support ~200 ft distances between channel and device
- Typical “bus & tag” channel/CU communications:
  - ***‘Initial selection’*** sequence to establish a connection with a control unit
  - ***‘Data transfer’*** sequence (CU always controlled data transfer)
    - Each byte of data sent requires a response
      - *Read: CU says “I’m sending a byte”, and channel says ‘I received a byte”*
      - *Write: CU says “I’m ready for a byte” and channel says “Here’s a byte”*
  - ***‘Ending sequence’*** (when one side recognizes that all of the available or required bytes have been transferred)
- Parallel channels are distance/speed limited, because of skew between the transmission lines in the channel
  - When a tag is received on one side, all of the bits of the data on the bus must be valid at that moment



Parallel Channels…initial types

- Byte Multiplex
  - Could have several devices performing a data transfer at a time because each device could disconnect between bytes of data transferred
- Selector
  - These were used with devices that have high data transfer rates, requiring the channel to remain connected to the device until the entire chain of CCWs is executed
  - Tape devices are typical examples

Parallel Channels…evolving types

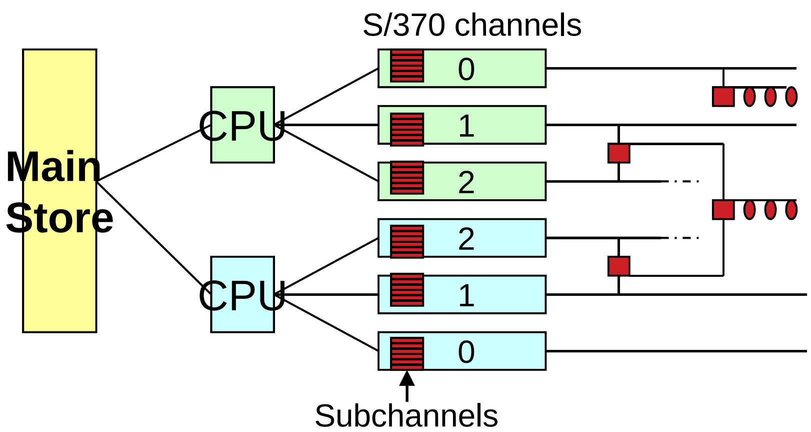
- Block Multiplex
  - Devices could disconnect only after the entire block of data for the command is transferred
    - When device is ready it presents ‘request in’ to request to be serviced again
  - Data streaming
    - CU could stream in ‘data in’ tags and count ‘data out’ tags
    - Removed the interlock between channel and control unit before next byte could be transferred
      - *Enabled 400 ft. distances*

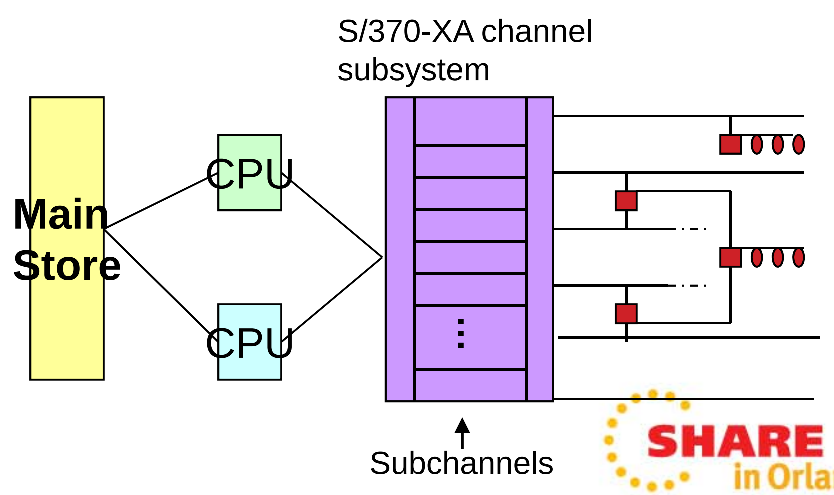
The winds of change

TRANSITION TO ESCON

S/370-XA (Extended Architecture)

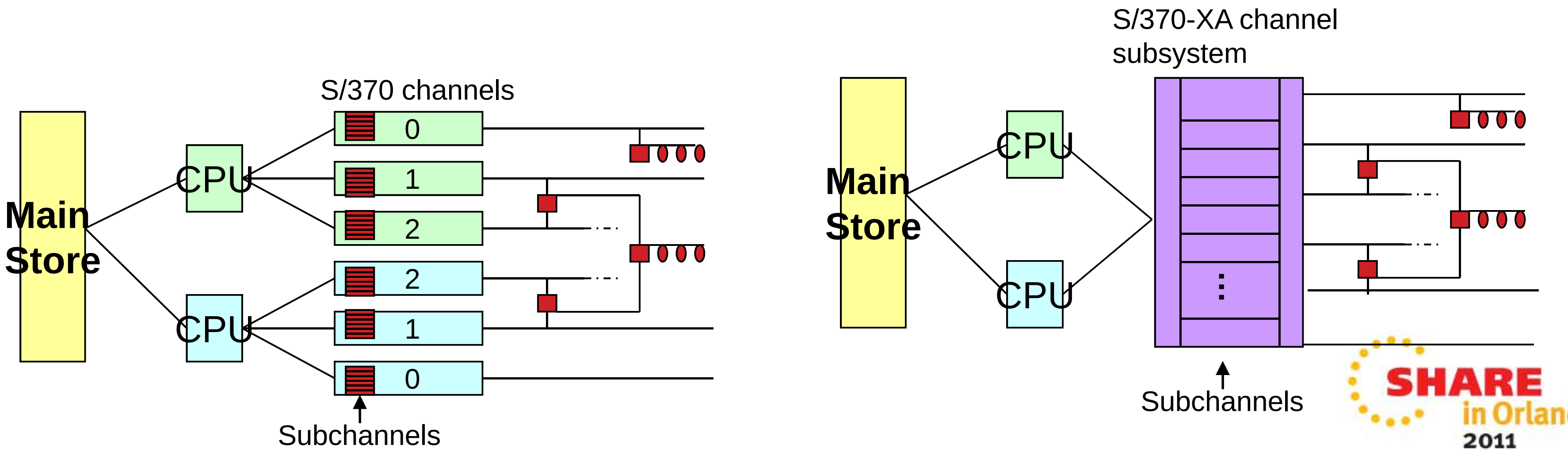
S/370	S/370-XA
Operating system selected the channel, and then the device on that channel	Operating system selected a device number (subchannel), and the channel subsystem knew all the addressing to get to that device and chose the best path
Channel can be addressed only by the single CPU to which it is connected and channel can interrupt only that CPU to which it is connected	Any CPU can initiate an I/O function with any device and can accept an I/O interruption from any device
Once a chain of operations is initiated with a device, same path must be used to complete transfer of all data, commands and status	Device can reconnect on any available path





S/370-XA (Extended Architecture)

- Enabled improved performance
  - All I/O instructions are executed asynchronously by the CPU with respect to the channel subsystem
  - All I/O busy conditions and path selection are handled by the channel subsystem rather than the CPU
    - Reduced CPU overhead in processing no-longer-busy conditions and encountering (possibly recurring) busy conditions on path selection
  - Eliminated program differences required to manage channels by type (e.g. selector vs. multiplexor)
  - Reduced the number of conditions that interrupted the CPU
    - Channel-available, control-unit-end, and device-end no-longer-busy eliminated



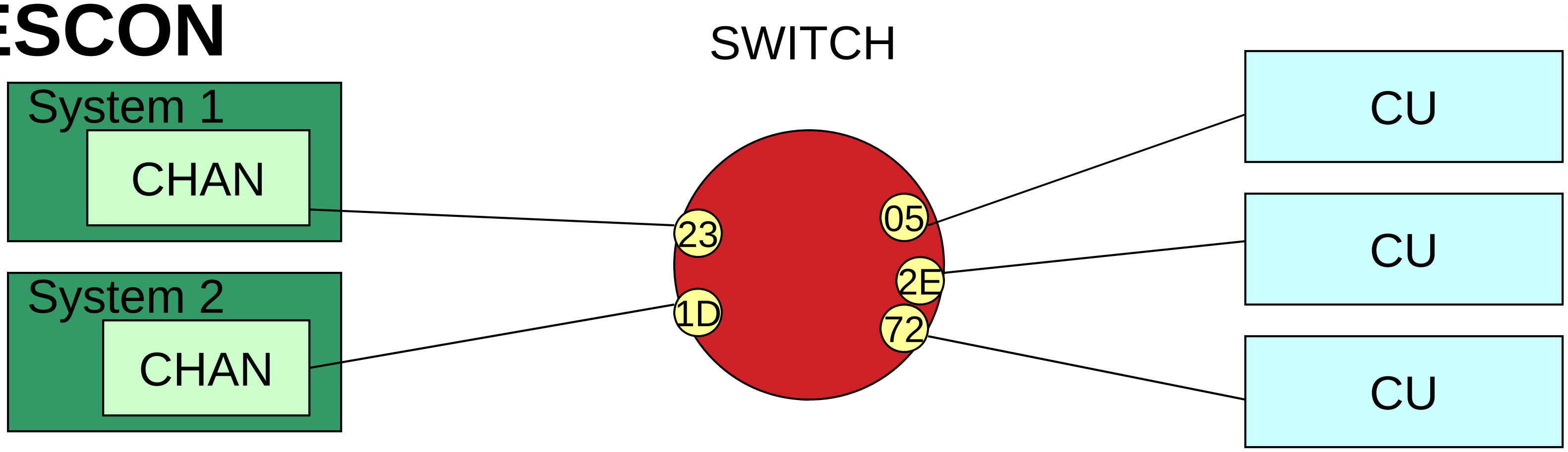
“A New Generation”

