

Technical Introduction to IEEE 1394

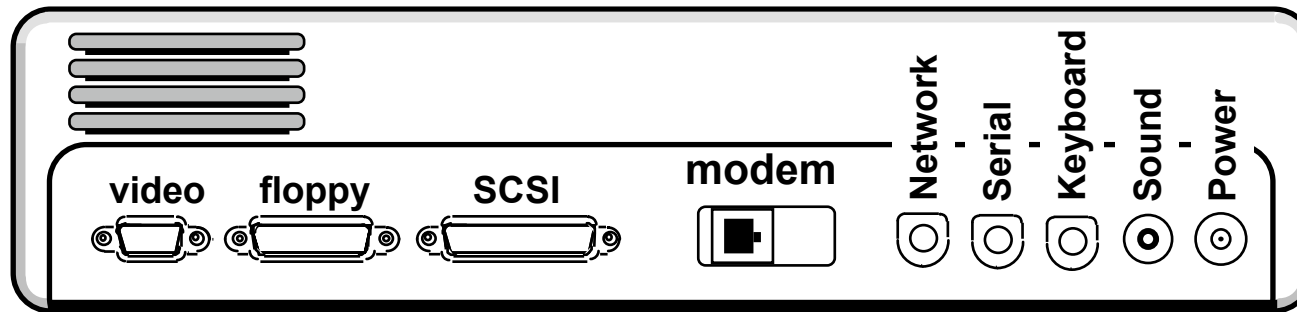
Michael Johas Teener

Chair P1394c WG

mike@teener.com

Background

(the way things were)



- No I/O Integration
 - lots of PCB area, silicon & software
 - no common architecture
- Hard to change
 - no realtime transport
 - performance not scalable

Goals

- Low cost, high performance ergonomic system interconnect
- Compatible architecture with other IEEE busses
 - Follow IEEE 1212 CSR (Control and Status Register) standard
- Peer-to-peer read/write memory model for asynchronous services
 - NOT an I/O channel
- Isochronous service
 - Multimedia transport

“Isochronous” ??

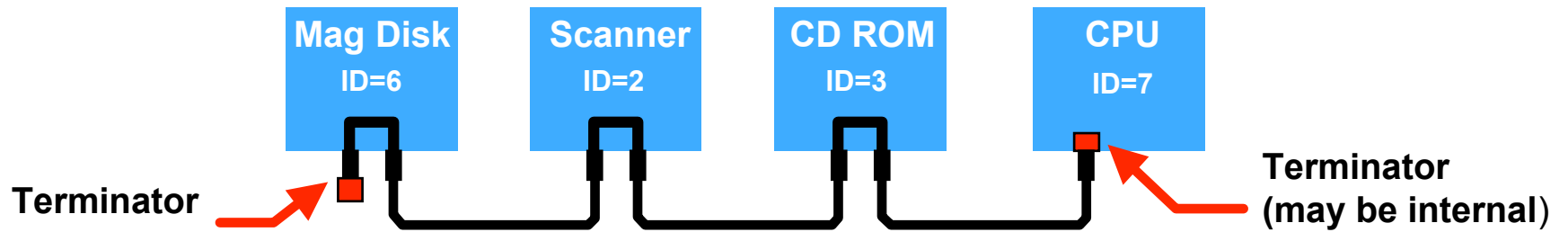
- Iso (same) chronous (time) :
 - Uniform in time
 - Having equal duration
 - Recurring at regular intervals

Data type	Sample size and rate	Bit rate
ISDN	8 kHz x 8 bits	64 Kbit/sec
CD	44.1 kHz x 16 bits x 2 channels	1.4 Mbit/sec
DAT	48 kHz x 16 bits x 2 channels	1.5 Mbit/sec
Video	25-30 frames/sec	1.5 – 216 Mbit/sec

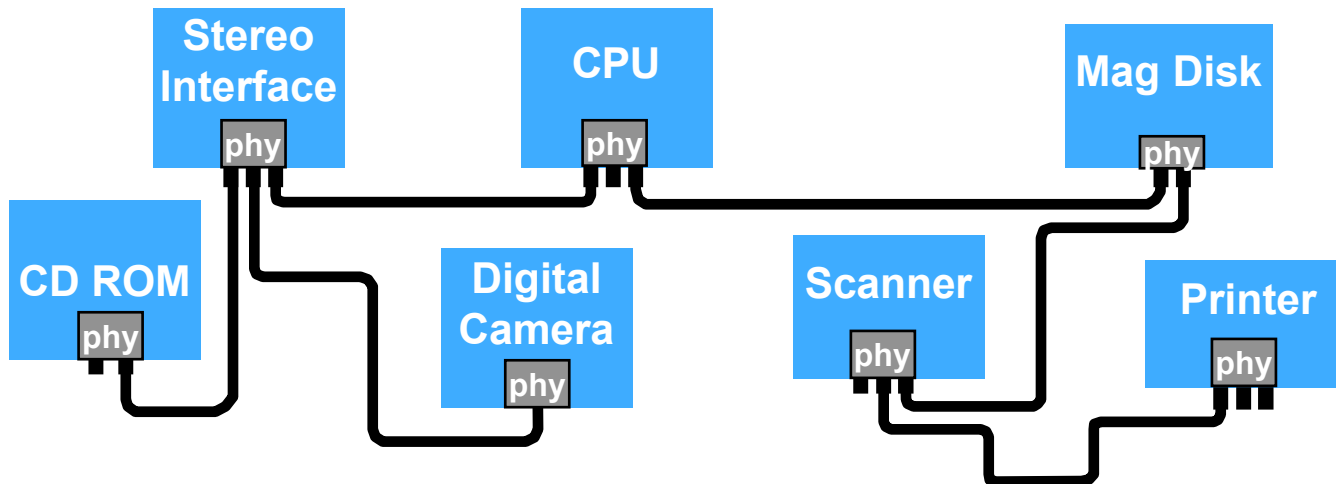
Asynch Vs. Isoch

- Asynchronous transport
 - “Guaranteed delivery”
 - Reliability more important than timing
 - Retries are OK
- Isochronous transport
 - “Guaranteed timing”
 - Late data is useless
 - Never retry

Unsupervised!



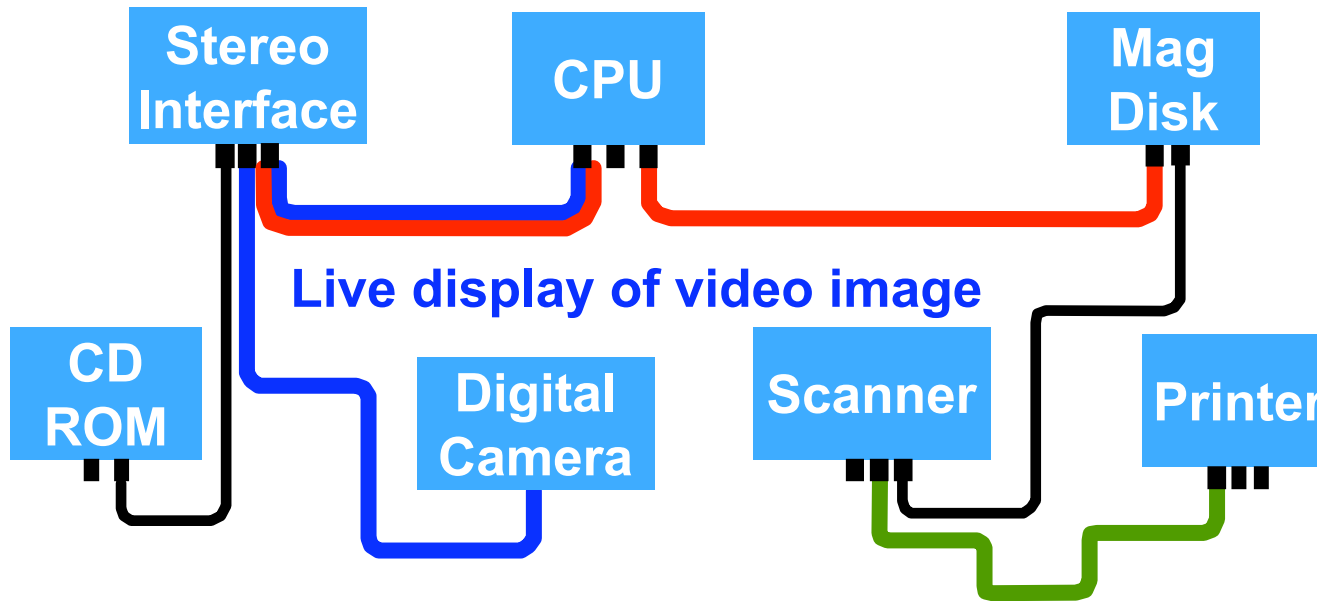
SCSI is typical “supervised cabling” — daisy-chain; manual or fixed addresses; terminators at ends; devices with internal terminations must be at one end



Serial Bus is “unsupervised cabling” — “non-cyclic network”; automatic address selection, no terminators, locations are arbitrary

Data paths (peer-to-peer)

