

PCI



SIGTM

10TH YEAR ANNIVERSARY

PCI



SIG™

10TH YEAR ANNIVERSARY

**PCI Express™
Protocol Overview
Part 1**

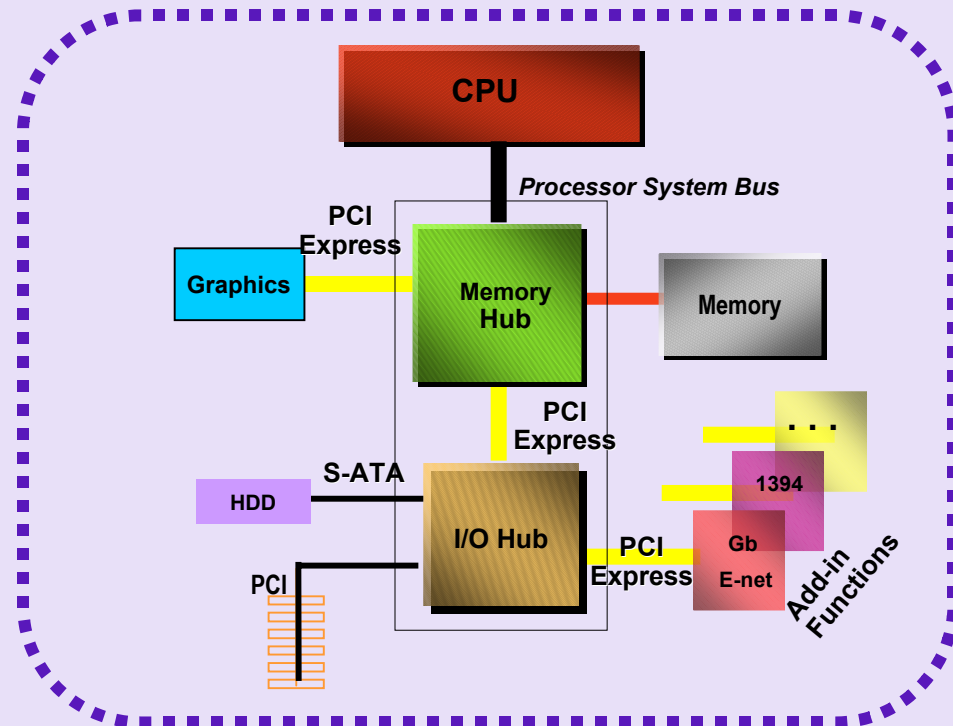
David Harriman

Agenda

- **PCI Express Features Summary**
- **Transactions & Packets**
- **Ordering**
- **Flow Control**
- **Messages**
- **Data Integrity**
- **<Break>**
- **Error Classification, Signaling and Reporting**
- **Hot Plug**
- **Power Management**
- **Transaction Classes & Virtual Channels**
- **Isochronous Support**
- **Summary**
- **Call to Action**

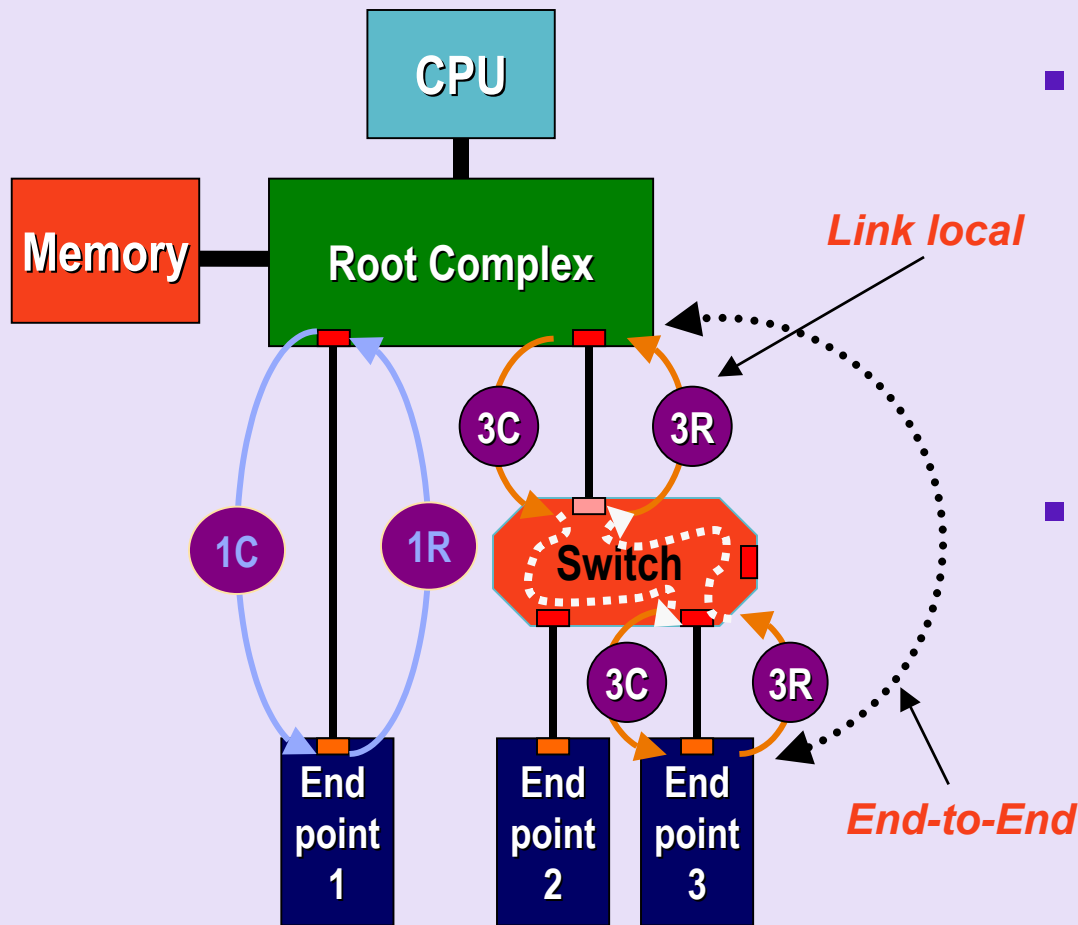
PCI Express™ Features Summary

- **Physical Interface:**
 - ✓ Point-to-point full-duplex interconnect
 - ✓ Differential low voltage signaling
 - ✓ Embedded clocking
 - ✓ Supports connectors and cables
- **Performance:**
 - ✓ Scalable frequency (2.5 Gb/sec initially)
 - ✓ Scalable width (1,2,4,8,12,16,32)
 - ✓ Low latency and high utilization
- **PCI Compatibility:**
 - ✓ Configuration model and PCI Software Driver model
 - ✓ PCI PM software compatible



- **Protocol:**
 - ✓ Fully packetized split-transaction
 - ✓ Credit-based flow Control
 - ✓ Hierarchical topology support
 - ✓ Virtual Channel mechanism
- **Advanced Capabilities:**
 - ✓ Enhanced Configuration and Power Management
 - ✓ RAS: CRC-based Data Integrity, Hot Plug, Advanced error logging/reporting
 - ✓ Advanced Switching Extensions