ESCON

ESCON

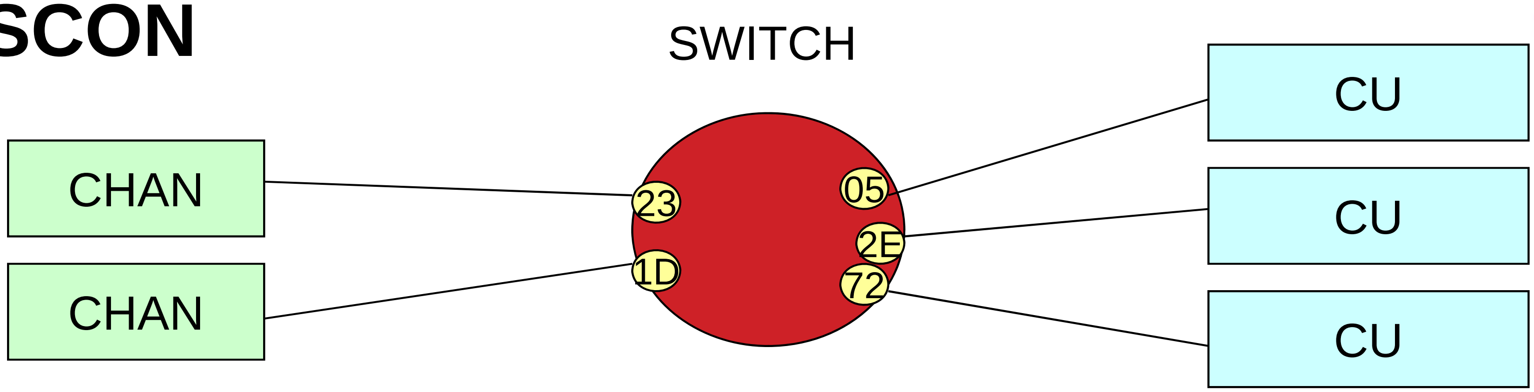
- Introduced in 1990 on the 3090 system
  - First supported device was a communication device
  - Followed closely by channel-to-channel support
  - Used fiber optic (serial) connectivity instead of copper
- Link technology supported 20MB/sec, but limited to ~10MB initially by the 3090 system bus
  - Achieved ~18MB/sec
- Supported introduction of the first modern SAN with dynamic switching capabilities
  - But still circuit-connect (communication with one device at a time)
- Distance limitations improved from 400ft to 9km (3km per link)
  - Serial transmission enabled greater distances between communication points
  - How far an LED could drive
- Supported 1000 devices per channel
  - Needed because multiple switch ports allow connection to multiple CUs
- I/O Configuration integrity checking with reset event architecture and self-describing devices (RNID)

ESCON



- Dynamic Switching
  - Enables multiple hosts to access the same storage device, with atomicity controlled by the device itself
  - When connection is established between two ports it appears as one continuous link
  - Switch can establish or remove a dynamic connection via information in frame delimiters or other sequences
  - Frame delimiter is at the beginning of a frame to indicate the establishment of a dynamic connection
  - Frame delimiter or sequence at the end of a frame to indicate whether to retain or remove the connection
- Fixed link addressing used in IOCDS controlled host access to CU resources

ESCON



Resiliency

- State Change Notification
  - Sent to each link level facility (channel or CU) which is potentially affected by state changes in switch ports or connected channels or control unit

Security

- Read Node Identifier
  - Enables host program to determine the specific physical device attached at the end of the link and revalidate attachments after link down conditions

Performance

- Streamlined command chaining
  - In parallel channels, when commands were chained together, after status was sent in for one command the channel would go through the selection sequence again
  - With ESCON, the channel responded to status directly with the new command
- Streamlined data transfer
  - On a 'Read' command, channel could request data based on its buffer size
  - On a 'Write' command, channel would indicate how much data could be sent before next data request was required?

The winds of change

TRANSITION TO FICON

Late 1990s - What Was Going on Inside IBM

- Evolution from System/360 to System/390 saw a significant increase in MIPS, main storage, and I/O capacity
  - I/O capacity had largely been improved through the continued addition of channel paths, thus adding cost, complexity, and overall bandwidth without significant improvement in the capacity of a single channel
    - Original 7 channels provided by S/360 evolved to the 256 channels provided by S/390 at that time
    - 256 was the architectural, programming, and machine limit
  - Projected processor MIPS growth along with improved controller technology and increased I/O densities also drove the need to significantly improve the I/O throughput over a single channel path

What was going on in the Industry

- ANSI Fibre Channel had its beginnings about 1988-9
  - Initial motivation was for higher bandwidth I/O channels that operated efficiently at fiber optic distances (10s of kms)
- FC proponents argued early on that the requirements and technologies of LAN and channel applications were converging, and that a layered architecture could deliver excellent performance, cost, and migration characteristics for many applications
  - By late 1990s FC found a toehold as a <u>storage interface</u>

FICON – IBM Goals

- Significantly improve the I/O throughput over a single channel path
- Decrease the execution time of a single channel program
- Significantly improve the data transfer rates of a single channel path
- Provide for more addressable units (devices) attached to a single channel path
- Accomplish the above in a fashion that preserves the S/390 investment in existing programming
- Provide connections that support ESCON equivalent distances with negligible performance penalties at the increased distances
- Provide a ***migration path*** to support existing controllers currently deployed in the field
- Provide the above using existing industry standards where applicable, and develop new industry standards where needed

Comparing Characteristics

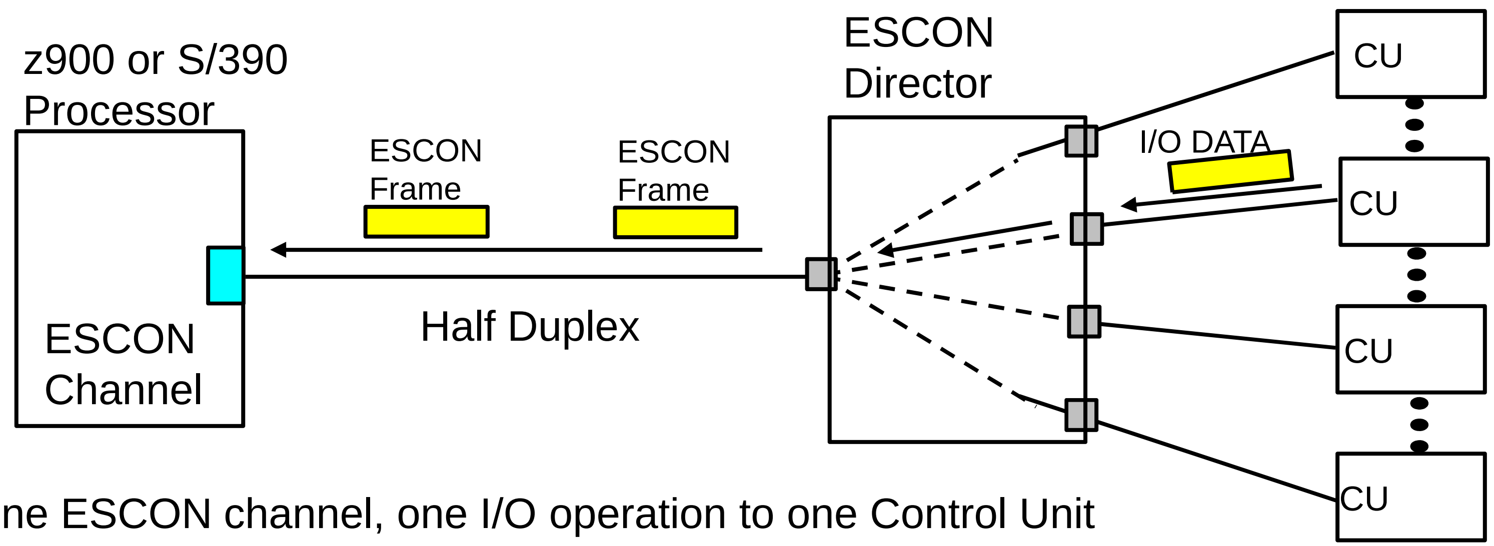
ESCON Channels

- Connection oriented
- Circuit Switching
- Read or Write
  - Half-duplex data transfers
- Dedicated path pre-established
- When packet is sent, the connection is locked
- Synchronous data transfer
- One operation at a time