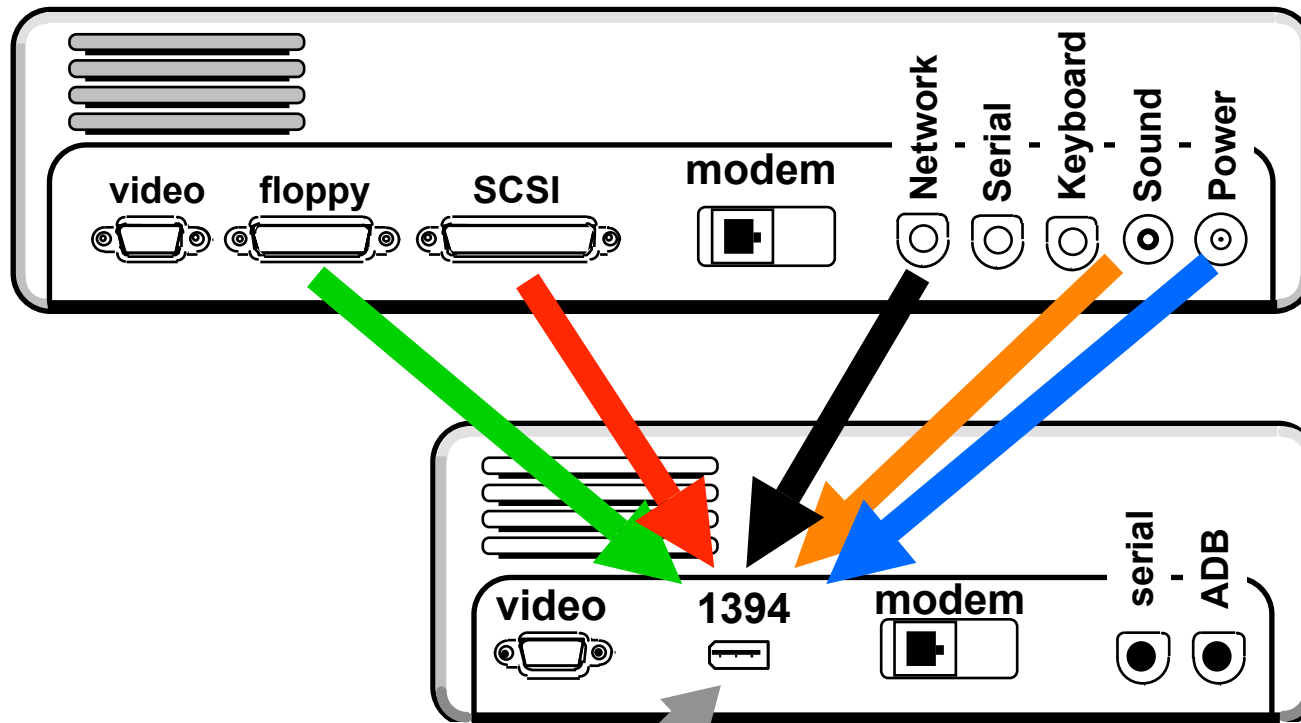
Digitized sound direct playback



Live display of video image

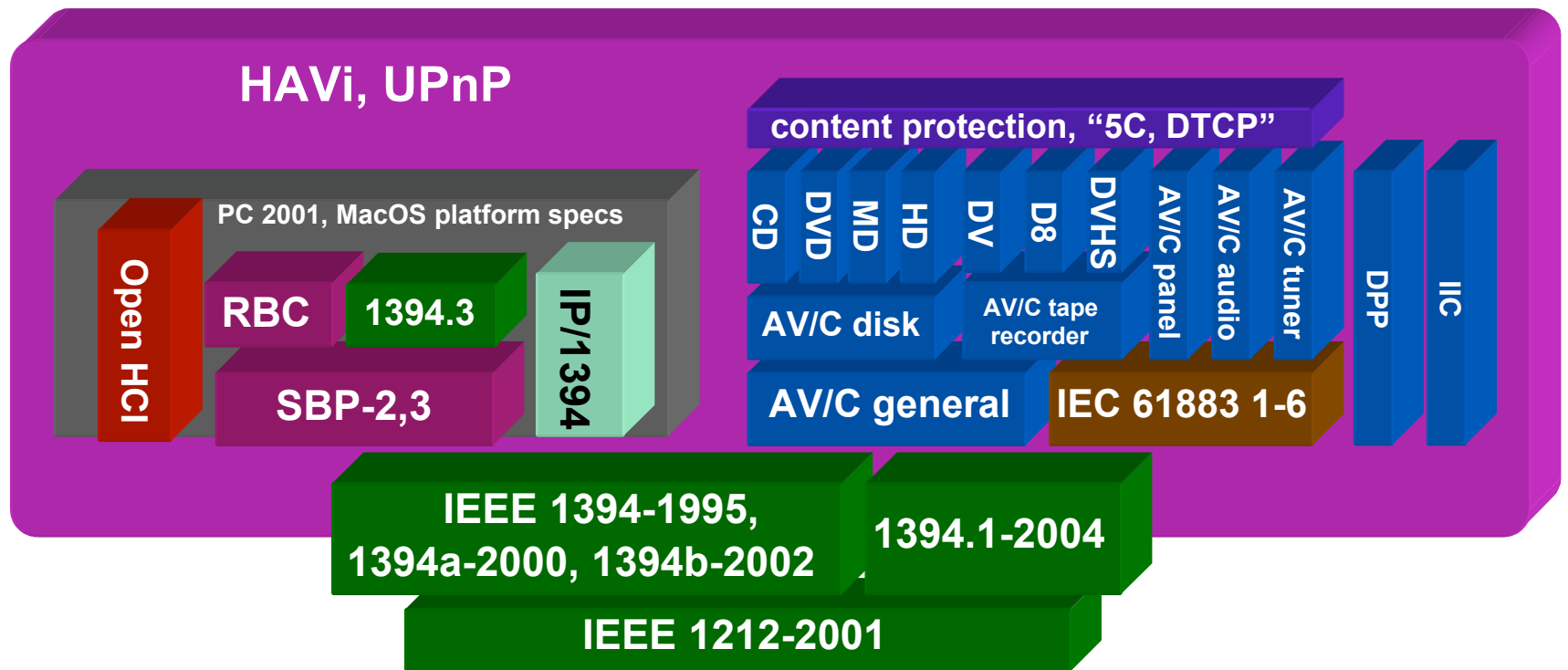
Direct printing of scanned image

Clean up the desktop cable mess!



plus telephone/voice, hi-fi
sound, compressed video

1394 family of specifications



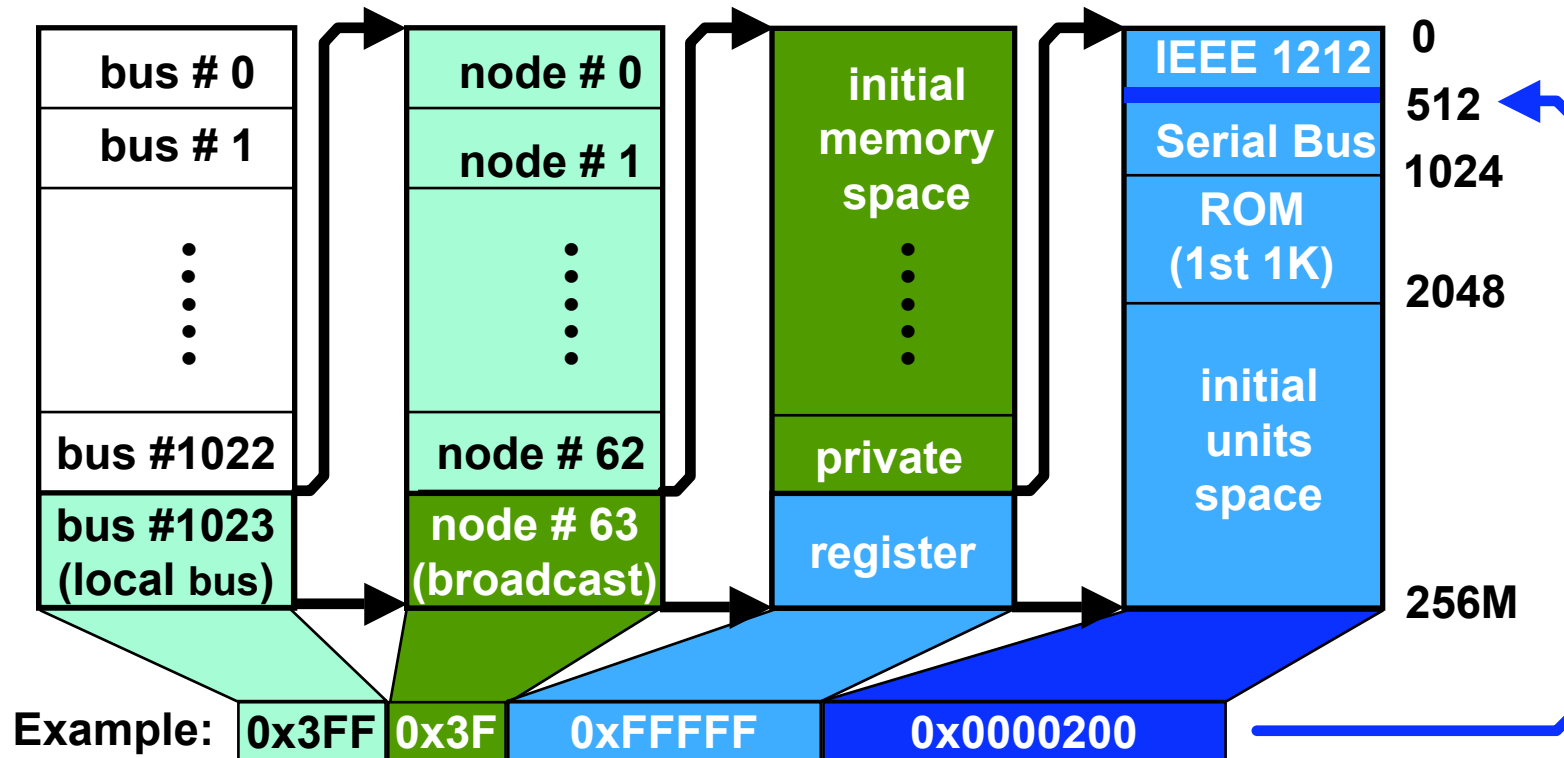
Key specifications

- IEEE 1394-1995, 1394a-2000, 1394b-2002 High Speed Serial Bus
 - “Memory-bus-like” logical architecture, isochronous support
 - Serial implementation of 1212 architecture
- IEEE 1212-2001 CSR Architecture
 - Standardized addressing, well-defined control and status registers, configuration declaration, standardized transactions
- “Higher layer” protocols
 - NCITS.325-1998 SBP-2 integrates DMA into I/O process
 - RBC (for mass storage) and IEEE 1394.3 PPDT (for printers)
 - IEC 61883 and 1394TA AV/C standards define control and data for AV devices
 - RFC 2734 defines Internet Protocol (v4) over 1394
 - IPv6 and DHCP also as RFCs
 - Digital Transport for Content Protection (“5C”/DTCP)
 - IEEE 1394.1 for bridges, IIC for instrumentation and industrial control, DPP for consumer cameras/printers, etc.

Some terminology

- “quadlet” - 32-bit word
- “node” - basic addressable device
- “unit” - part of a node, defined by a higher level architecture ... examples:
 - SBP disk drive (X3T10 standard)
 - A/V device - VCR, camcorder (IEC 61883 standard)

IEEE 1212 addressing



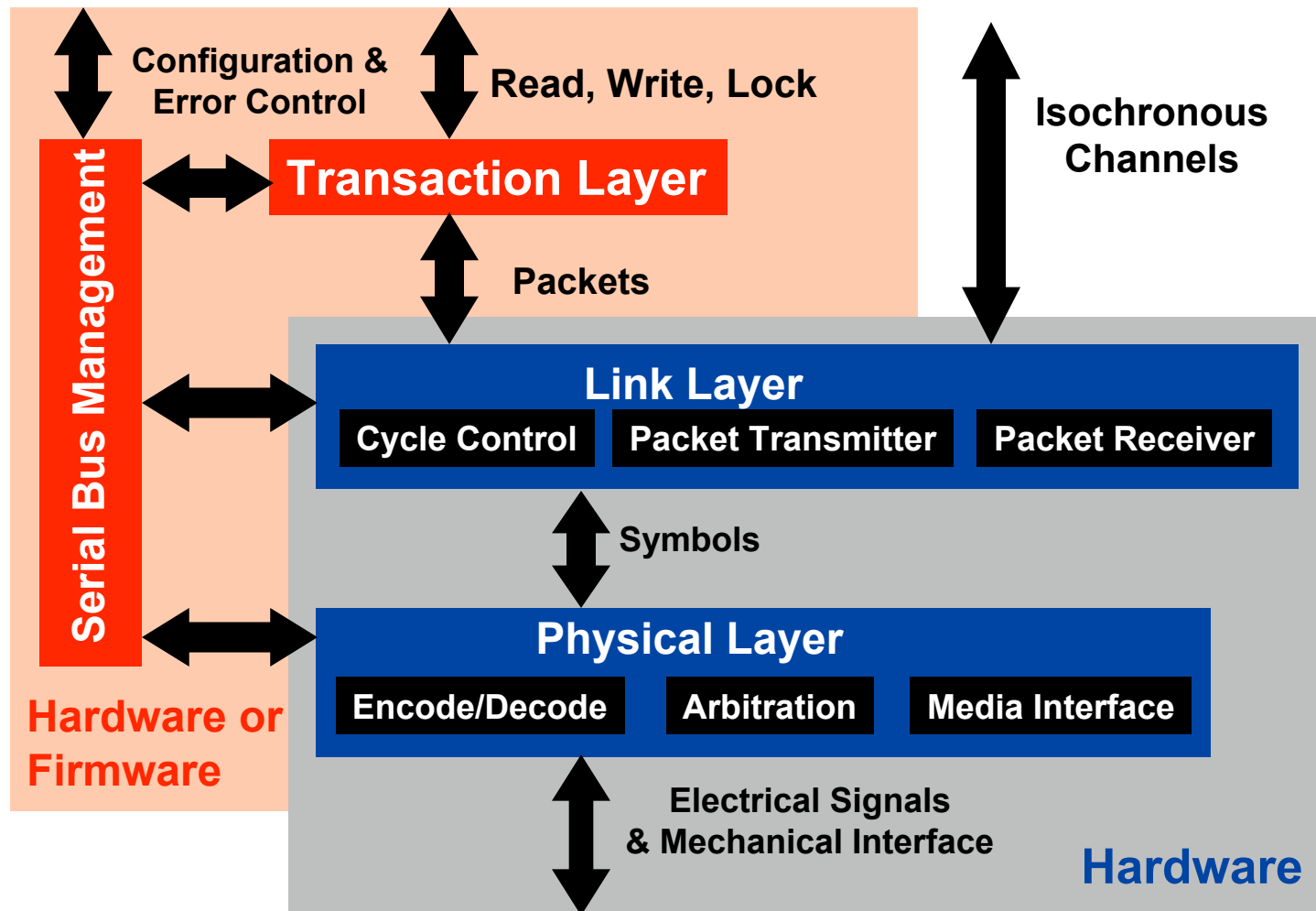
Example:

0x3FF 0x3F 0xFFFF 0x0000200

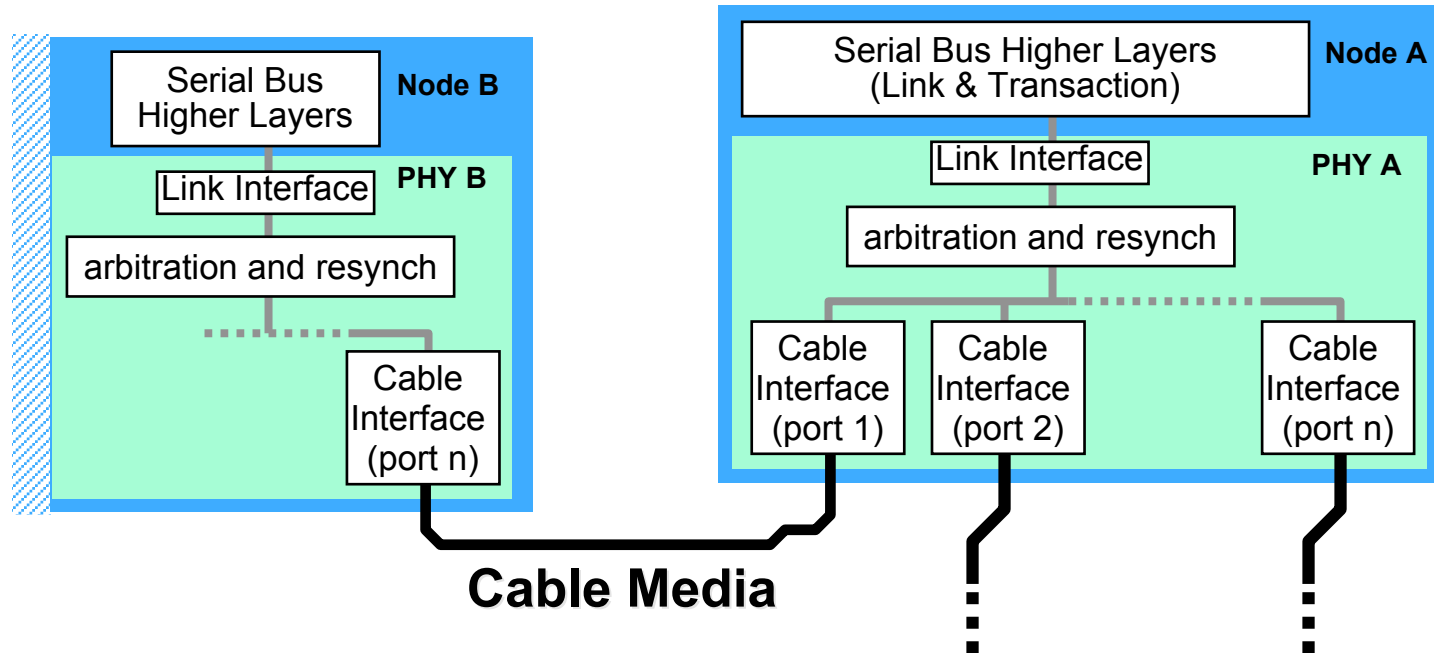
= all cycle timer registers on local bus

- 1394 uses "64-bit fixed" addressing for asynchronous transactions

IEEE 1394 protocol Stack

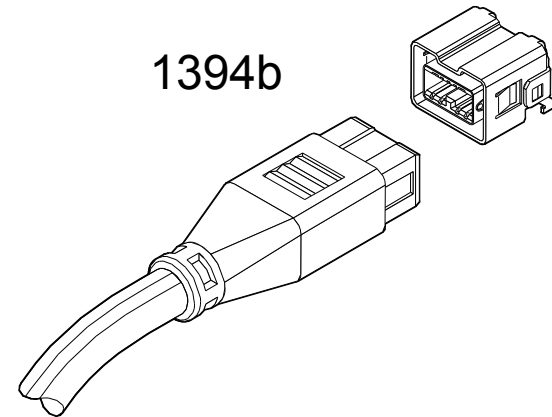
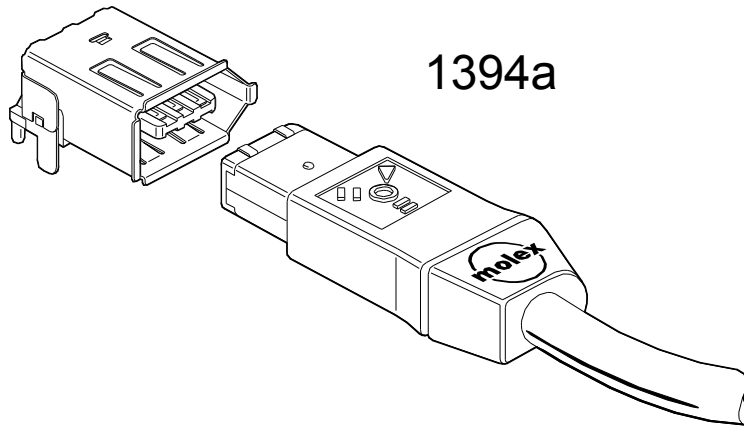


Cable interface



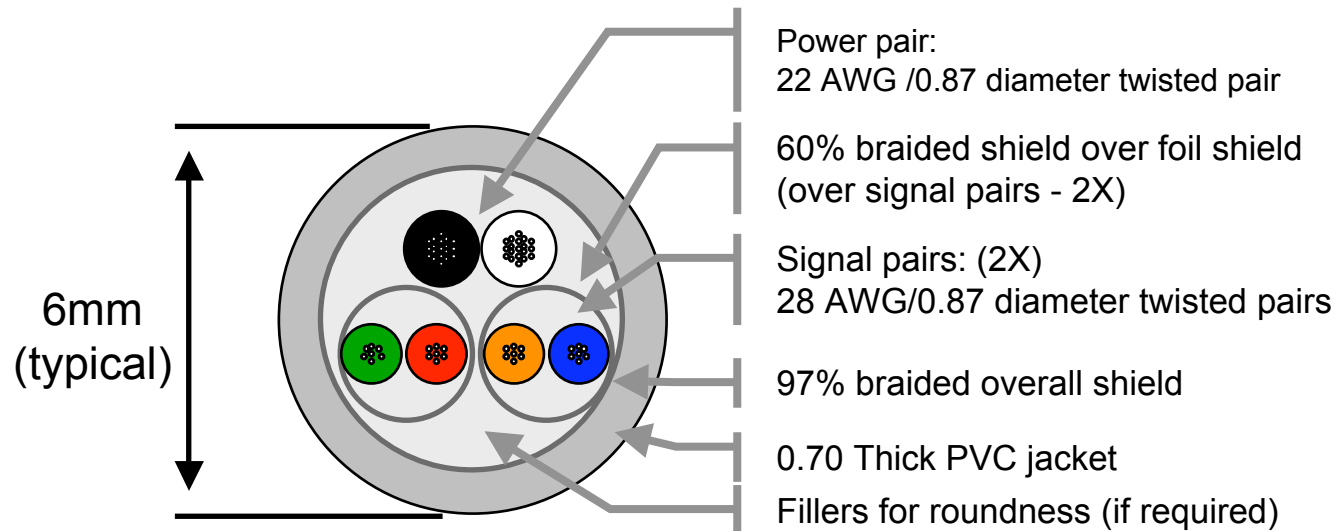
- PHY transforms point-to-point cable links into a logical bus
- Cables and transceivers are bus repeaters

Standard cable media



- 3-pair shielded cable
 - Two pairs for data transport
 - One pair for peripheral power
 - 2-pair used for most CE equipment
- Small and rugged connector
 - Two sockets in the same area as one mini-DIN socket
 - Even smaller for 1394b
- CMOS transceiver
 - Differential drive, low voltage, low current
 - DC-coupled for legacy, AC-coupled in new Beta mode

Standard cable media example



- Capable of operation at 400 Mbit/sec for 4.5m
 - Slightly thicker wire allows 10 meter operation
 - 1394b encoding allows at least 3.2 Gbit/sec

Standard cable interface features

- Live attach/detach
 - System protected from power on/off cycling
 - Higher layers provide simple management
- Power carried by cable
 - 30-8 VDC, up to 1.5A per link
 - Mac desktops provide 15-25W, PowerBooks 7W
 - “SuperHub” reference design provides 90W continuous/120W peak
 - 1394 TA defined management protocols
 - Total available power is system dependent
 - Node power requirements must be declared in configuration ROM
 - Basic source capabilities and sink requirements included in “self-ID” packets at system initialization
 - Nodes can either source or sink power
 - Multiple power sources on one bus provide additional flexibility