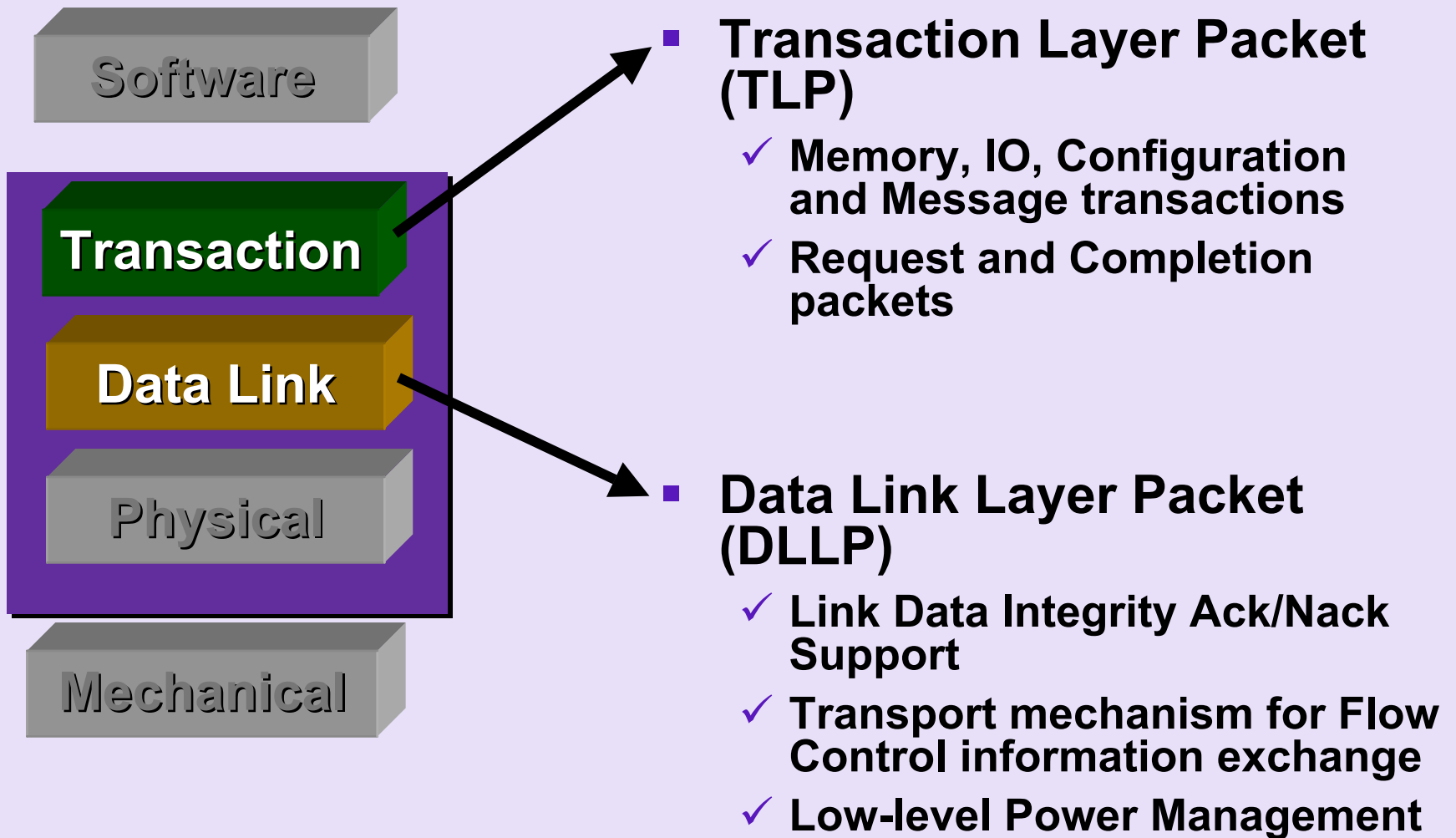
Transaction Basics



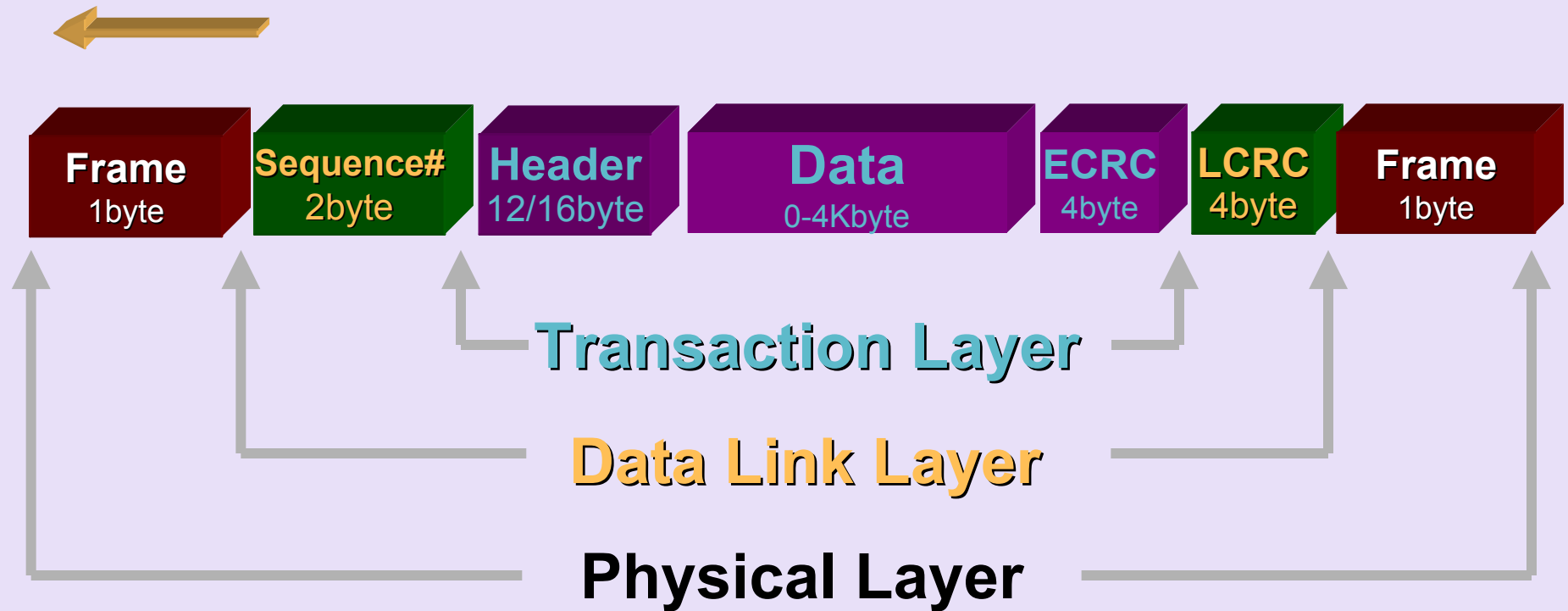
- Full split-transaction packet protocol
 - ✓ Request Packet (e.g. 1R)
 - ✓ Completion Packet (e.g. 1C)
- Transactions flow between two ends
 - ✓ Switches are transient elements
 - ✓ Subject to Ordering, Flow Control and Data Integrity mechanisms