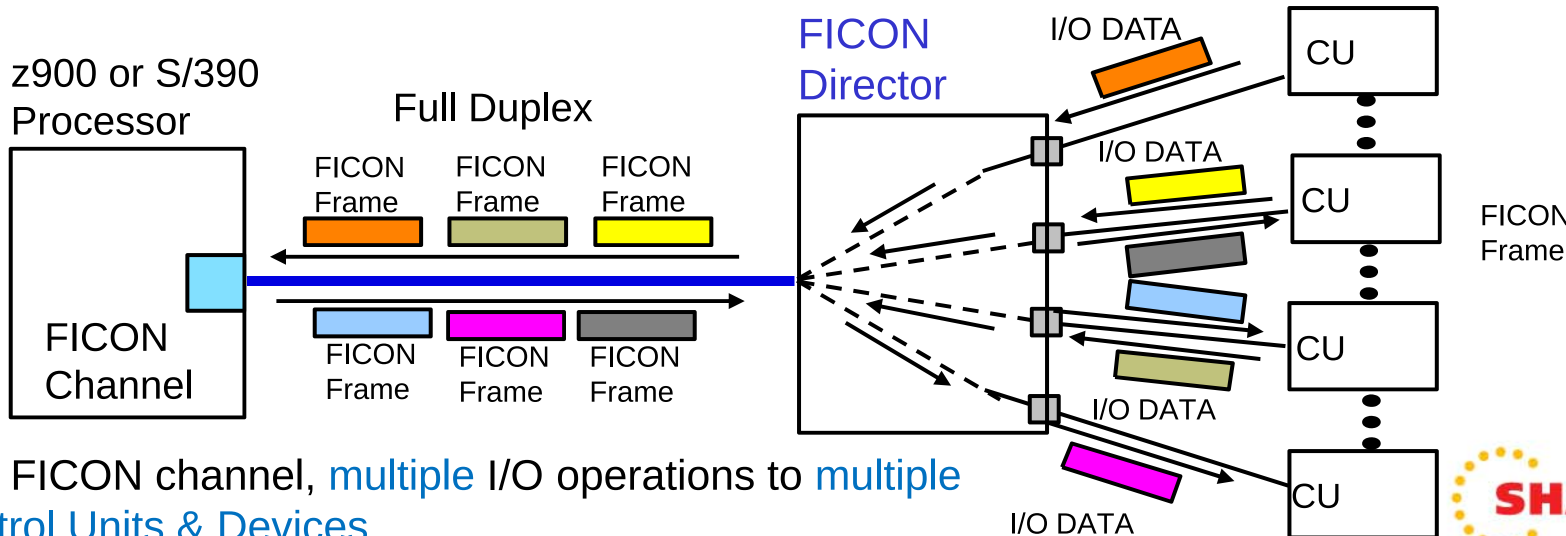
FICON Channels

- Connectionless
- Packet Switching
- Simultaneous read/write
  - Full-duplex data transfers
- Packets individually routed
- When packet is sent, connection is released
- Asynchronous data transfer
- Pipelined and multiplexed operations

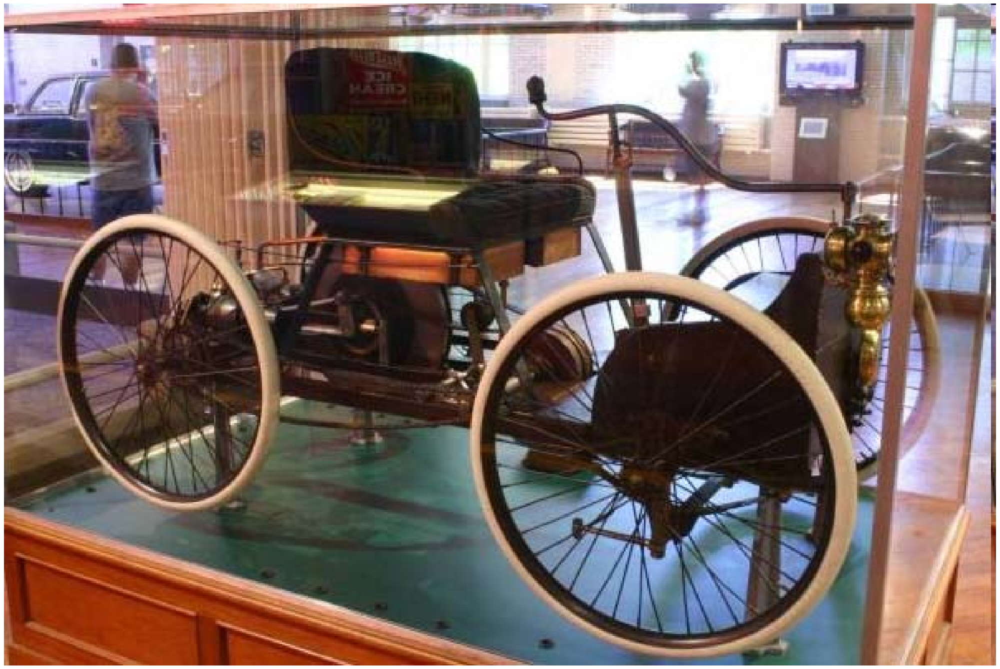
ESCON vs FICON Frame Processing



One ESCON channel, one I/O operation to one Control Unit



One FICON channel, multiple I/O operations to multiple Control Units & Devices

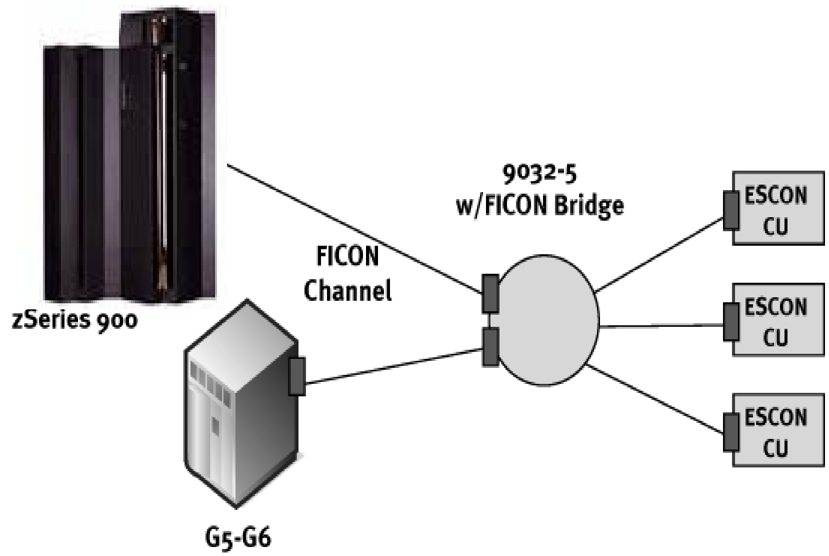


Almost a bridge too far

THE FICON BRIDGE

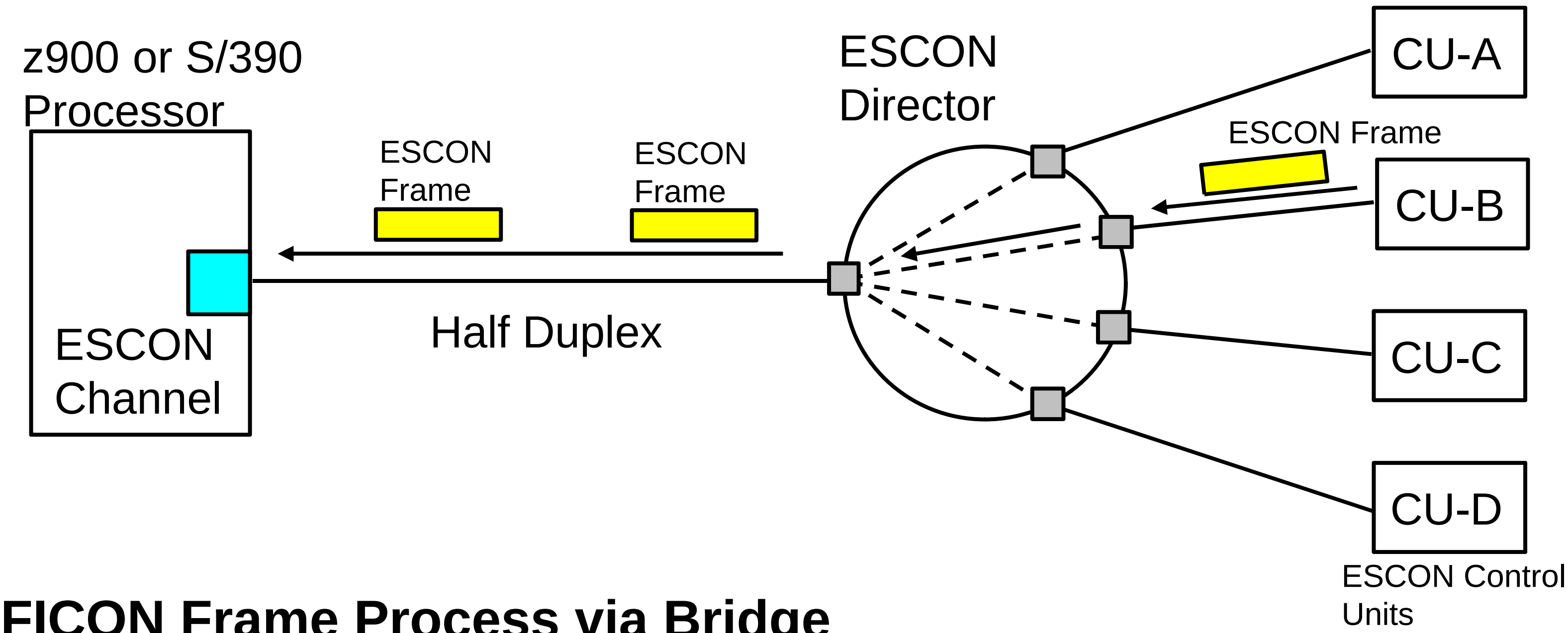
The “FICON Converter” (FCV)