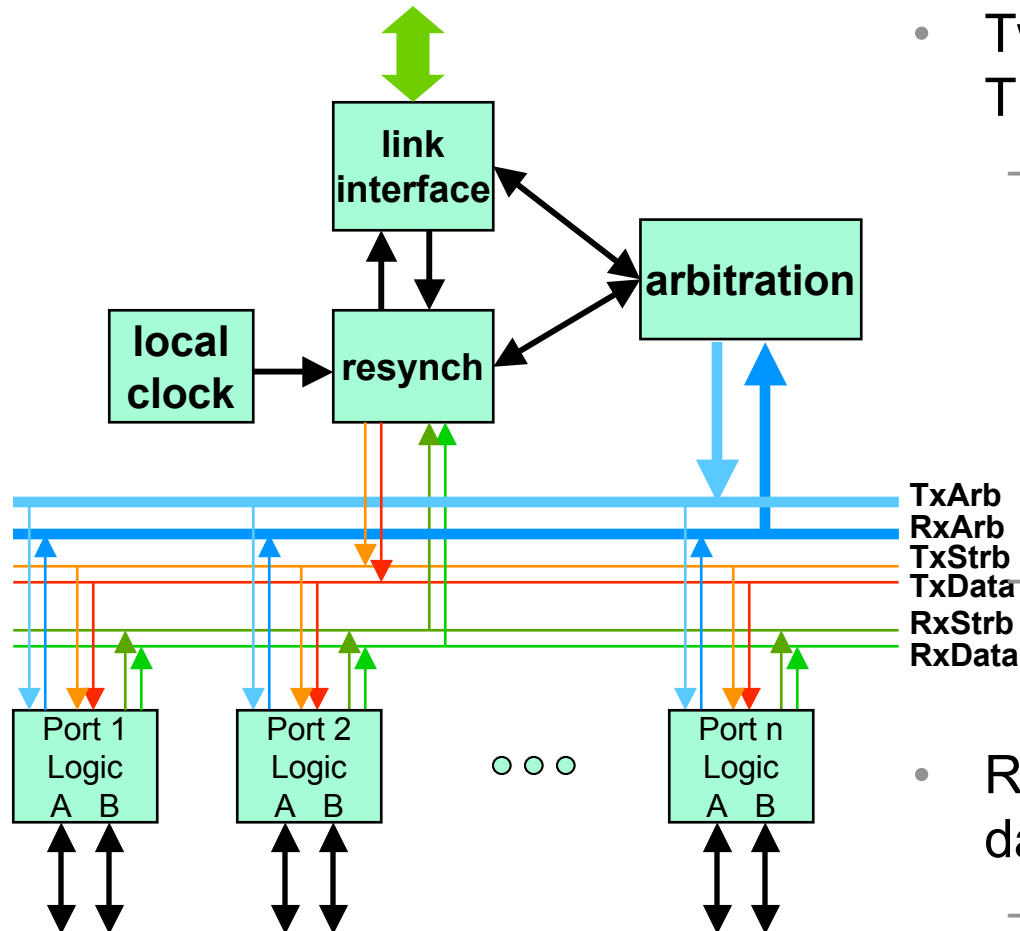
Alternate media

- New media defined in 1394b for long distances
 - Requires new “beta” mode
- Long distance at S100 can be done via cat 5 UTP
 - same as 100 Mbit Ethernet
 - 1394b encoding scrambles data better than 100baseTX, easier to meet FCC class B emissions limits
- Long distance at S200 and above currently requires optical fiber
 - 50 micron graded index OK for S3200 up to 100m
 - Plastic optical fibers good for S200 up to 50m
 - New technology fibers good for S400 up to 100m, but not mentioned in current draft
 - Hard polymer-coated fiber good for S200 up to 100m
- p1394c defines S800 cat 5e operation
 - 1394 TA is defining S400 UTP for automobiles (30m)

Physical layer

- 98.304 Mbit/sec half duplex data transport
 - Data reclocked at each node
 - 196.608 (2x), 393.216 (4x) Mbit/sec
 - 1394b provides 1x, 2x, 4x, 8x, 16x, 32x full duplex
- Data encoding
 - Data and strobe on separate pairs
 - 1394b uses improvement of FibreChannel's 8b10b encoding
 - Automatic speed detection
- Arbitration signaling done with DC levels
 - "1", "0" and "Z" states
- Fair and priority access
 - Tree-based handshake arbitration
 - Automatic assignment of addresses

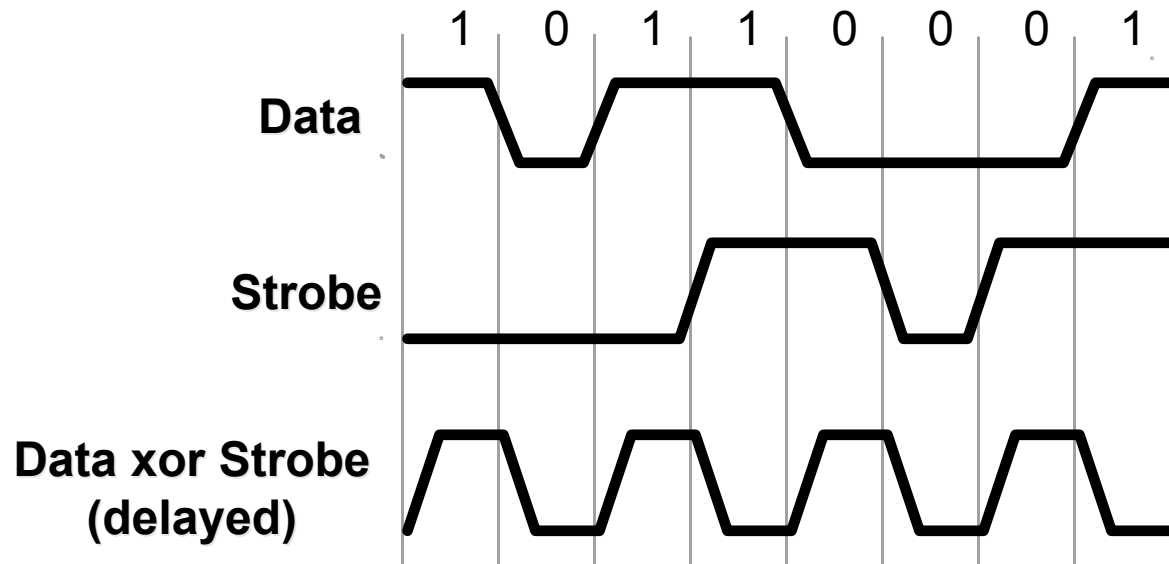
Example cable PHY IC



- Two twisted pairs for data: TPA and TPB
 - For legacy:
 - TPA is transmit strobe, receive data
 - TPB is receive strobe, transmit data
 - Both are bidirectional signals in arbitration
 - For Beta mode:
 - TPA is receive
 - TPB is transmit
- Reclocks repeated packet data signals using local clock
 - 1394 is “pleisiochronous”

Data-strobe encoding

(legacy)

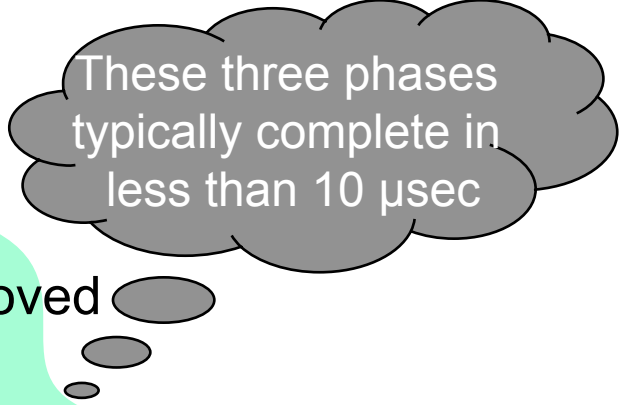


- Either Data or Strobe signal changes in a bit cell, not both
 - Gives 100% better jitter budget than conventional clock/data

New “beta” connection model

- High speed PHYs communicate using continuously transmitted dual simplex signal using 8B10B encoding
 - Data is first scrambled, then sent to 8B10B encoder
 - No repeating data, reduces EM radiation
 - Control symbols are not standard IBM 8B10B codes, but have Hamming distance 2 from each other and all data codes
 - Little chance of data error confusing protocols
 - Also scrambled by same mechanism
- Definitely cool technology, Alistair Coles of HP