Link (local) = Between two components
End to End = Between Requester/Completer

Packet Sources and Types

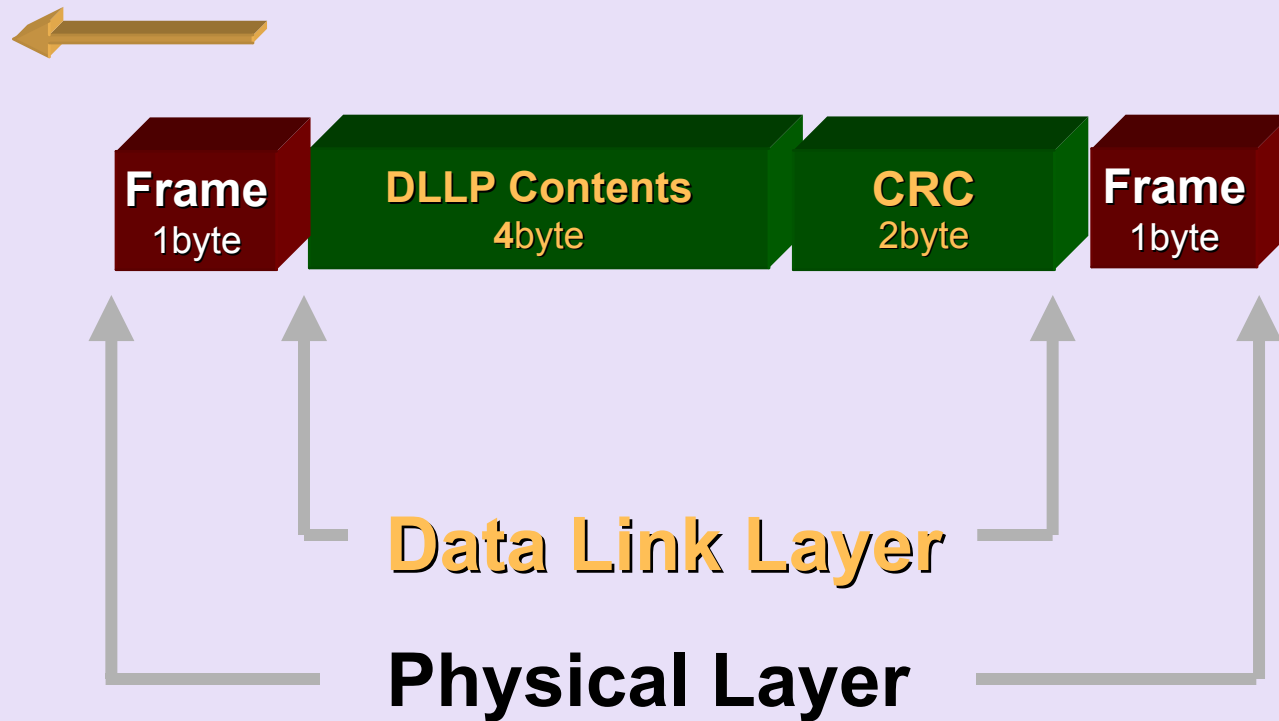


Packet Formation - TLP



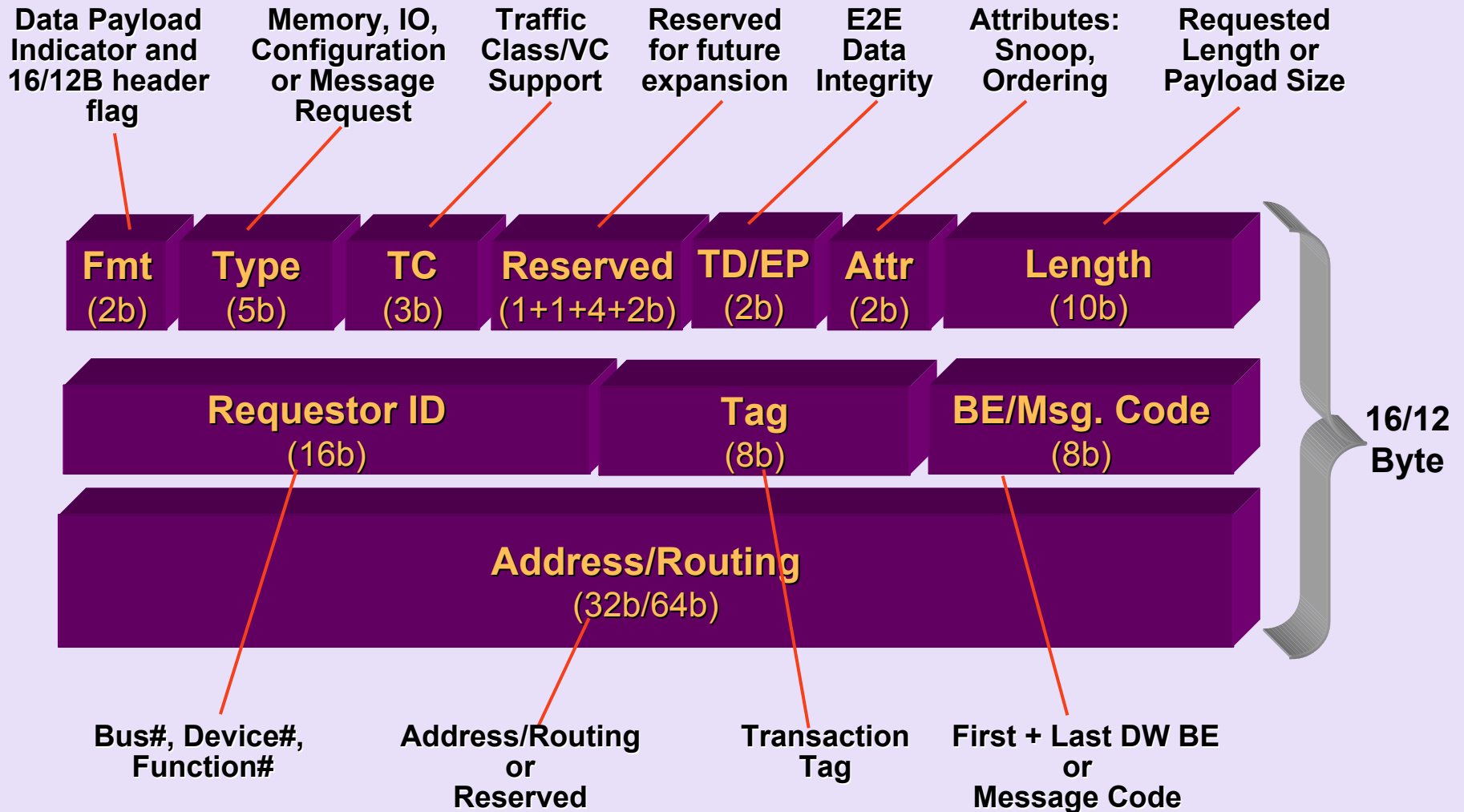
Core of TLP not modified by Data Link or Physical Layers

Packet Formation - DLLP

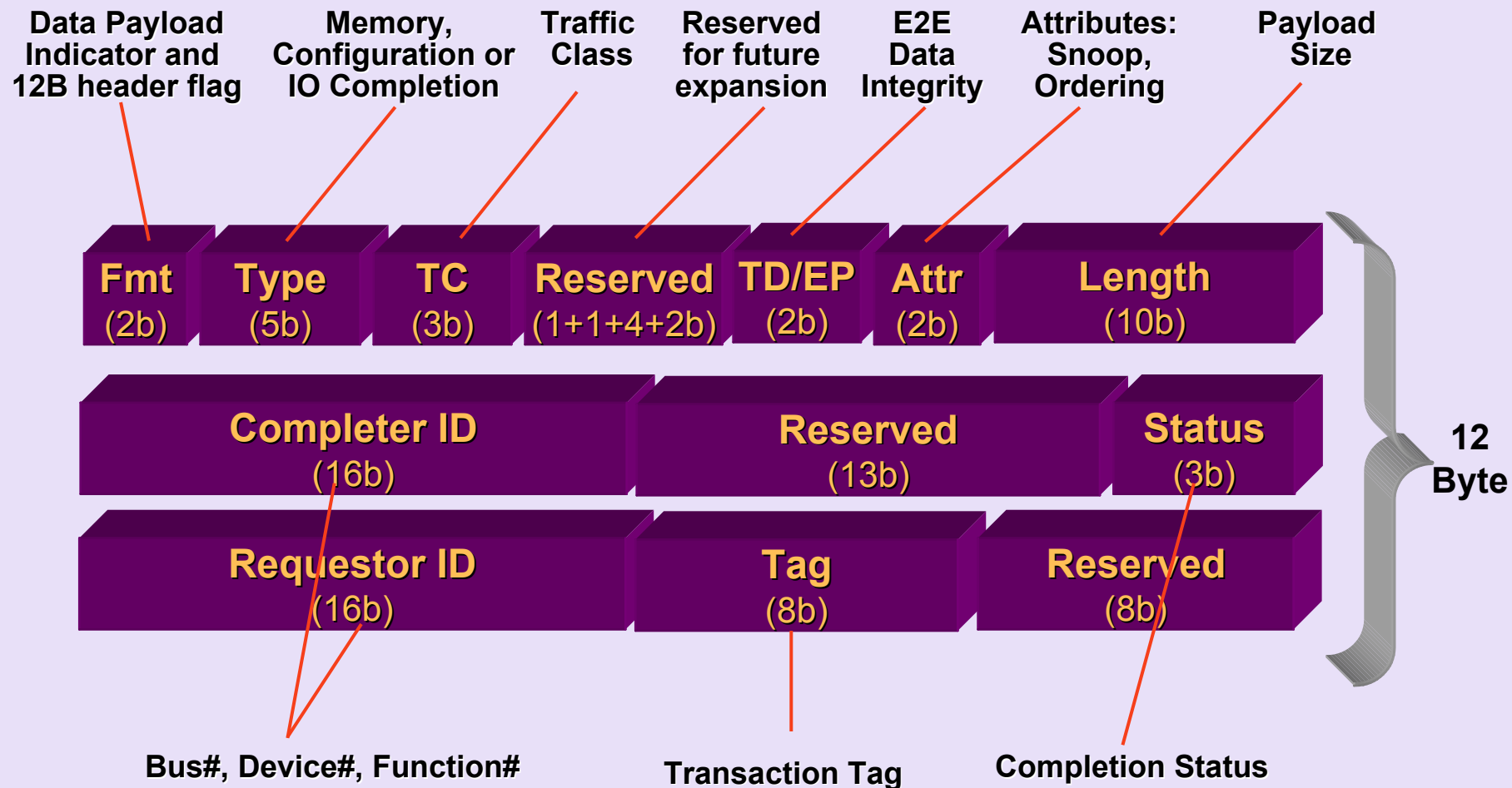


Core of DLLP not modified by Physical Layer

TLP Request Headers - Detail

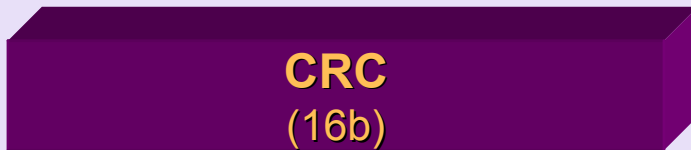


TLP Completion Headers - Detail



DLLP - Detail

DLLP Type:
Ack/Nak,
InitFC/UpdateFC
PM DLLP

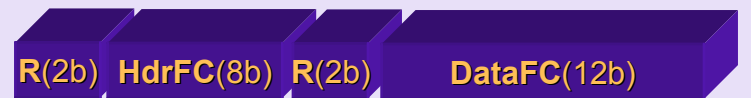


DLLP Data Integrity Protection

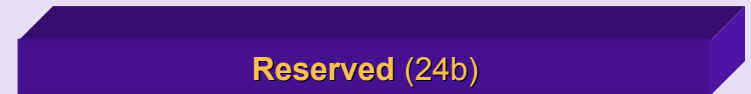
Ack/Nak:



InitFC/
UpdateFC:

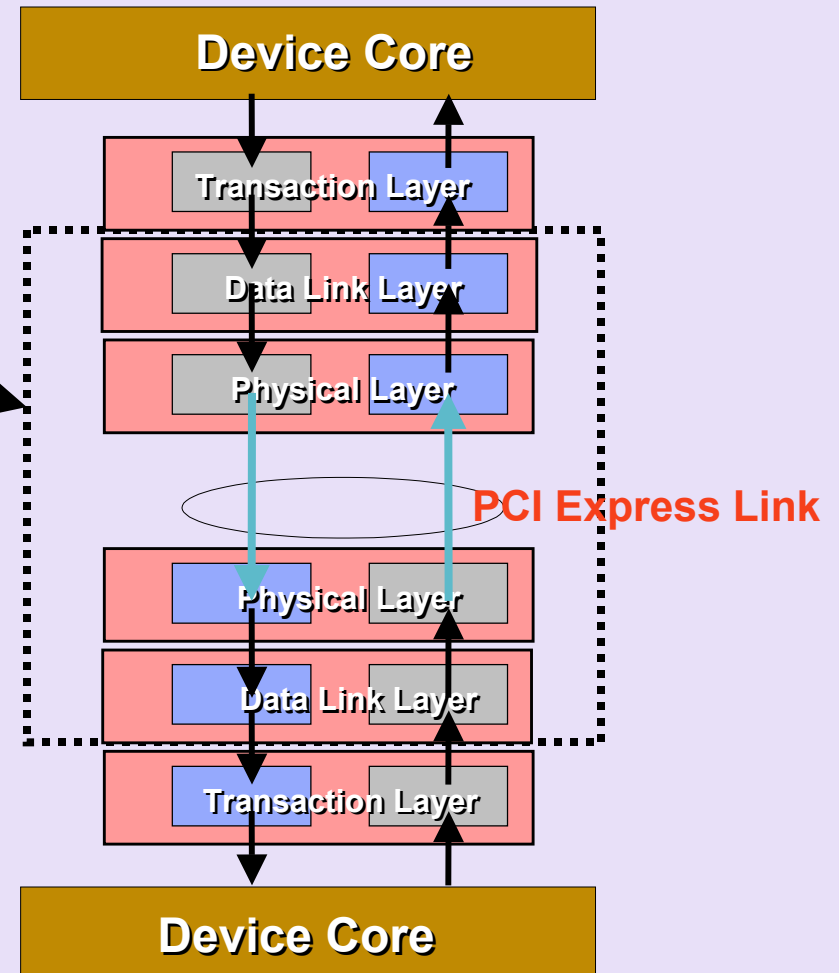


PM DLLP:



Transaction Ordering

- Ordering rules cover transactions end-to-end
- Data Link to Data Link Layer
 - ✓ Transactions serialized i.e. no reordering
- Transaction Layer ordering
 - ✓ Baseline = PCI-X Ordering
 - ✓ Ordering Relaxations via Transaction Attributes
 - ✓ Ordering Independent Virtual Channels

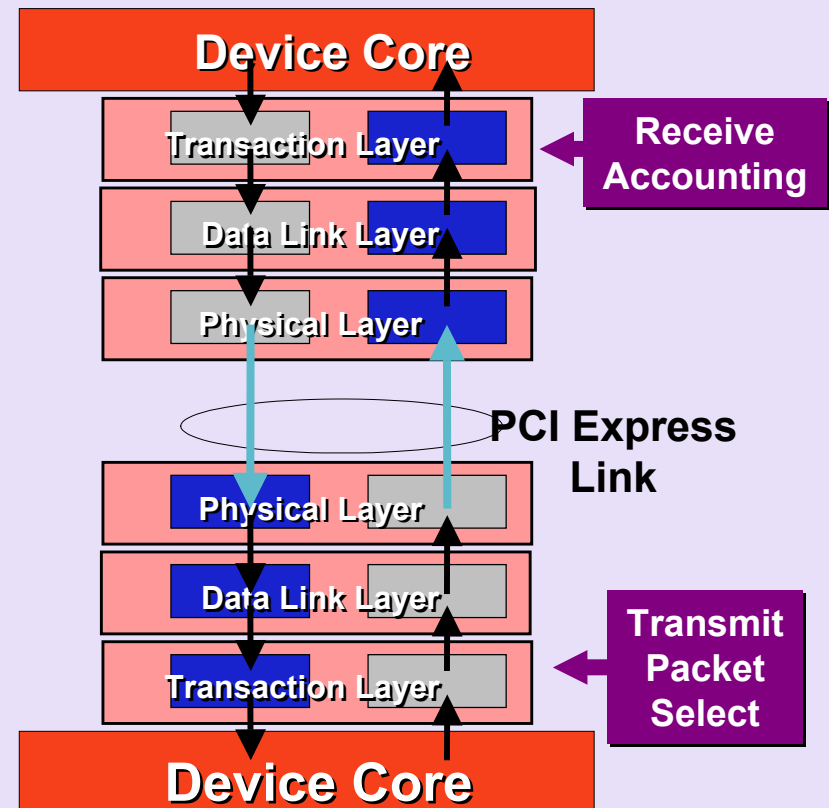


Ordering Rules

Row Pass Column?		Posted Request	Non-Posted Request		Completion	
		Memory Write or Message Request (Col 2)	Read Request (Col 3)	I/O or Configuration Write Request (Col 4)	Read Completion (Col 5)	I/O or Configuration Write Completion (Col 6)
Posted Request	Memory Write or Message Request (Row A)	a) No b) Y/N	Yes	Yes	a) Y/N b) Yes	a) Y/N b) Yes
Non-Posted Request	Read Request (Row B)	No	Y/N	Y/N	Y/N	Y/N
	I/O or Configuration Write Request (Row C)	No	Y/N	Y/N	Y/N	Y/N
Completion	Read Completion (Row D)	a) No b) Y/N	Yes	Yes	a) Y/N b) No	Y/N
	I/O or Configuration Write Completion (Row E)	Y/N	Yes	Yes	Y/N	Y/N

Flow Control

- Transaction to Transaction Layer across one Link
- Prevents overflow of receiver buffers
- Enables compliance with ordering rules
- Credit-based scheme
 - ✓ Transmitter throttles according to its supply of credits
 - ✓ Requesters can not use FC to throttle completions