- An ESCON to FICON bridge card was needed to support existing ESCON devices.
- The bridge card was a feature of the 9032-5 ESCON Directors.
- Bridged FICON gains some of the benefits of FICON:
  - 8 simultaneous transactions.
  - Increased I/O (3,200 I/O/sec).
- The bandwidth is about half that of Native FICON and is limited to 1 Gbps links.
- Bridge cards were used as a first step toward a FICON native infrastructure.

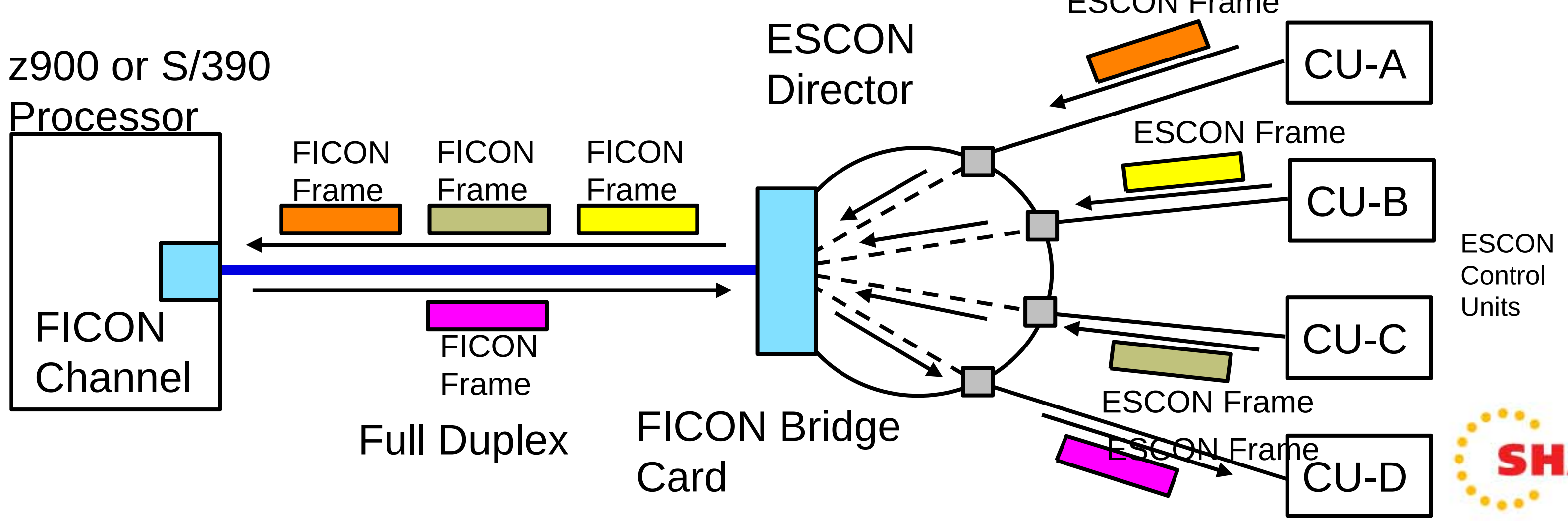


•The bridge card has 8-ESCON connections.
•The 1-external FICON port and 8-internal ESCON ports.
•Replacing an ESCON card with a FICON Bridge card swapped 8-ESCON connections for 1-FICON connection which attached to 8-ESCON devices.

ESCON vs FICON Bridge Frame Processing

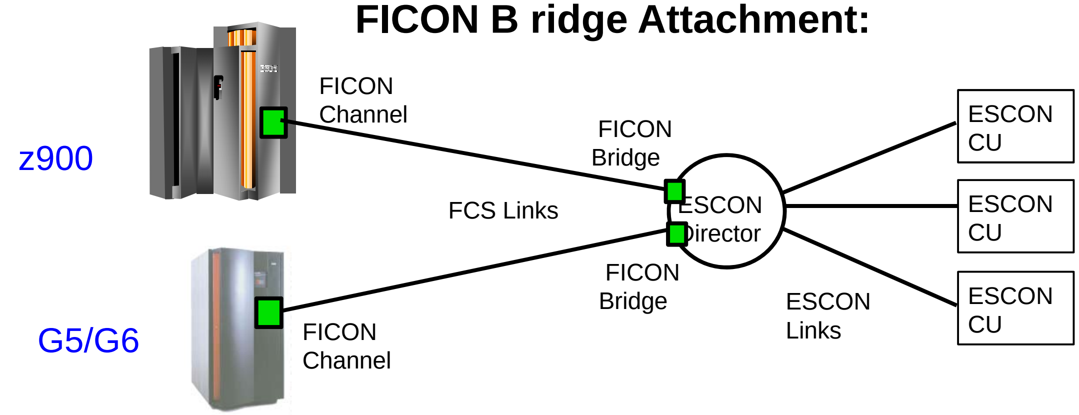


FICON Frame Process via Bridge



FICON Operating Modes

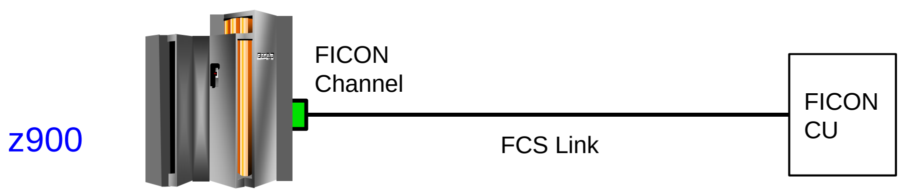
FICON B ridge Attachment:



- *Exploit FICON Channel with Existing ESCON Control Units*

Type=FCV

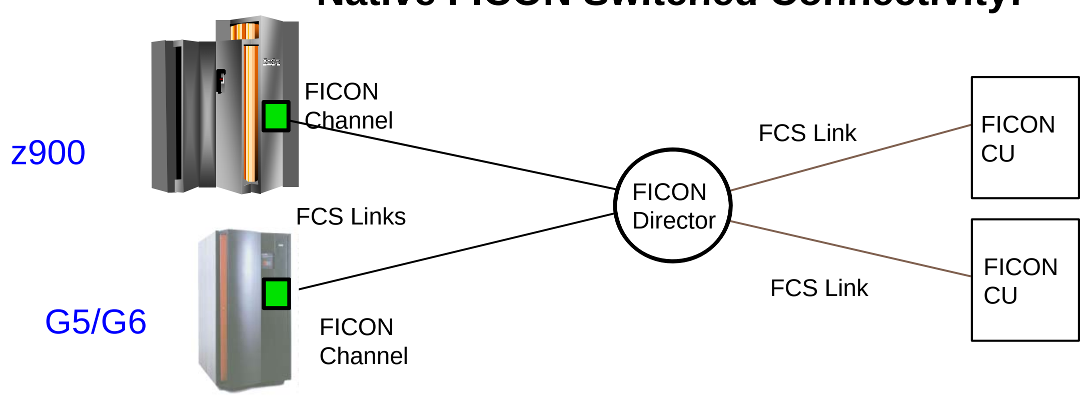
Native FICON Direct Attachment:



- *Native FICON Control Units*

Type=FC

Native FICON Switched Connectivity:



- *Full Dynamic Switching of FICON Control Units*

Type=FC

ESCON & FICON by the Numbers

A comparison

Capabilities	ESCON	FICON Bridge	FICON	FICON Express 4
I/O operations at a time	Only one	Any eight	32 open exchanges	64 open exchanges
Logically daisy-chained Control Units to a single channel	Any one I/O, take turns	Any 8 I/Os concurrently	Any number of I/Os concurrently	Any number of I/Os concurrently
Average I/Os per channel (4k blocks)	2,000-2,500	2,500	6,000	13,000
Unit addresses per channel	1,024	16,384+	16,384 +	16,384 +
Unit addresses per control unit	1,024	1,024	16,384 +	16,384 +
Bandwidth degradation	Beyond 9 km	Beyond 100 km	Beyond 100 km	Beyond 100 km