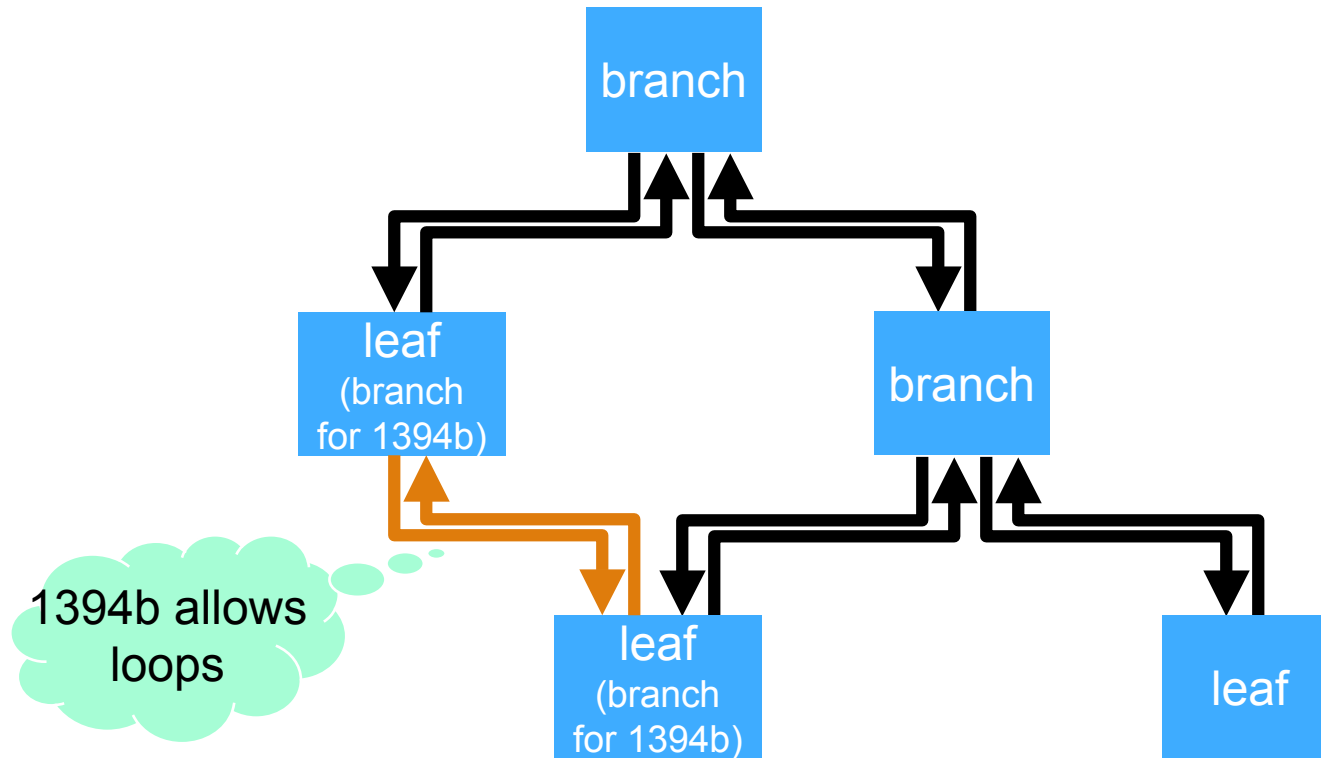
Cable arbitration phases



These three phases typically complete in less than 10 μ sec

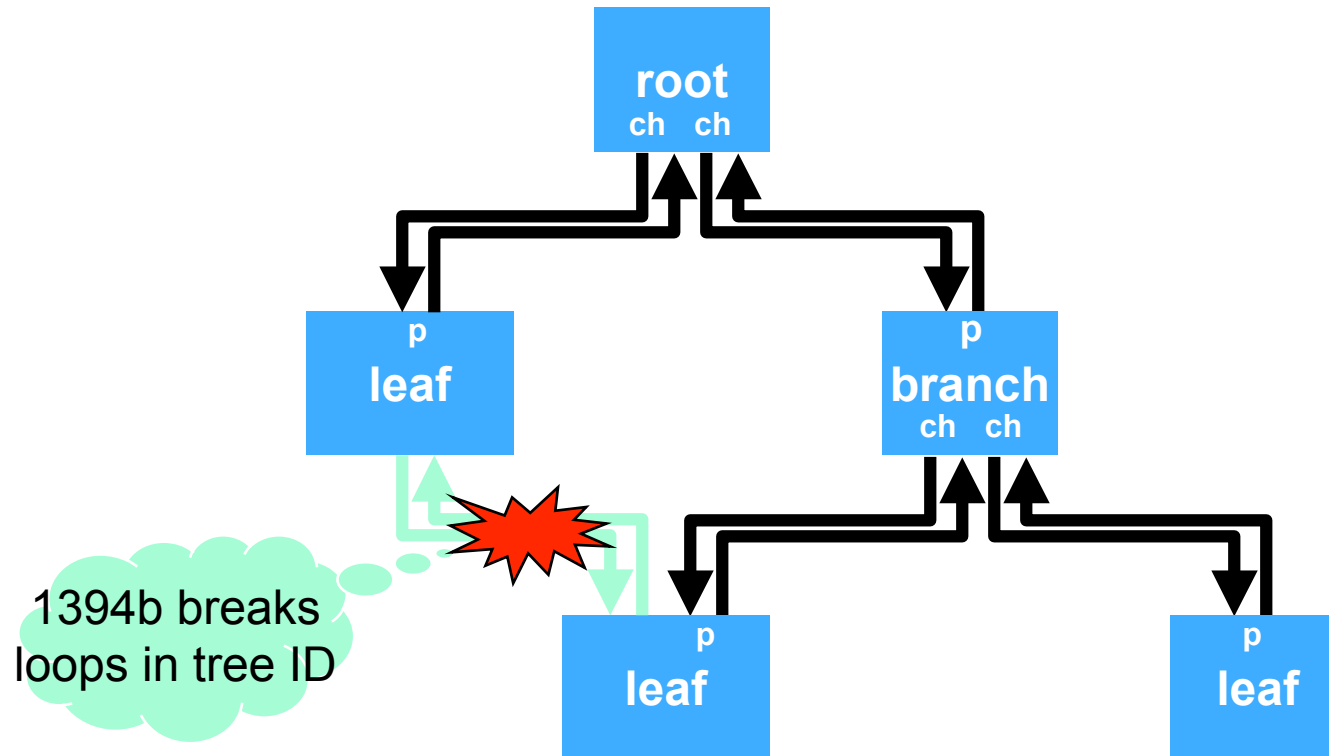
- **Reset**
 - Forced whenever node is attached or removed
- **Tree Identification**
 - Transforms a non-cyclic topology into a tree with a “root” node at its head
 - 1394b beta connections can be broken, allows arbitrary topology at reset
- **Self Identification**
 - Assigns physical node number (Node ID), exchange speed capabilities with neighbors, broadcast basic capabilities
- **Normal Arbitration**
 - Requests sent towards root in 1394-1995 and 1394a (requests and data cannot overlap)
 - Sent towards currently transmitting node in 1394b beta mode (requests can overlap data)

Tree identification #1



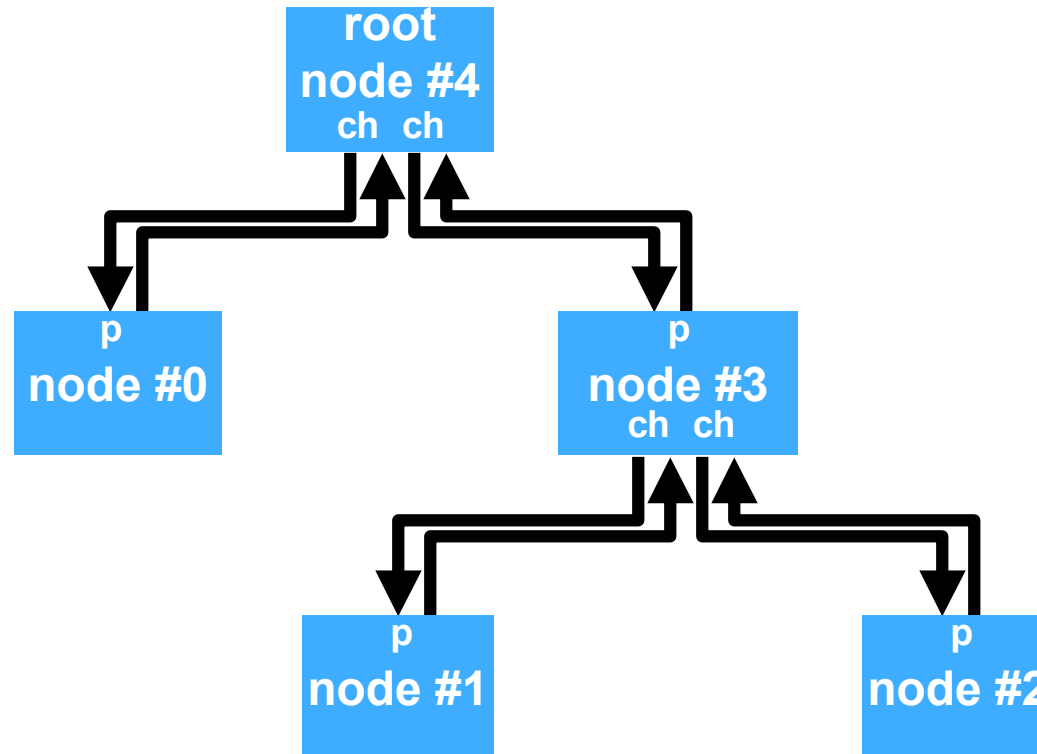
- After reset, each node only knows if it is a leaf (one connected port) or a branch (more than one connected port)

Tree identification #2



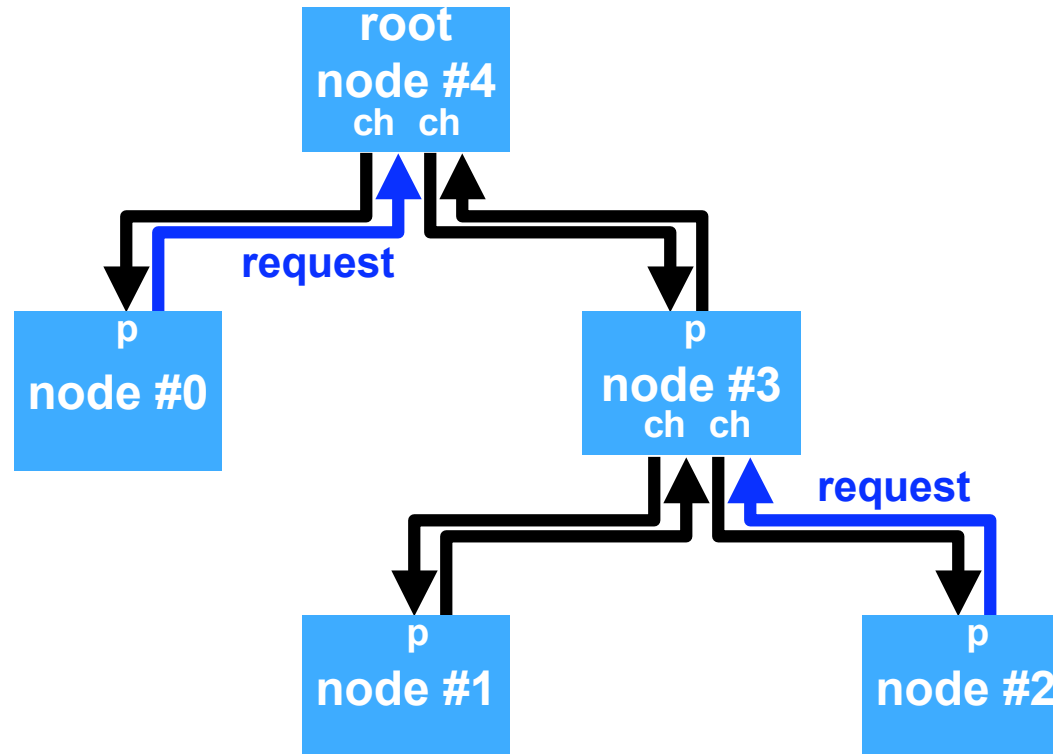
- After Tree ID process, the Root node is determined and each port is labeled as pointing to a child or a parent
 - Root assignment is “sticky”, will normally persist across a bus reset.
 - 1394b breaks any loops on beta connections even before Tree ID starts

Self identification



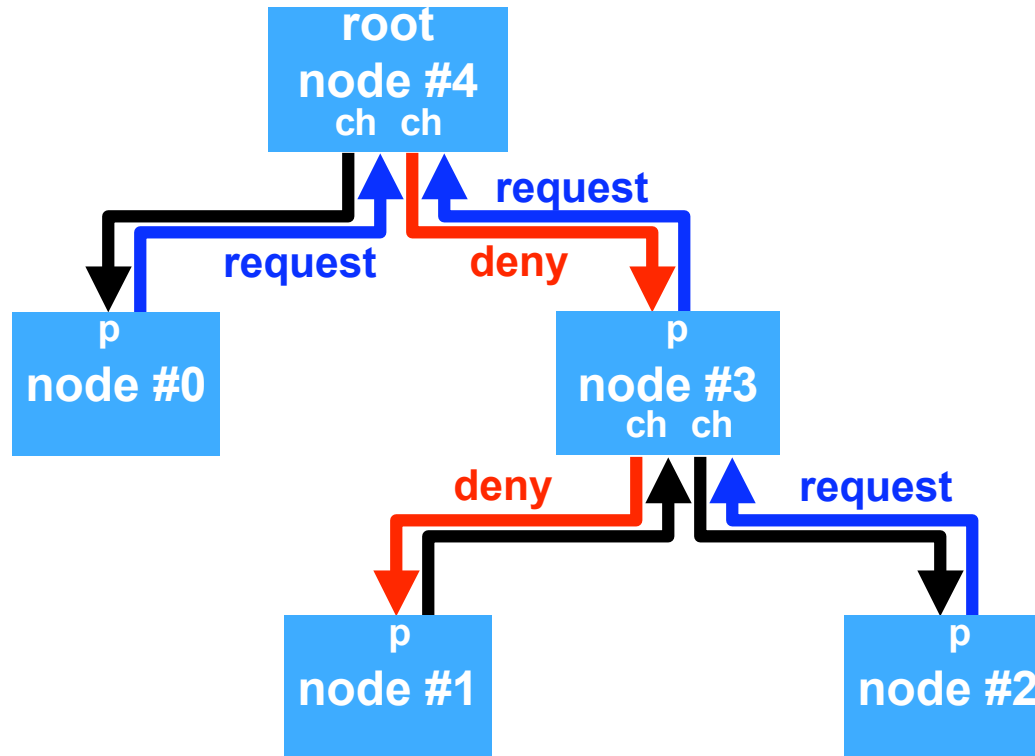
- After the self ID process, each node has a unique physical node number, and the topology has been broadcast

Legacy arbitration #1



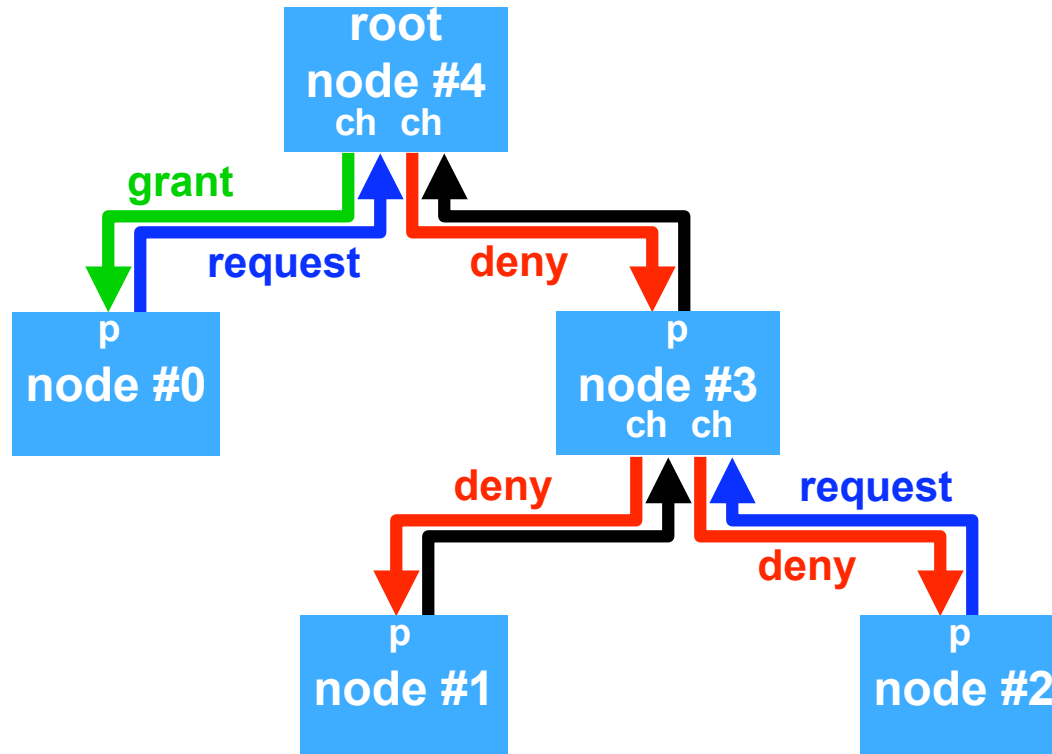
- Suppose nodes #0 and #2 start to arbitrate at the same time, they both send a request to their parent ...