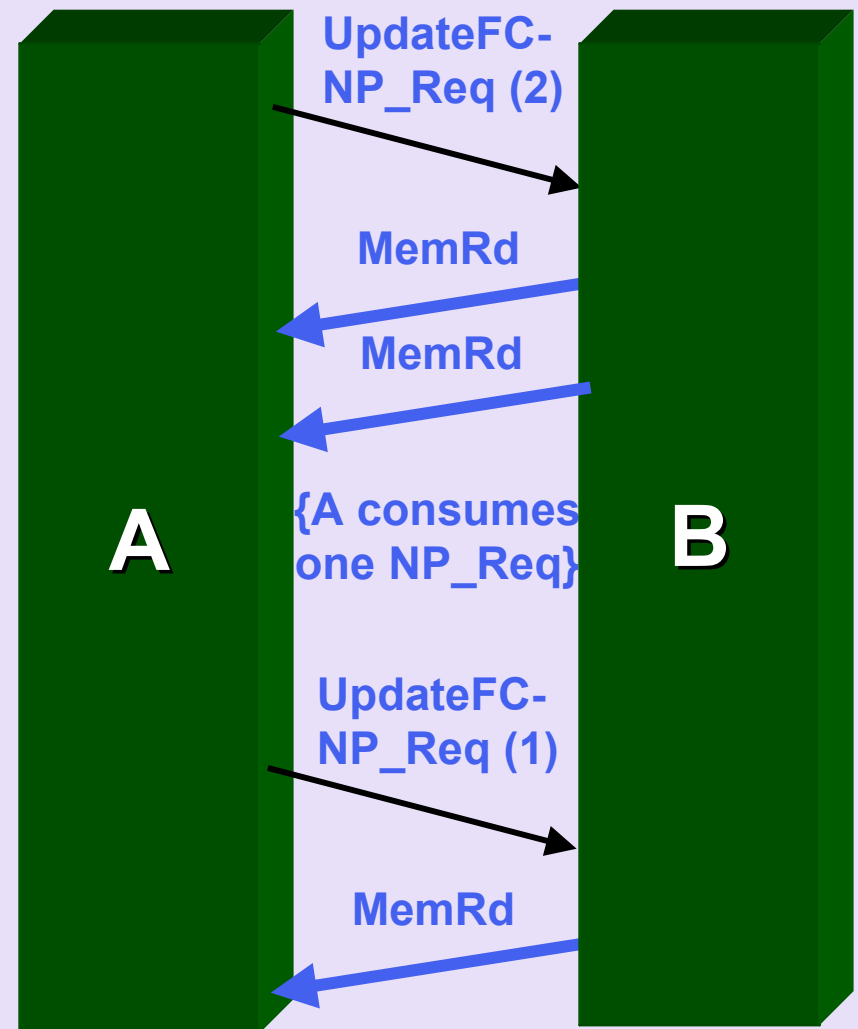
Flow control and ordering rules enable deadlock-free operations

Flow Control (cont.)

- **Handled by the Transaction Layer in cooperation with the Data Link Layer**
 - ✓ DLLPs used to exchange FC information
- **FC information covers separately:**
 - ✓ Posted and Non-Posted Request Queues, Completion Queues
 - ✓ Separate Header vs. Data Queues
- **FC is orthogonal to the data integrity mechanisms**
- **FC does not imply anything about the completion status of a Request**

Flow Control Example

1. A advertises buffer space for two Non-Posted Requests
2. B sends two Memory Read Requests
3. A consumes one of the Non-Posted Requests
4. A advertises the released buffer space to B
5. B sends another Memory Read Request



Message Transaction Support

- **Formal definition of Message transactions**
- **Initial applications for In-Band “Virtual Wire” Signaling**
 - ✓ **PCI Express replaces miscellaneous “side-band” signals with in-band messages**
 - ✓ **Error Signaling, Interrupt Signaling, Power Management, Hot-Plug Signaling, Lock Support**
- **Vendor-specific Messages**
- **Advanced Switching (AS) Extensions**
 - ✓ **AS Specification provides definition of Messages for Packet Switching applications**

RASUM Capabilities

- **RASUM = Reliability, Availability, Serviceability, Usability and Manageability**
- **Server, Communication and Client platforms**
- **PCI Express Portfolio**
 - ✓ **Data Integrity**
 - ✓ **Error Signaling and Logging**
 - ✓ **Hot-Plug/Swap and Surprise Removal**
 - ✓ **Fabric Resource Management**

**Standard Scaleable RASUM Capabilities from
Server to Client**

Data Integrity Support

- **Requirements for Robust Data Integrity**
- **Data Link Layer Mechanisms (Link/local):**
 - ✓ TLPs protected using 32bit CRC
 - ✓ DLLPs protected using 16bit CRC
 - ✓ TLP error recovery through Data Link-level retry
 - ✓ Supplemental coverage through 8b/10b
 - ✓ Loss of packets detected using Sequence Numbers
- **Transaction Layer Mechanisms (End-to-End):**
 - ✓ Optional coverage using 32bit CRC
 - ✓ Data Poisoning capability

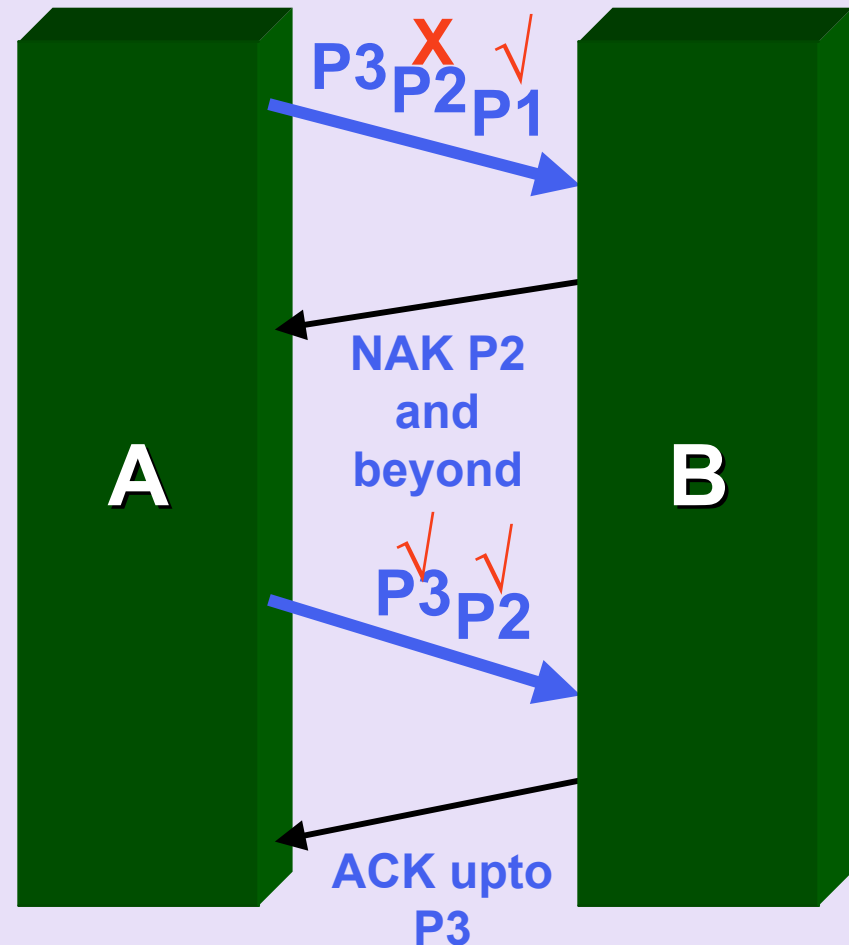
**Robust Data Integrity Allows for Signaling
Frequency Headroom**

Link Data Integrity for TLPs

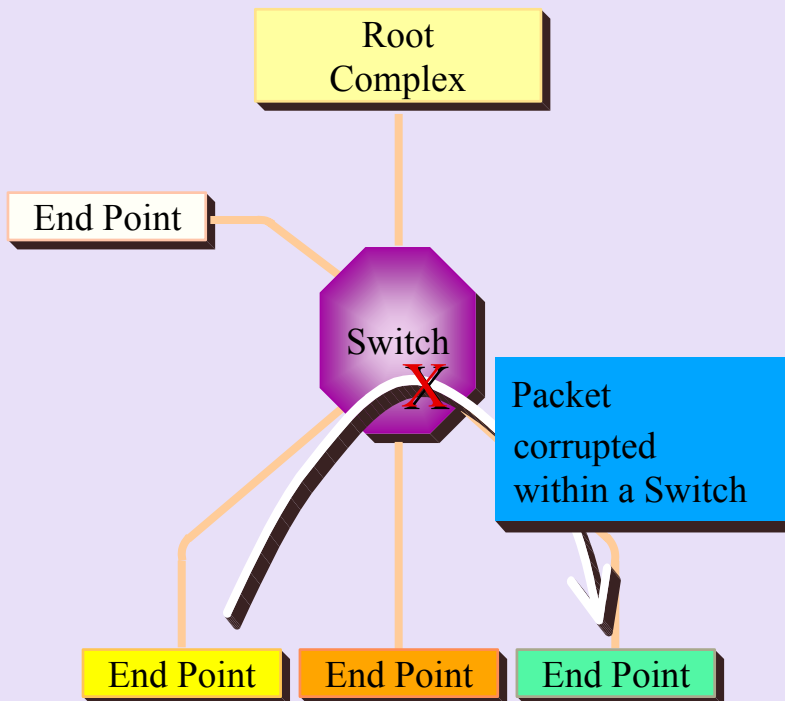
- **Covers integrity of Link between two directly attached PCI Express devices across one Link**
- **Transmit side :**
 - ✓ **Applies 32bit CRC and Sequence # to Transaction Layer Packets**
 - ✓ **Buffers TLPs to allow retransmission**
- **Receive side:**
 - ✓ **Validates received TLPs by:**
 - **Checking the CRC code**
 - **Checking the Packet Sequence Number**
 - **Checking Phy Layer status for 8b/10b errors and framing errors**
 - ✓ **In the case of error:**
 - **Affected packet and following packets are discarded**
 - **NACK DLLP is sent to Transmitter to request retransmission**
 - ✓ **Transmitter time-out causes re-transmission if TLP completely lost**