Today

Current channel technology

A Fibre Channel Standard

FICON

FICON – Security, Resiliency, Performance

- Security & Resiliency
  - Frame Delivery
    - High Integrity Fabrics
      - FC fabrics will validate the WWPN of a re-established ISL before allowing any data to flow on it
      - FICON ***requires*** high integrity fabrics
    - In Order Delivery
      - Sequence numbers, IU numbers
      - CCW numbers – enables pipelining – used to associate response with the particular command sent
  - Bit Stream Integrity
    - FC2 CRC created by adapter ensures that data as it flows on the link is not corrupted
    - FICON adds separate CRC to insure integrity of data at FC4 layer (end-to-end between hosts)
  - Carried over from ESCON:
    - Link Incident Reporting (LIRR), State Change Notification, RNID

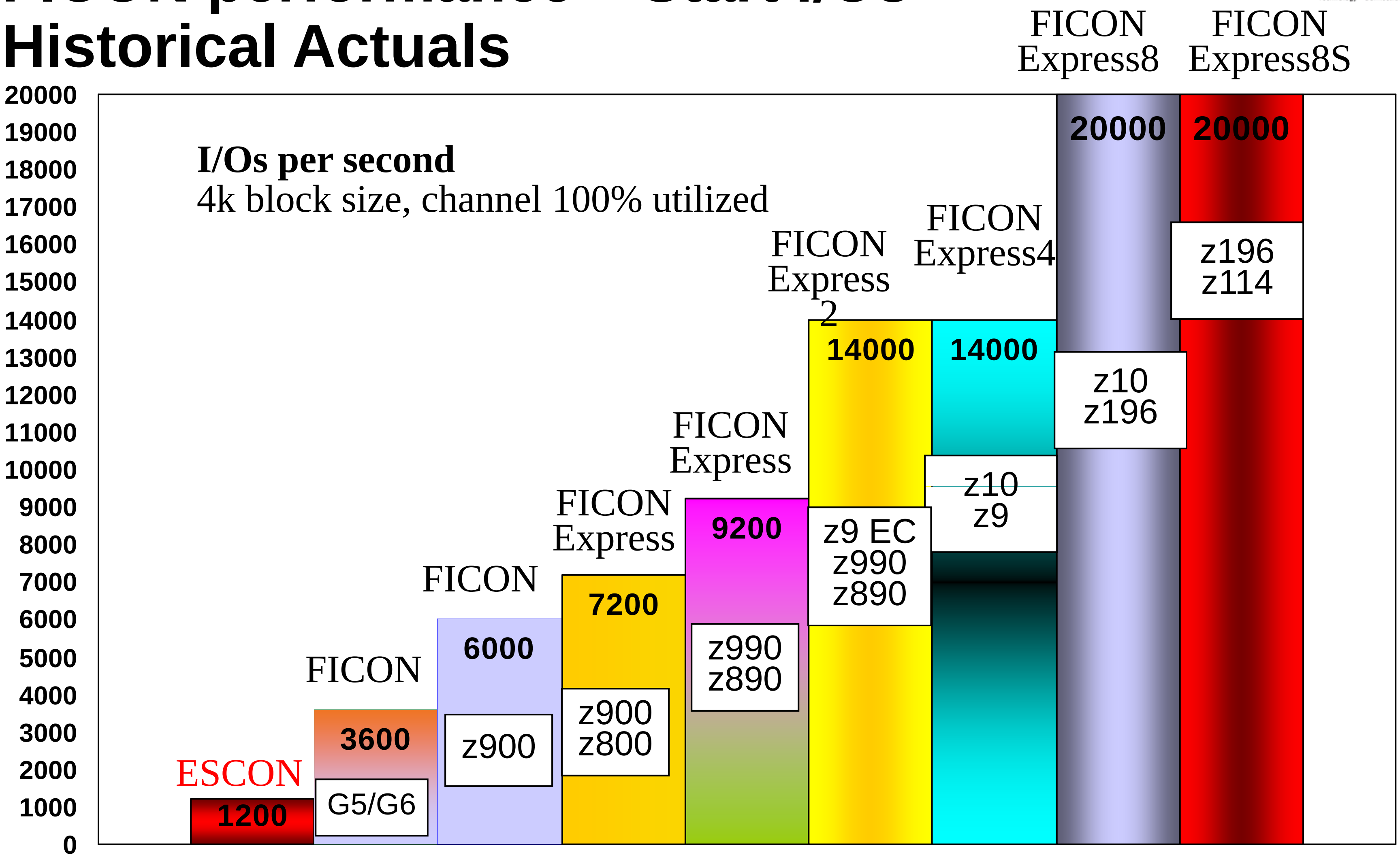
FICON – Security, Resiliency, Performance

- Performance Management and Droop
  - Fibre Channel standard provides for link flow control through buffer-to-buffer credit scheme
    - FC2 function that controls stream of frames between end points on a link (i.e. between nearest neighbors)
    - Determines the distance two nodes can be apart and still maintain full link frame rate
  - IU Pacing - FICON architectural provision for end-to-end flow control
    - Prevents flooding of target N-Port
      - *With command pipelining needed mechanism to prevent over-running the control unit*
    - Avoids droop at distance

FICON – Security, Resiliency, Performance

- Performance Management and Droop
  - I/O Priority
    - Separate priority mechanisms in I/O Subsystem, Channel and Control Unit
  - Modified Indirect Data Addressing Words (MIDAWs)
    - A method of gathering and scattering data into & from non-contiguous System z storage locations during an I/O operation
    - Removed IDAW restriction that appended data must be on 2K storage boundary
    - Improved performance of certain applications (e.g. DB2 sequential workloads) that process small records with Extended Format data sets
  - Measurements – granular to service class
    - Allows algorithms for WLM based I/O priority, DCM & intelligent data placement
  - Dynamic CHPID Management
    - allows adding/removing bandwidth to a control unit as workload needs dictate

FICON performance – Start I/Os Historical Actuals



Rev’ing the Engine

ZHPF

High Performance FICON

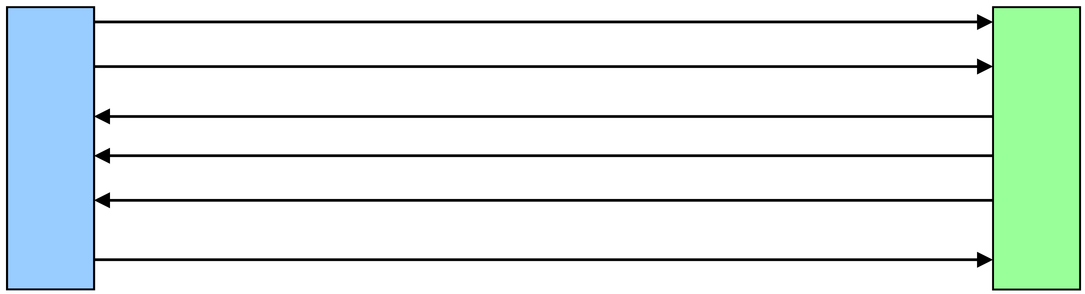
The host communicates directly with the control unit

- The channel is acting as a conduit
- No individual commands or state tracking
- The CCW program is sent to the control unit in one descriptor
- Uses the Fibre Channel FCP link protocol. The channel provides both the new and old protocols
- HPF provides increased performance & bandwidth for small block transfers
- Complex channel programs that are not easily converted to the new protocol still execute with the existing FICON protocol
- Devices accessible using both old and new protocols

Link Protocols for 4K Read

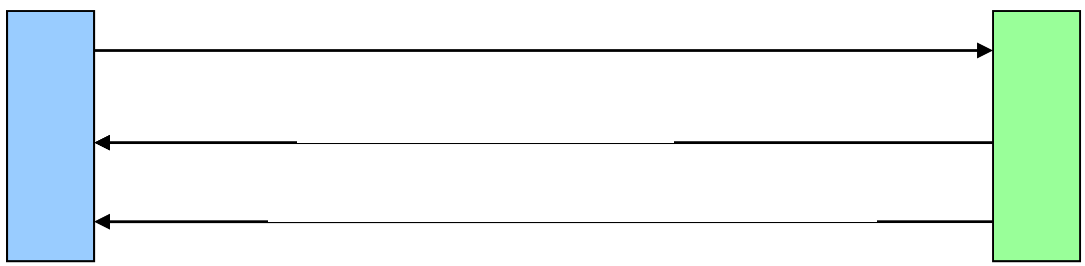
FICON:

Two Exchanges opened and closed
Six Sequences opened and closed



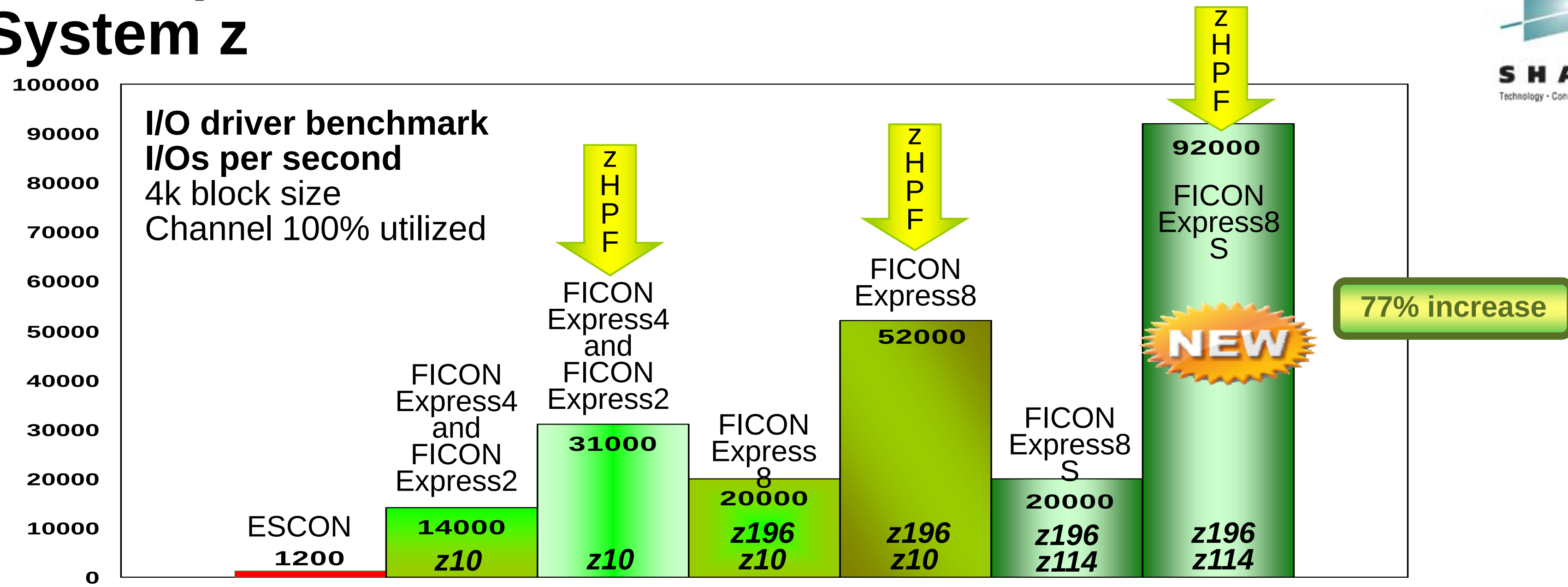
zHPF:

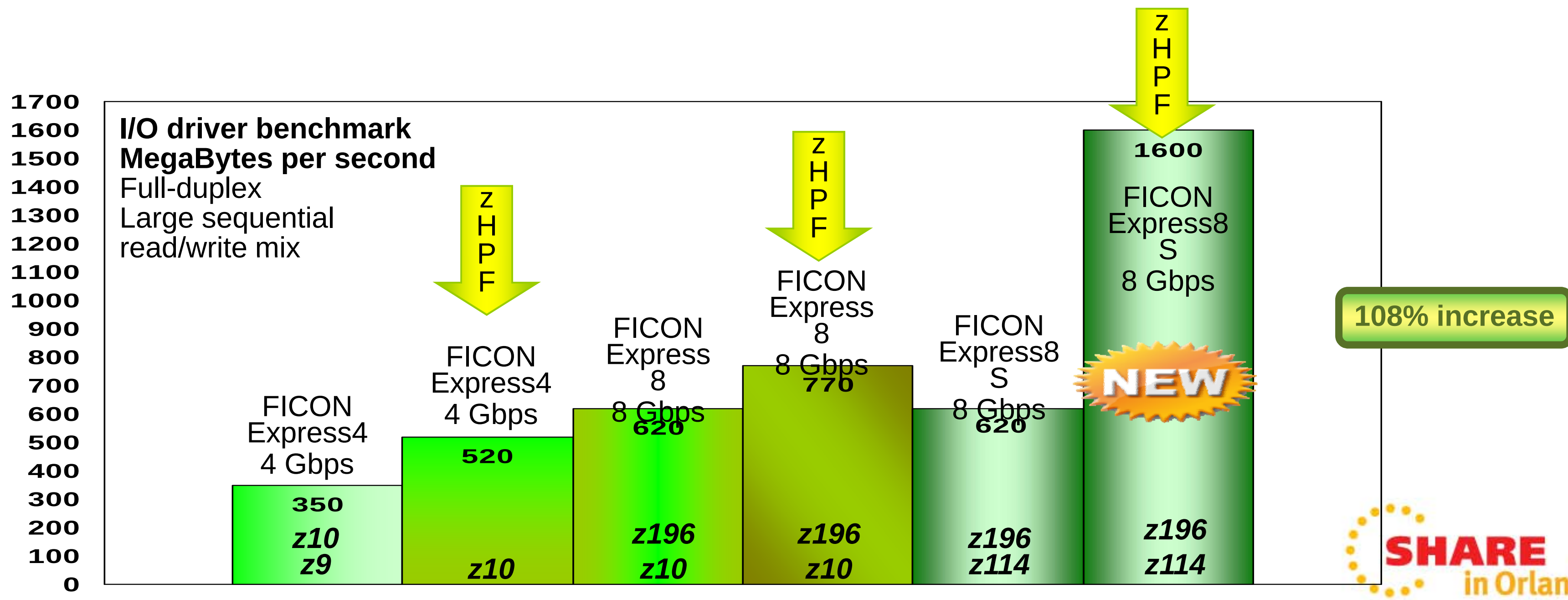
One exchange opened and closed
Three Sequences opened and closed



CHANNEL CONTROL UNIT

FICON performance on System z





Looking Ahead

But no reading of the tea leaves

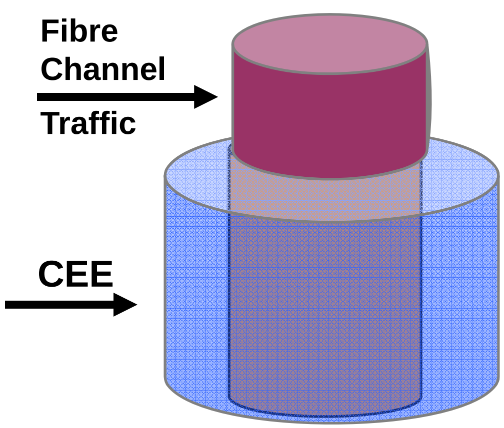
How fast can this go? And over what roads?

What's Rumbling About in the Industry?

- 16Gb Fibre Channel
  - Physical interface approved as an ANSI INCITS T11.2 standard in Sept. 2010
  - Auto-negotiated backward compatibility to 4Gb
  - Aimed at improved price-performance (as market matures)
  - Twice the bandwidth of 8Gb
  - Leapfrogs fiber channel storage SAN fabric over 10GbE
- 32Gb Fibre Channel
  - Next logical step for Fibre Channel after 16Gb
  - Work on-going in ANSI with proposals put forward
  - Future will depend on market demand
- Fibre Channel over Ethernet (FCoE)
  - See next charts for description
  - Future will depend on market demand

What is FCoE?

FCoE is a technology that straddles two worlds: Fibre Channel and Ethernet

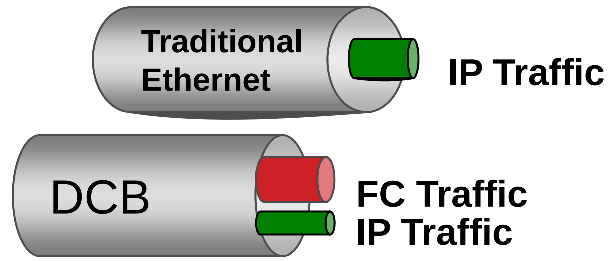


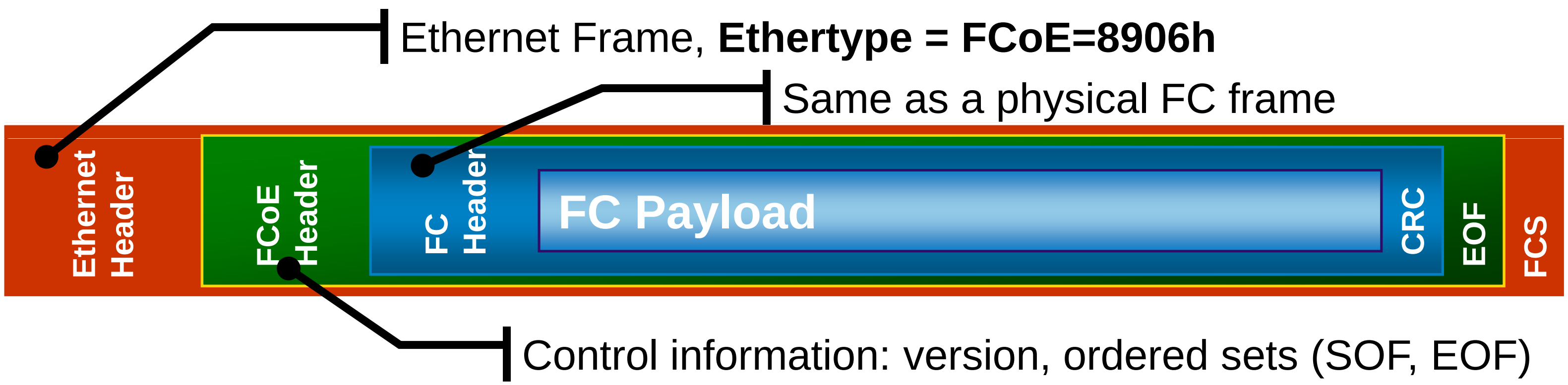
- **FC View**: FC gets a new transport in the form of lossless Ethernet (CEE)
- **Ethernet view**: A new upper-layer protocol, or storage application, that runs over a new lossless Ethernet

What does FCoE look like?