Legacy arbitration #2



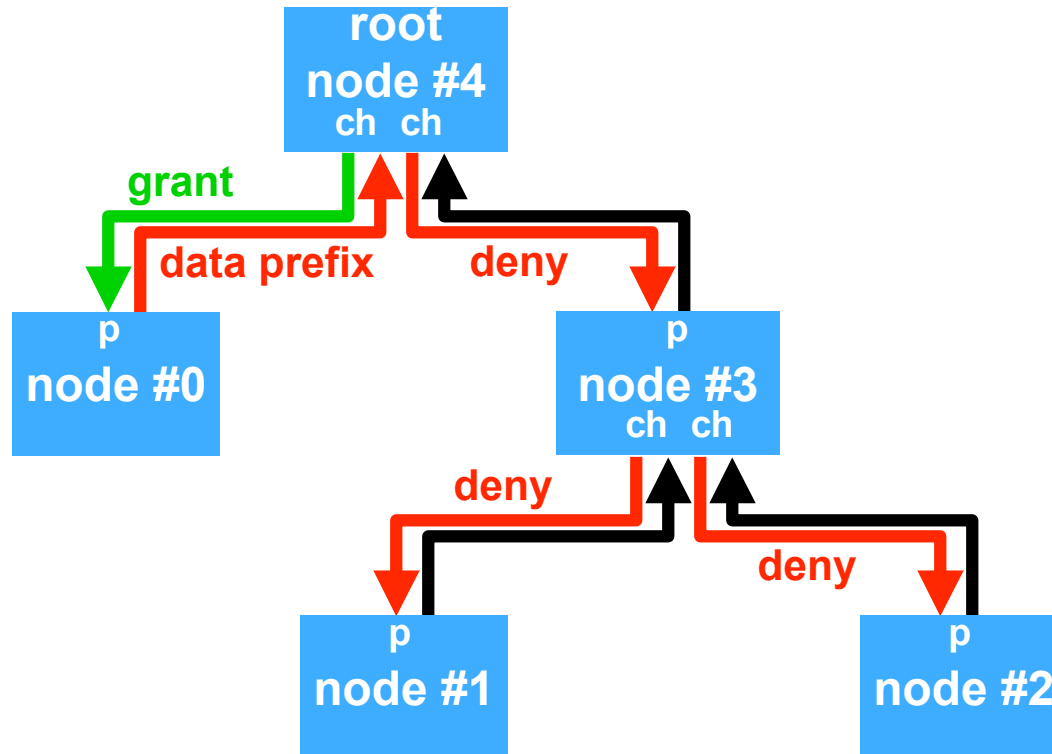
- The parents forward the request to their parent and deny access to their other children ...

Legacy arbitration #3



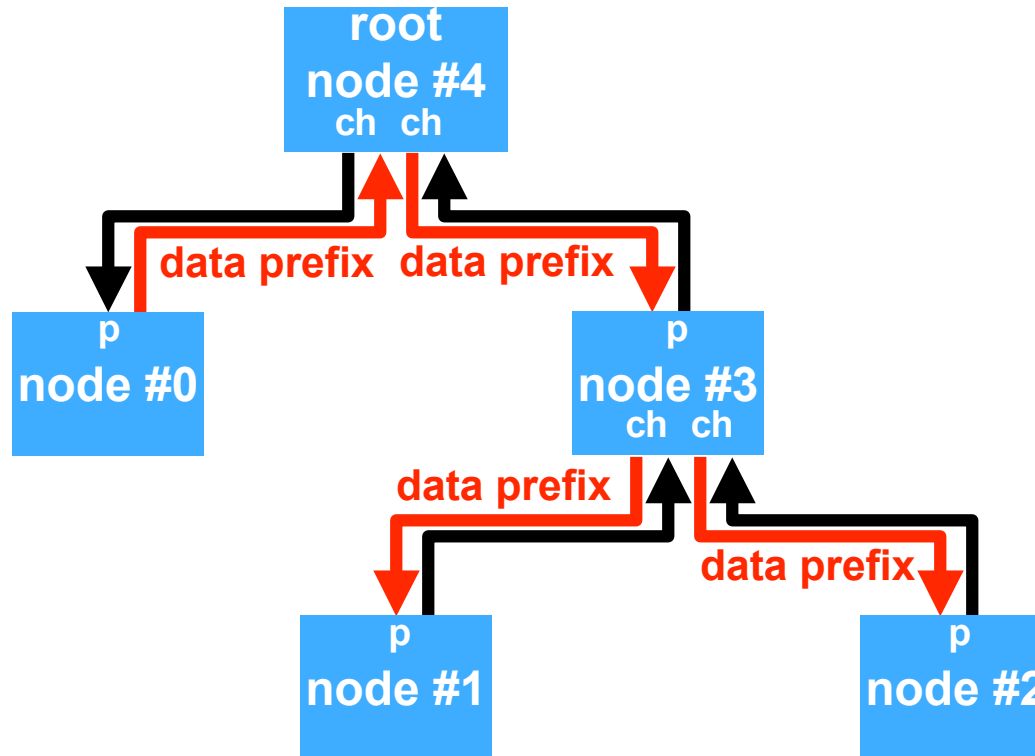
- The root grants access to the first request (#0), and the other parent withdraws it's request and passes on the deny ...

Legacy arbitration #4



- The winning node #0 changes its request to a data transfer prefix, while the losing node #2 withdraws its request ...

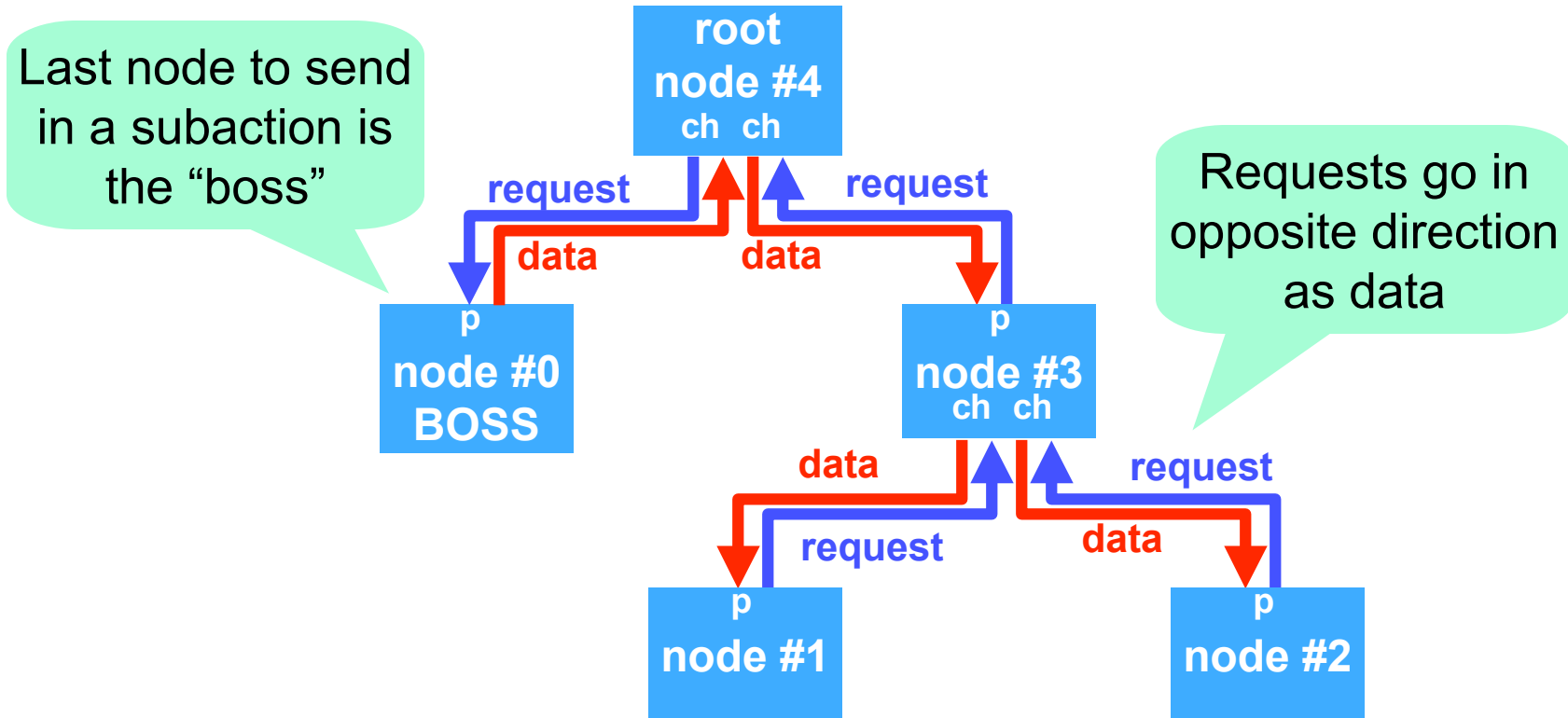
Legacy arbitration #5



- The parent of node 1 sees the data prefix and withdraws the grant, and now all nodes are correctly oriented to repeat the packet data (a "deny" is a "data prefix!") ...

Beta arbitration

- Dual simplex connections means that arbitration is overlapped with data transmission!