Link Data Integrity – Retry Example

1. Three TLPs sent from A to B
2. Packet 2 corrupted
3. B detects corruption and issues Nak DLLP
4. A resends Packet 2 and following Packet
5. B acknowledges successful receipt of Packets



End-to-End Data Integrity - ECRC



- **Component internal errors are critical**
 - ✓ Header errors → TLP misrouting
 - ✓ Data corruption → application and system failure
- **End-to-end data integrity using ECRC**
 - ✓ Protecting from system-wide errors
 - ✓ Enabling upper layers error recovery
- **ECRC basics:**
 - ✓ Optional Capability – additional 32bit field (part of TLP)
 - ✓ Generated by the source component – applies to all invariant TLP fields
 - ✓ Switches must pass ECRC unchanged
 - ✓ Checked in the destination component – resulting behavior is device specific

PCI



SIG™

10TH YEAR ANNIVERSARY

End of Part 1

PCI



SIG™

10TH YEAR ANNIVERSARY

**PCI Express™
Protocol Overview
Part 2**

David Harriman

Error Signaling and Logging

- **Consistent mechanism for managing PCI Express errors**
 - ✓ Signaling using Completion Status
 - ✓ Signaling using Error Messages
 - ✓ Control and Logging using configuration registers
 - ✓ Baseline and Advanced Error Reporting
- **Error Classification and Mapping**
 - ✓ Correctable, Uncorrectable (Fatal, Non-Fatal)
 - ✓ Mapping of Physical, Data Link and Transaction Layer Errors

**Robust Data Integrity Complemented With
Standardized Error Signaling/logging**

Error Classification

Programmable Severity with
Advanced Error Support

- **Correctable Errors (Msg: ERR_COR)**
 - ✓ Hardware corrects the error without software impact beyond performance
 - ✓ Useful for link integrity profiling
 - ✓ Examples: Invalid Symbol in packet; CRC error
- **Uncorrectable – Non-Fatal (Msg: ERR_UNC)**
 - ✓ Hardware cannot correct the error
 - ✓ Transaction lost → impacts software
 - ✓ Link otherwise fully functional
 - ✓ Examples: Unsupported Request; Completer Abort
- **Uncorrectable – Fatal (Msg: ERR_FATAL)**
 - ✓ Hardware cannot correct the error
 - ✓ Link unreliable
 - ✓ Likely component/hierarchy reset required
 - ✓ Examples: Physical Layer Training Failure; Malformed Packet

Advanced Error Reporting Infrastructure

- **Uncorrectable Errors**
 - ✓ Mask, Status and Severity Controls
 - ✓ Severity controls Fatal or Non-Fatal
- **Correctable Errors**
 - ✓ Mask and Status Controls
- **Error Pointer identifies First Uncorrectable Error**
 - ✓ Detailed transaction-level information
- **Root Complex specific controls/status**
 - ✓ Requestor ID for faster fault isolation
 - ✓ Interrupt generation for handling of errors

Advanced Error Reporting – Error List

- **Physical Layer:**
 - ✓ Receiver Error, Training Error
- **Data Link Layer:**
 - ✓ Bad TLP, Bad DLLP, Replay Timeout, REPLY NUM Rollover, DLL Protocol Error
- **Transaction Layer:**
 - ✓ Poisoned TLP Received, ECRC Check, Unsupported Request (UR), Completion Timeout, Completer Abort, Unexpected Completion, Receiver Overflow, Flow Control Protocol Error, Malformed TLP

Hot-plug Support Requirements

- **Current status with PCI technology families**
 - ✓ Different usage models, user interfaces and HW support
 - ✓ PCI(X) SHPC, Compact PCI, Cardbus/PCMCIA
- **PCI Express Requirements**
 - ✓ Unified usage model – support all form factors
 - Card, Module, PC Card, Communications....
 - ✓ Lower complexity/cost
 - SHPC expensive for client systems
 - ✓ Legacy and Native OS-level support

PCI Express electricals and protocol developed from the ground up to support hot-plug

Native Hot-plug Support

- **Elements of Standard Usage Model**
 - ✓ **Indicators: Attention and Power**
 - ✓ **MRL and MRL Sensor**
 - ✓ **Electromechanical Interlock**
 - ✓ **Attention Button**
 - ✓ **Software User Interface**
 - ✓ **Slot Numbering**
- **Unified set of hot-plug models using simplified sensor/controller mechanisms**

Native Hot-plug Support (cont.)

- **Replaces SHPC as hot-plug mechanism for PCI Express**
 - ✓ SHPC continues to be the mechanism for parallel bus PCI implementations
- **Flexibility in placing buttons/indicators by using PCI Express in-band messaging:**
 - ✓ On the card (CompactPCI style)
 - ✓ On the chassis (SHPC style)
 - ✓ Not at all (PC Card style)

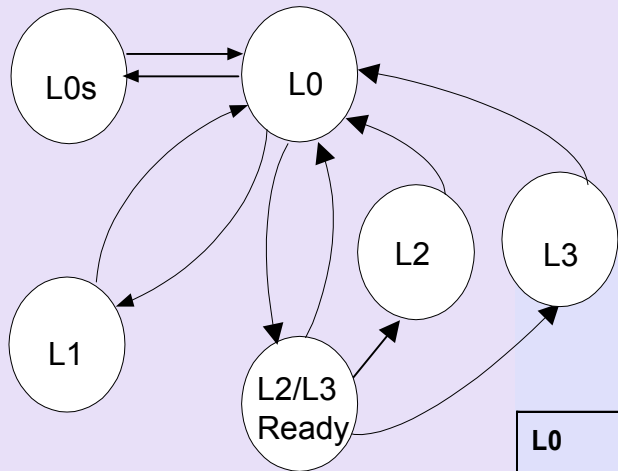
PCI Express Enables Hot Plug Capability for the Mainstream

Power Management

- **Builds on PCI Power Management (PM)**
 - ✓ Compatible with existing PCI PM software stacks
- **Device PM States: D0-D3_{hot/cold}**
- **Link PM States: L0, L0s, L1-L3**
- **Enhanced PM capabilities**
 - ✓ Aggressive power reduction through Active State PM (L0s, L1)
 - ✓ Improved PME using in-band messaging
 - ✓ Improved definition and SW control of Vaux