It's an encapsulation protocol

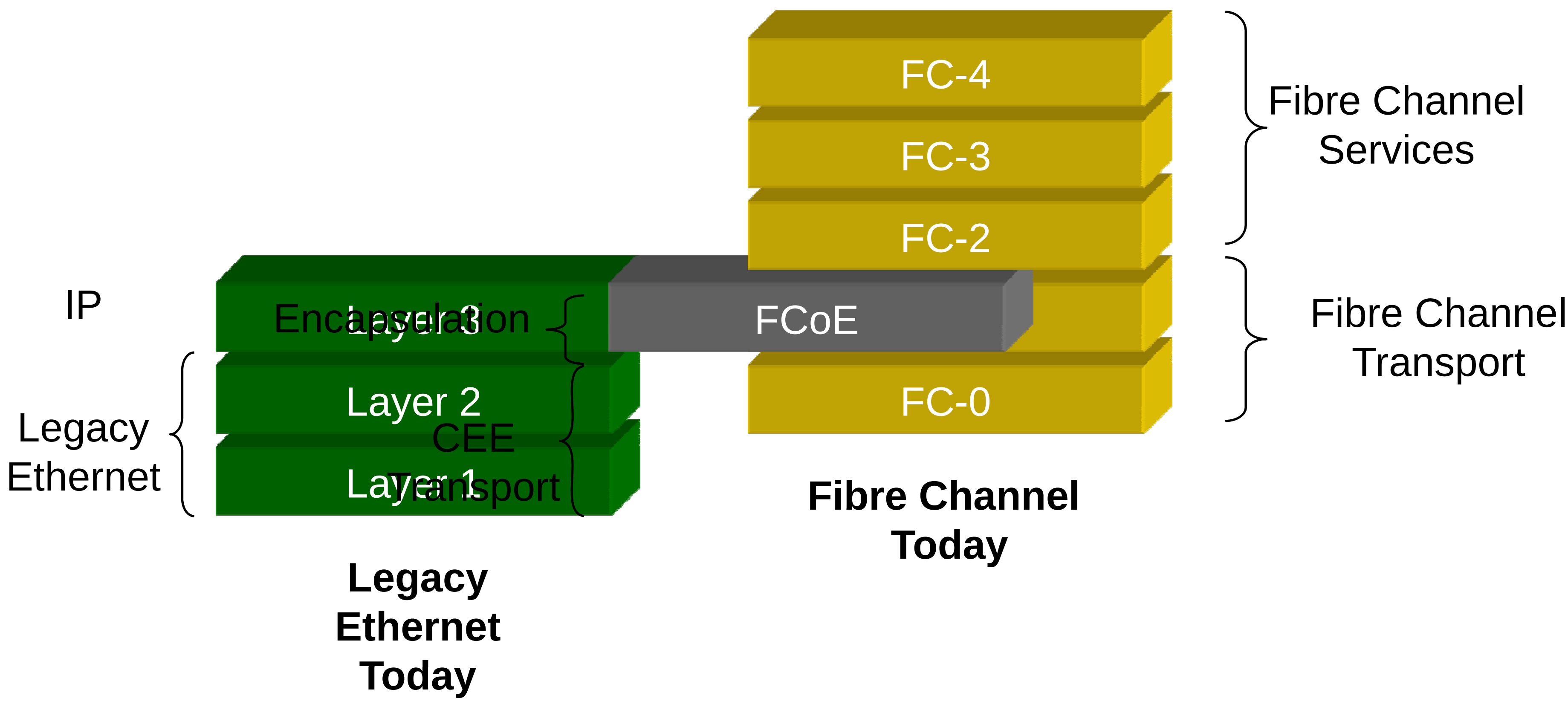
Encapsulation protocol for transporting FC over Ethernet (Lossless Ethernet: DCB)





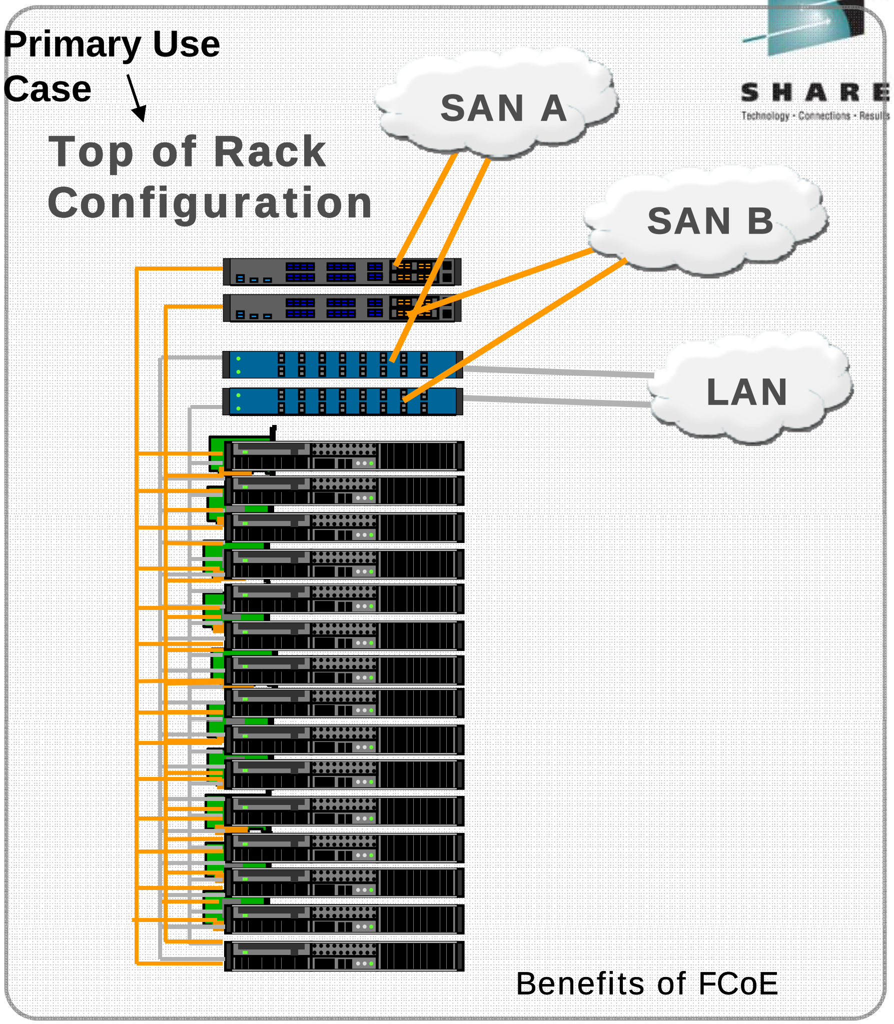
- FC frame remains intact: FC does not change
- Ethernet needs a larger frame: Larger than 1.5 KB
- Ethernet must become lossless to carry storage data with integrity

Where does FCoE fit in the stack?



FCoE & CEE Benefits

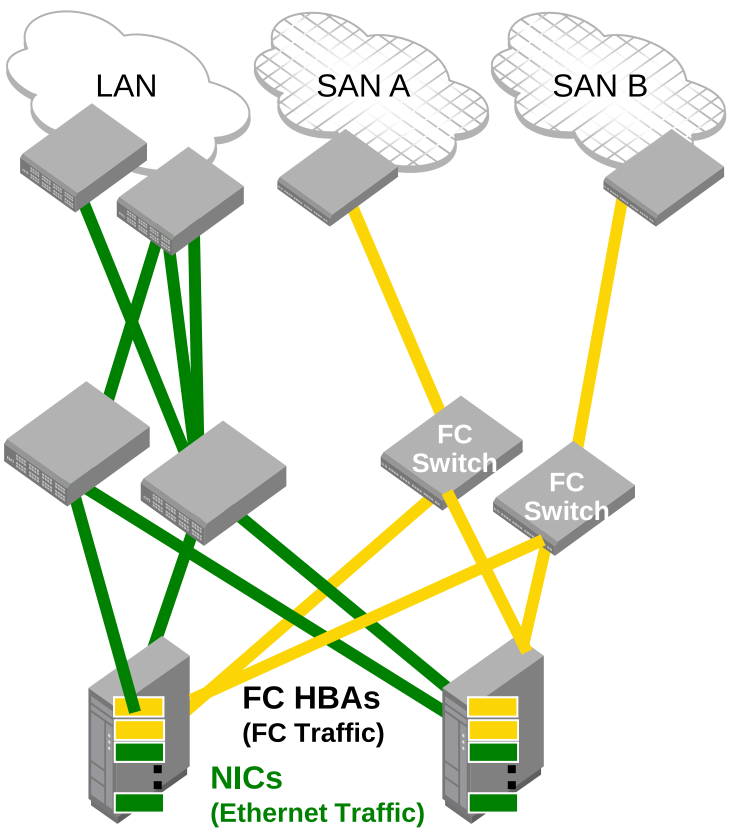
- Reduce number of server ports
- Reduce number of switches ports
- Reduce cabling
- Reduce power consumption
- Increase speed of links
- Increase utilization of links