Beta arbitration concepts

- General rule: send requests continually, and in every direction that is not carrying packet information
 - Send highest priority request from all ports and attached PHY
- Node sending data is the “BOSS”
 - If BOSS knows that data being sent is the end of a subaction, it can issue a grant to the highest priority request
 - Examples: sending an ACK, sending a broadcast packet ...
- Many fallback methods, error recovery straight-forward
 - More cool technology, David LaFollette of Intel, Jerry Hauck and Michael Johas Teener of Zayante

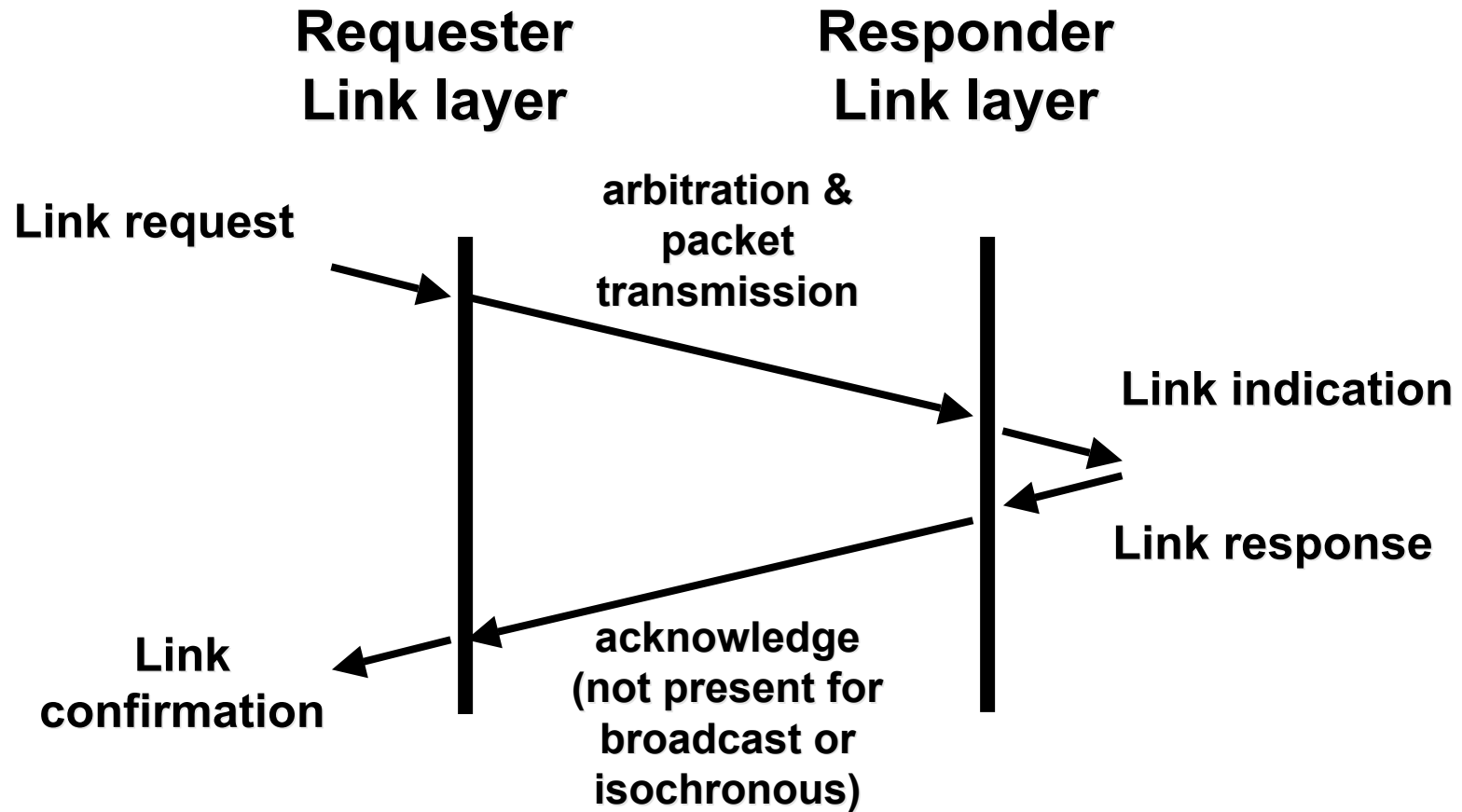
Beta and Legacy interoperation

- “Bilingual” modes specified for “border” PHYs
 - A border PHY is one that has at least one port operating in legacy (“DS” encoded) mode and one port operating in beta mode
 - Ports can be either “legacy” (DS-only), “beta”, or “bilingual”
 - Bilingual ports support both beta and legacy connections
- Beta-only PHYs are quite simple!
 - Border PHYs somewhat more difficult, but just more logic
 - Bilingual ports have difficult analog designs

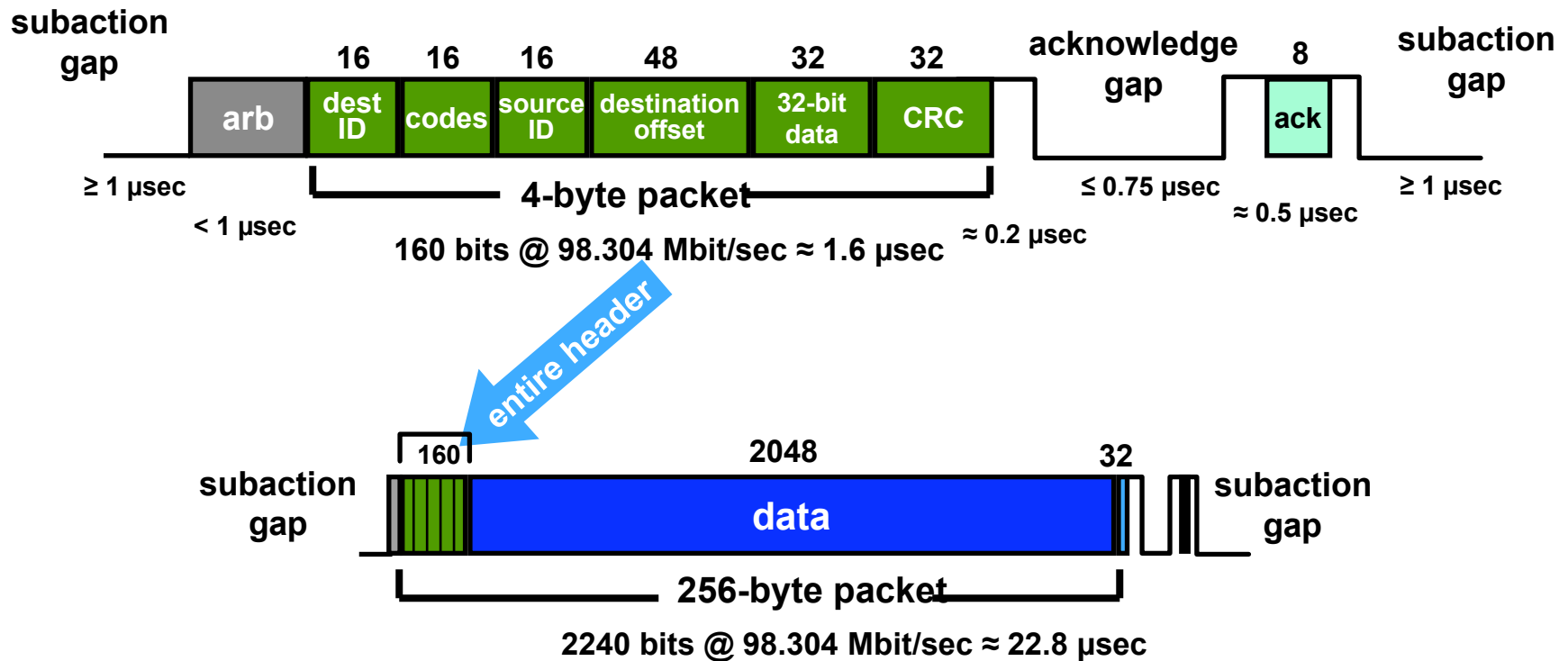
Link layer

- Implements acknowledged datagram service
 - Called a "subaction" of arbitration, packet transmission, and acknowledge
- Isochronous access provides multiple "channels" each 125 μ sec "cycle" period
 - Channels are broadcast packets to one of 64 channel addresses
 - Channel count limited by available bandwidth
 - Channel numbers and bandwidth managed by higher layers

Link layer operation

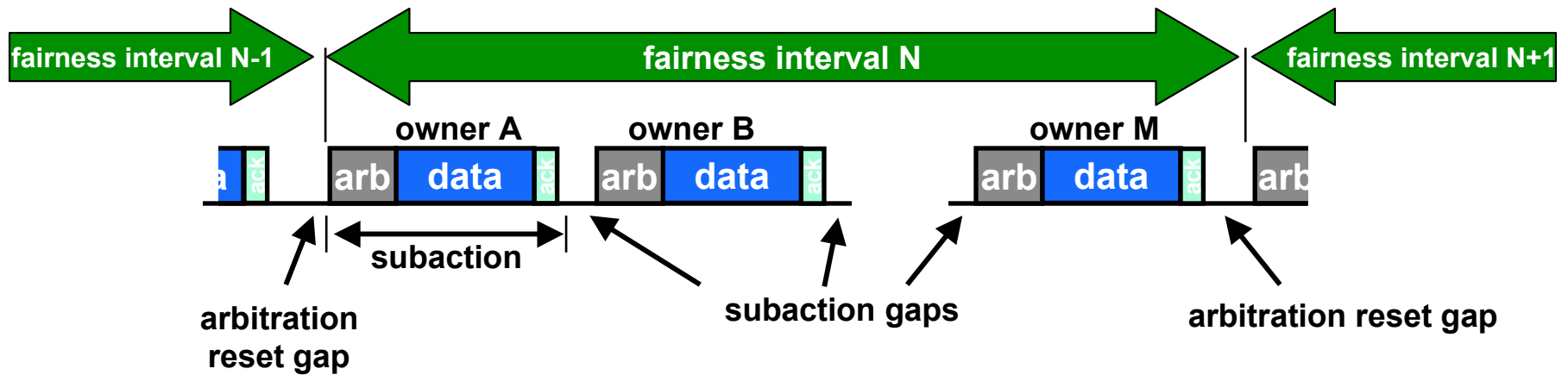


Example packets



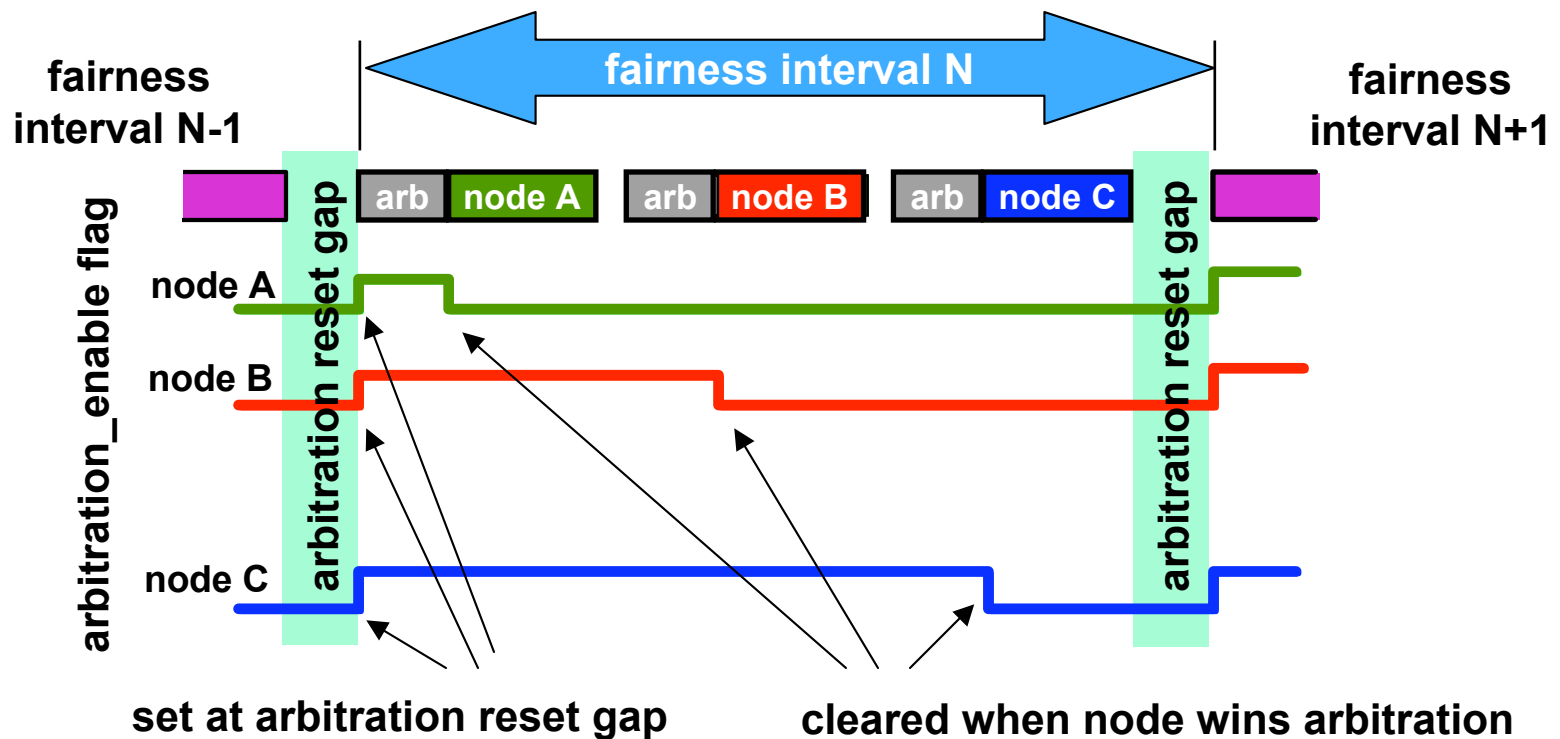
- Actual efficiency very good
 - 10 Mbytes/sec information throughput including all of the SBP disk protocol using 100 Mbit/sec rate (~80%)