PCI Express Advances Platform PM While Preserving Software Investment

Link PM States Summary



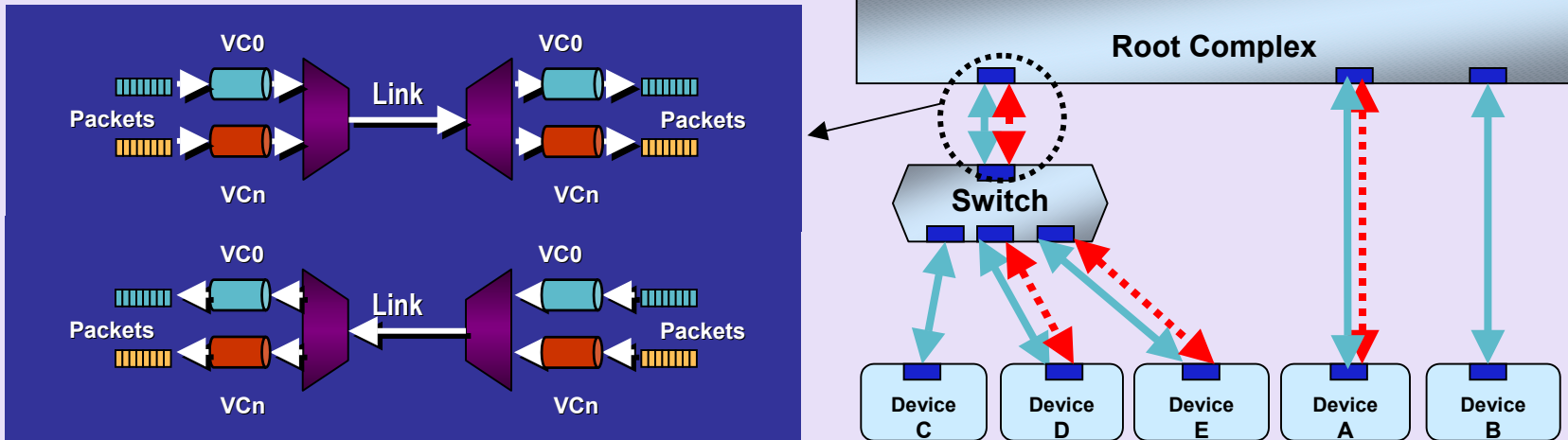
	L-State Description	Used by SW Directed PM	Used by Active State Link PM	Platform Reference Clocks	Platform Main Power	Component Internal PLL	Platform Vaux
L0	Fully active Link	Yes (D0)	Yes (D0)	On	On	On	On/Off
L0s	Standby State	No	Yes (D0)	On	On	On	On/Off
L1	Lower Power Standby	Yes (D1-D3 _{hot})	Yes (opt., D0)	On	On	On/Off	On/Off
L2/L3 Ready	Staging point for power removal	Yes	No	On	On	On/Off	On/Off
L2	Low Power Sleep State (all clks, main power off)	Yes	No	Off	Off	Off	On
L3	Off (zero power)	n/a	n/a	Off	Off	Off	Off

Multiple PM Levels for Power/performance Flexibility

Enhanced PM Capabilities

- **Active State PM for idle-yet-on devices**
 - ✓ Fine-grain PM on-top of SW-controlled capabilities
 - ✓ Downstream device determines link power state
 - ✓ Programming model for configuring Active State PM
 - ✓ Configuration depends on Endpoint latency requirements
- **Power Management Event (PME)**
 - ✓ In-band signaling of wake event to power manager
 - ✓ Compatible with existing PME SW
 - ✓ Includes precise geographical ID of requesting agent
 - ✓ Switches route PME messages from any downstream port to their upstream port.
 - ✓ Support for PCI Express-to-PCI bridges

Virtual Channels



- VCs are independent (ordering and flow control) paths
- Traffic Class (TC) labeling for differentiation of traffic
- Up to 8 Virtual Channels with associated servicing priorities
- Mapping of TCs to Virtual Channels for platform flexibility
- Configuration of TC/VC mapping and VC arbitration by software

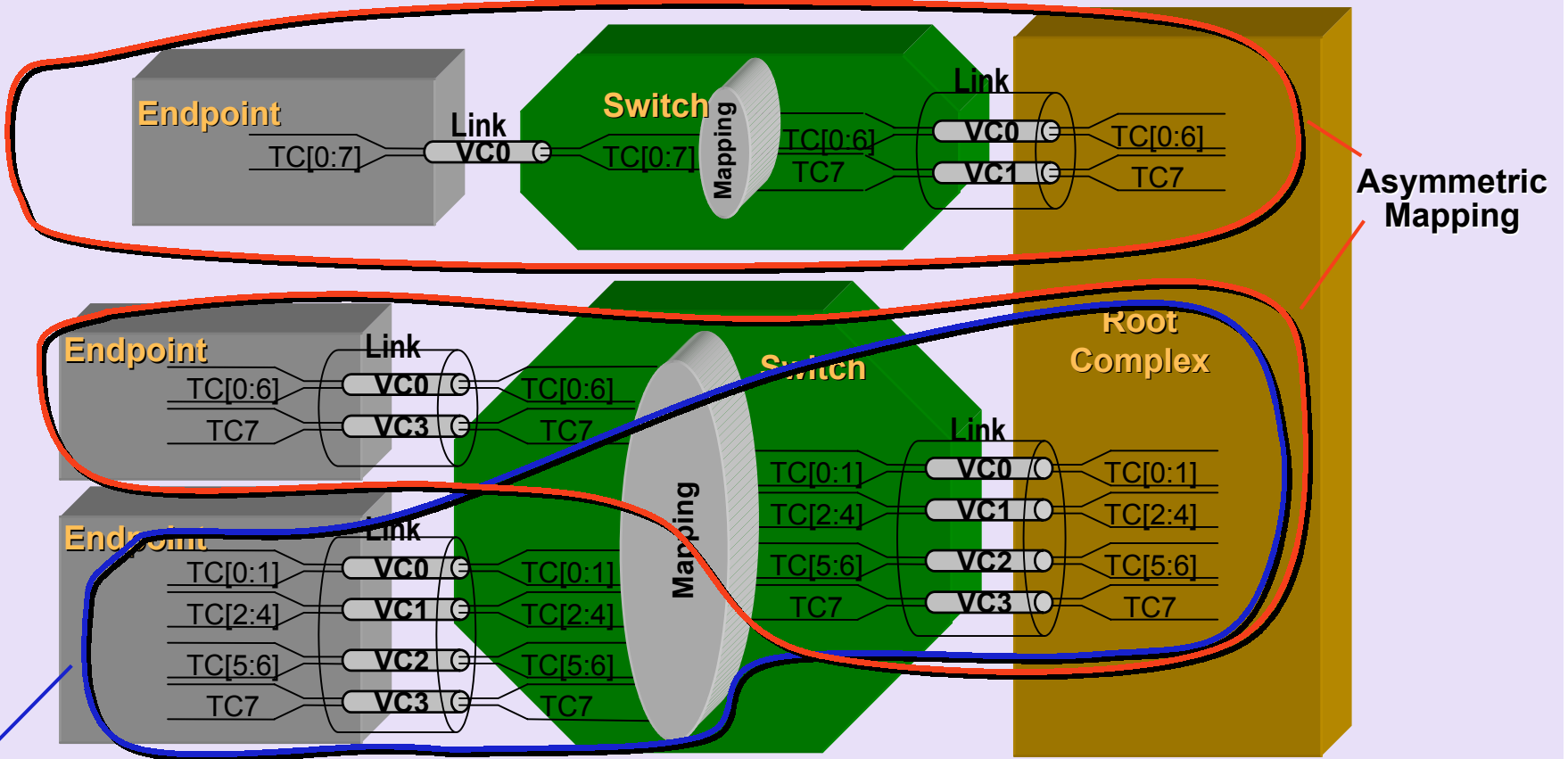
Virtual Channels = Support of Qos

Traffic Classes and TC/VC Mapping

- All transactions tagged with Traffic Class (TC) label
 - ✓ 8 TCs arranged in priority order (0=lowest, 7=highest)
 - ✓ TC0 required (default); support of other TCs is optional
- Each TC is an independently ordered stream
 - ✓ Ordering rules only apply to transactions of the same TC
- Example of supported TC/VC Configurations:

Supported VC Configurations	TC/VC Mapping Options
VC0	TC(0-7)/VC0
VC0, VC1	TC(0-6)/VC0, TC7/VC1
VC0-VC3	TC(0-1)/VC0, TC(2-4)/VC1, TC(5-6)/VC2, TC7/VC3
VC0-VC7	TC[0:7]/VC[0:7]