Consolidating I/O Interfaces Lowers CapEx and OpEx

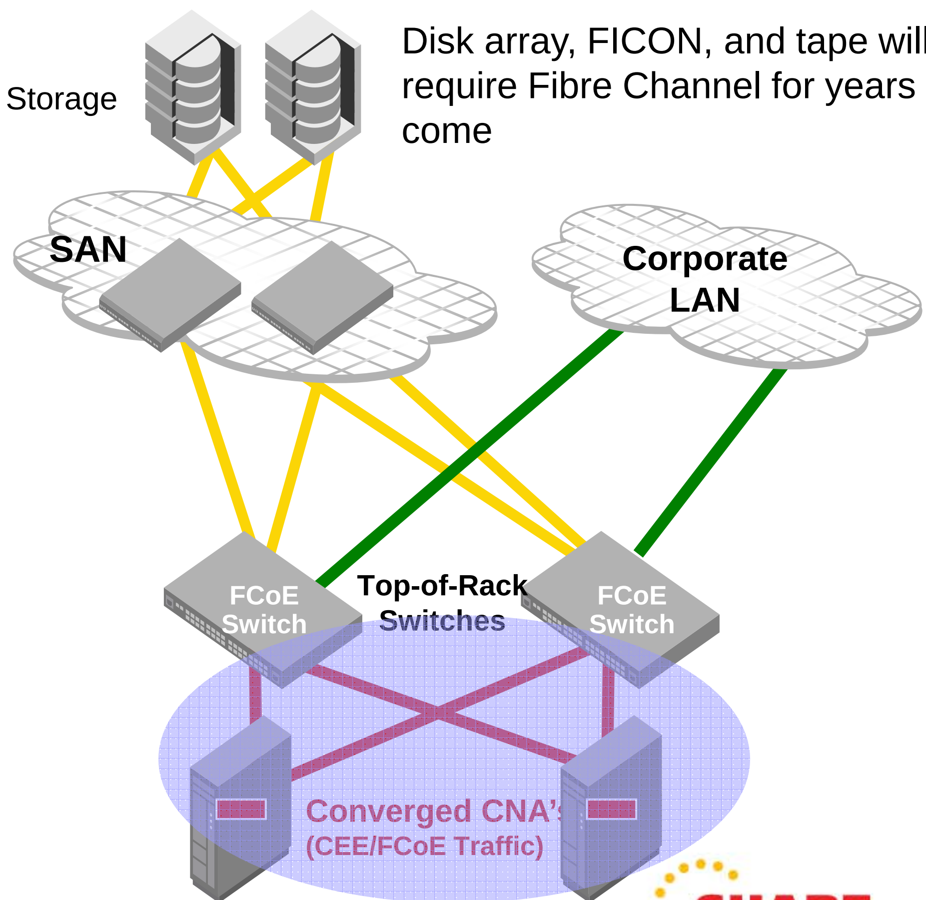
Primary FCoE Use Case

Before Unified I/O



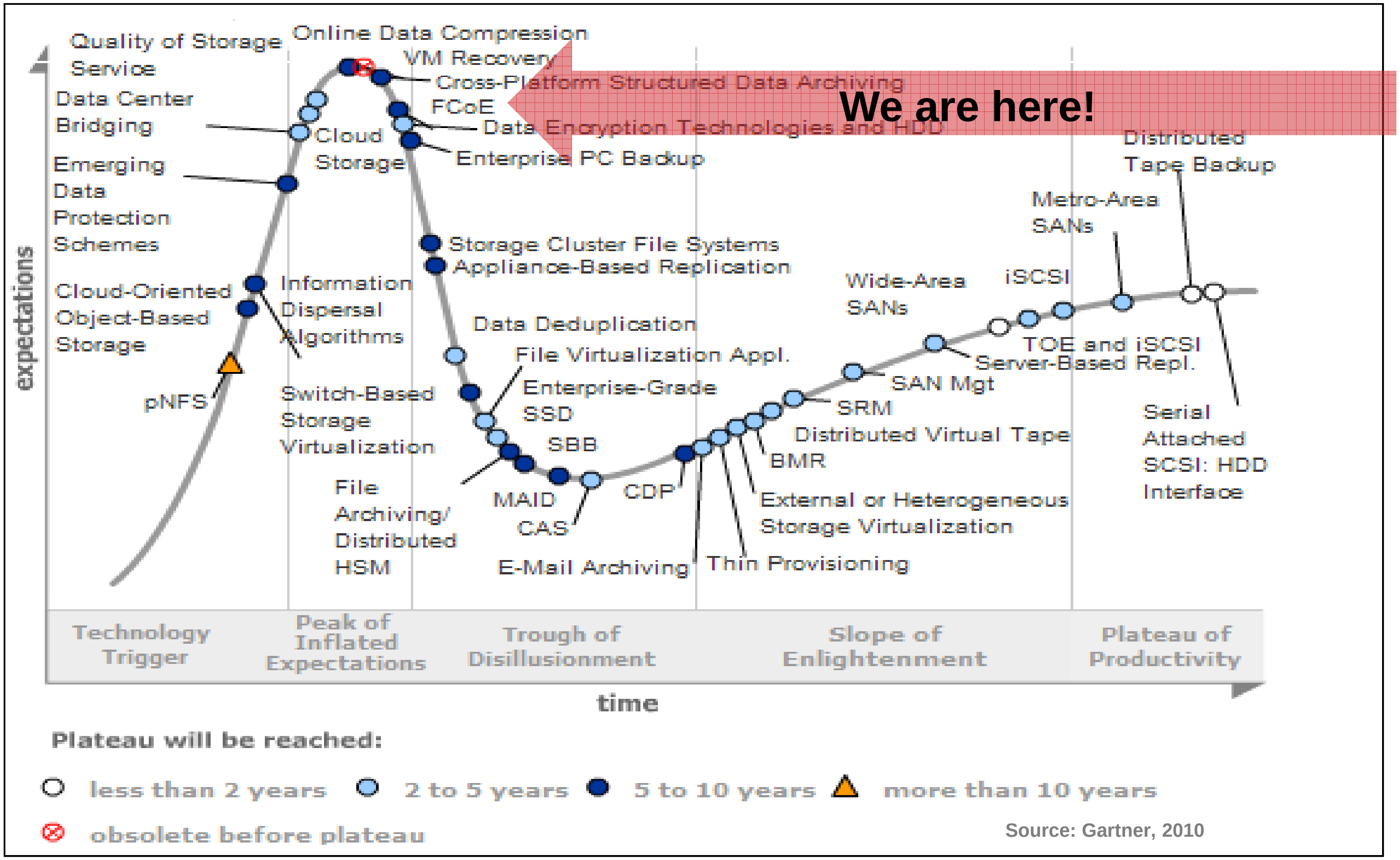
After Unified I/O

Disk array, FICON, and tape will require Fibre Channel for years to come



FC Ethernet FCoE and CEE

Storage Technology Hype Cycle Curve



SHARE in Orlando 2011

As the Evolution Continues….

- System z continues focus on key Enterprise storage fabric attributes:
  - Security
  - Resiliency
  - Performance

  through:

    - Continued innovation in channel and control unit architectures
    - Working within ANSI Standards Committees to drive requirements into the standards of evolving technologies (such as FCoE)
    - Working with eco-system vendor partners to provide differentiating functions

SHARE, Orlando, August 2011

Evolution of the System z Channel

Session 9934

THANK YOU!

Fun Links

- http://www.vikingwaters.com/htmlpages/MFHistory.htm
- http://en.wikipedia.org/wiki/Mainframe_computer
- http://en.wikipedia.org/wiki/Channel_program#Channel_Program
- http://en.wikipedia.org/wiki/IBM_System/360#Channels
- http://en.wikipedia.org/wiki/Channel_I/O
- http://en.wikipedia.org/wiki/ESCON
- http://en.wikipedia.org/wiki/FICON

REFERENCES

Speaker Biography

- Patty Driever
  - IBM
  - System z I/O and Networking Technologist
- Contact Information
  - pgd@us.ibm.com

Speaker Biography

- Howard L. Johnson
  - BROCADE
  - Technology Architect, FICON
    - 27 years technical development and management
- Contact Information
  - howard.johnson@brocade.com

THIS SLIDE INTENTIONALLY LEFT BLANK