TC/VC Mapping Example

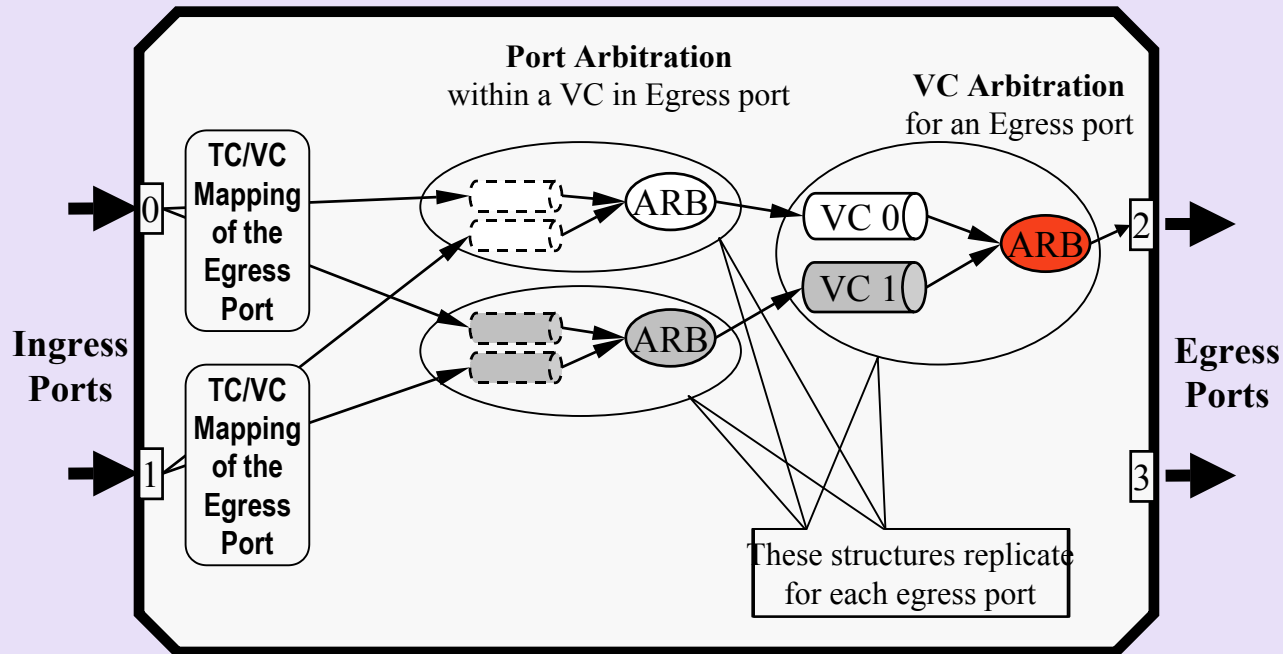


Symmetric Mapping

Asymmetric Mapping

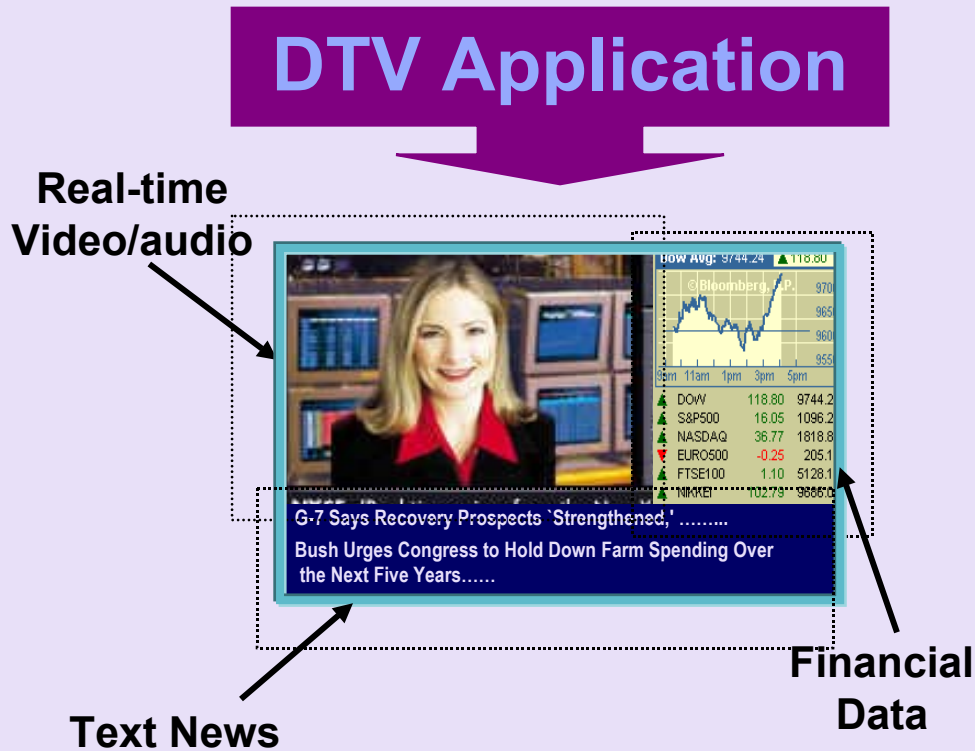
TC/VC configuration model supports both symmetric mapping and re-mapping

VC and Switch Arbitration



- **Routing of Traffic:**
 - ✓ Determine Egress Port based on address/routing info
 - ✓ Determine target VC within Egress based on TC/VC map
- **Port Arbitration:** arbitration among traffic targeting same VC/Egress Port
 - ✓ Fixed round-robin (RR), programmable Weighted RR, programmable time-based WRR
- **VC Arbitration:** arb. among traffic from different VC competing for the same Link
 - ✓ Strict priority, Round-robin, Weighted RR

Why Isochronous?

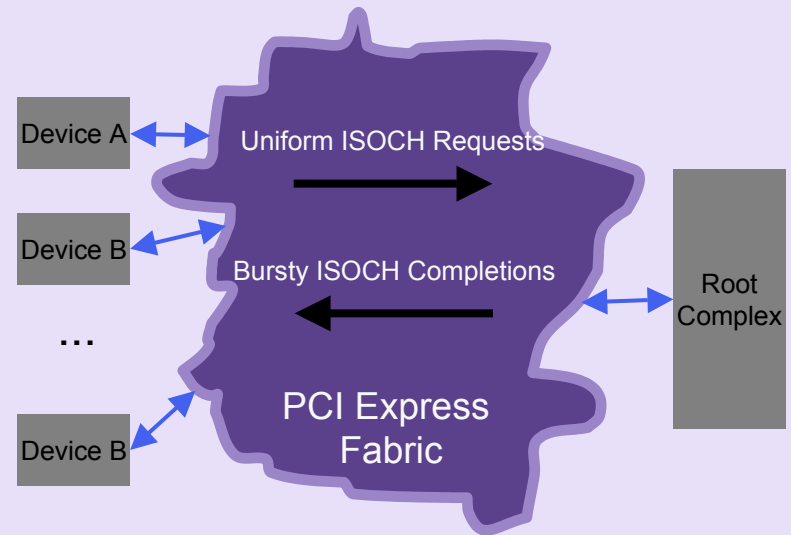


- **Isochronous**
 - ✓ Deadline sensitive traffic
- **Digital multimedia applications**
 - ✓ DTV, video on demand, Video-conferencing
 - ✓ Peripheral connectivity (USB, 1394, e.t.c)
- **Client- and Comm/Server-level capability**

PCI Express Isoc Capability - Enabler for Streaming Media Applications

Isochronous Support

- Virtual Channels and Traffic Class labeling as a foundation
- Connection Admission – Isoc resource management
 - ✓ Fully exposed software interface (HW API thru PCI Express Config Space)
- Traffic Regulation and Policing - the {N, T, t} Contract
 - ✓ $t = 100 \text{ nsec}$, $T = 12.8 \text{ usec}$
- Bandwidth and latency control within fabric
 - ✓ Time-based WRR arbitration



Isoc = Platform Level Capability
PCI Express Provides Interconnect Portion

Other PCI Express Enhancements

- **Extended Configuration Mechanism**
- **Power Budgeting/Limiting Control**
 - ✓ **Control of Power Limits per Slot**
- **Unique Device Identification**
 - ✓ **IEEE 64-bit Extended Unique Identifier (EUI-64)**

Summary

- **PCI Express advances overall platform capabilities while preserving PCI architecture and software investments**
- **Layered approach and scaleable features provide a foundation for technology stability**
- **New capabilities such as VCs and Native Hot-Plug enable important emerging applications**

Call to Action

- **Comprehend PCI Express technology in your product roadmaps**
- **Invest early in PCI Express building blocks and infrastructure to establish market leadership**
- **Stay engaged with PCI-SIG – help review and ratify the PCI Express Specification**



Want More Info on PCI Express?

- PCI-SIG Web Site

- ✓ <http://www.pcisig.com>

- Intel PCI Express Web Site

- ✓ [http://developer.intel.com/technology/PCI Express](http://developer.intel.com/technology/PCI%20Express)
 - White Paper, FAQ

**Thank you for attending the 2002 PCI-SIG
Developers Conference and your continued
efforts in advancing PCI I/O Technology!**

**Visit www.pcisig.com for additional information
or contact PCI-SIG at (+1) 503.291.2569**