Fairness interval (legacy)



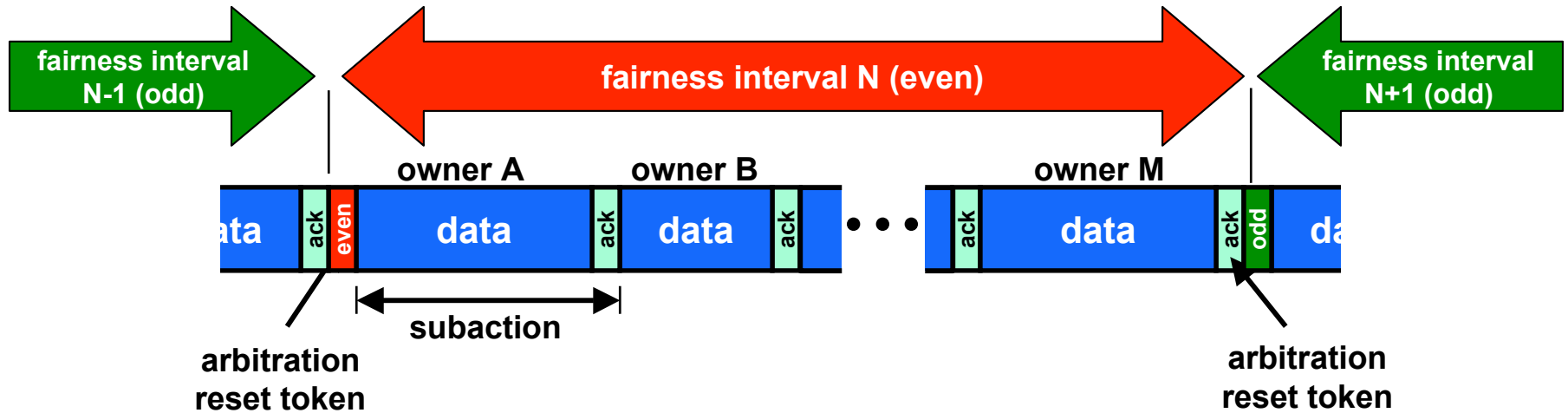
- Fairness Interval is bounded by “arbitration reset gaps”
- Reset gaps are longer than normal subaction gaps

Fair arbitration (legacy)



- Each node gets one access opportunity each fairness interval
 - special case for isochronous data

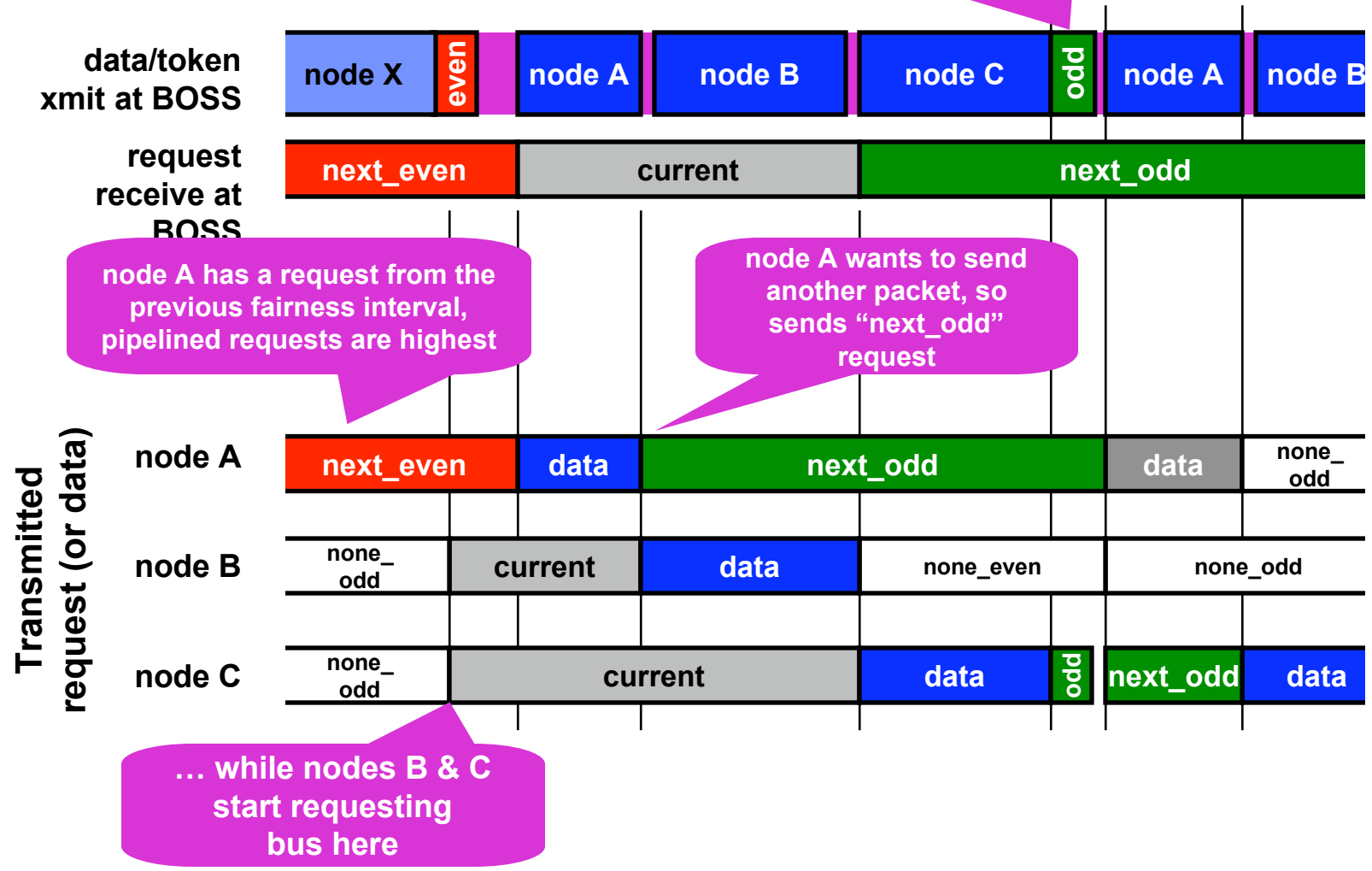
Fairness interval (Beta)



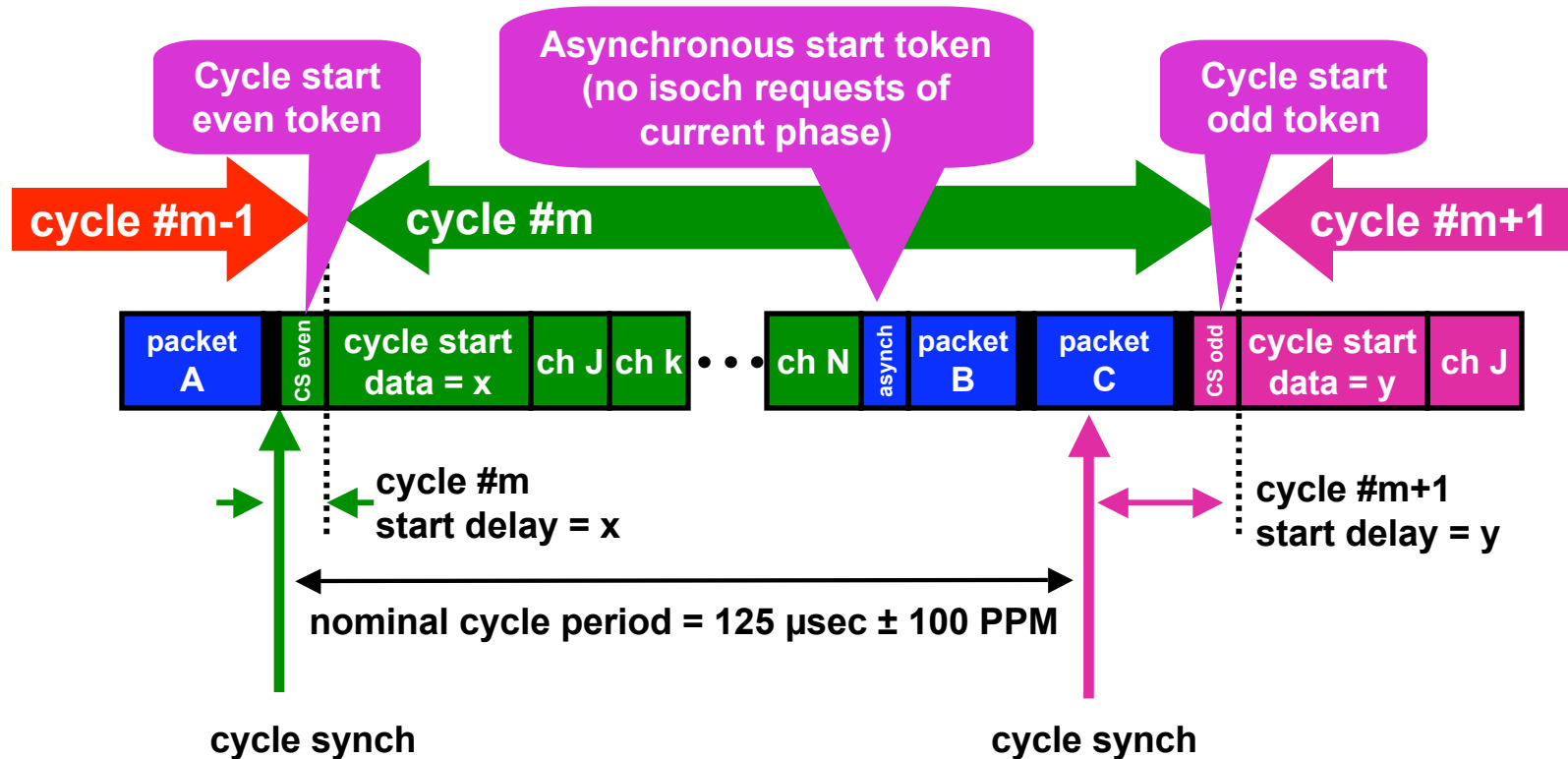
- Fairness Interval is bounded by “arbitration reset tokens”
- Arbitration reset tokens are labeled “ODD” or “EVEN”, as are the fairness intervals
 - Allows pipelining of arbitration for the next fairness interval

Fair arbitration (Beta)

Node C is BOSS, and is only seeing "next_odd" or "idle_even", so sends arb_reset_odd



Cycle structure (Beta)

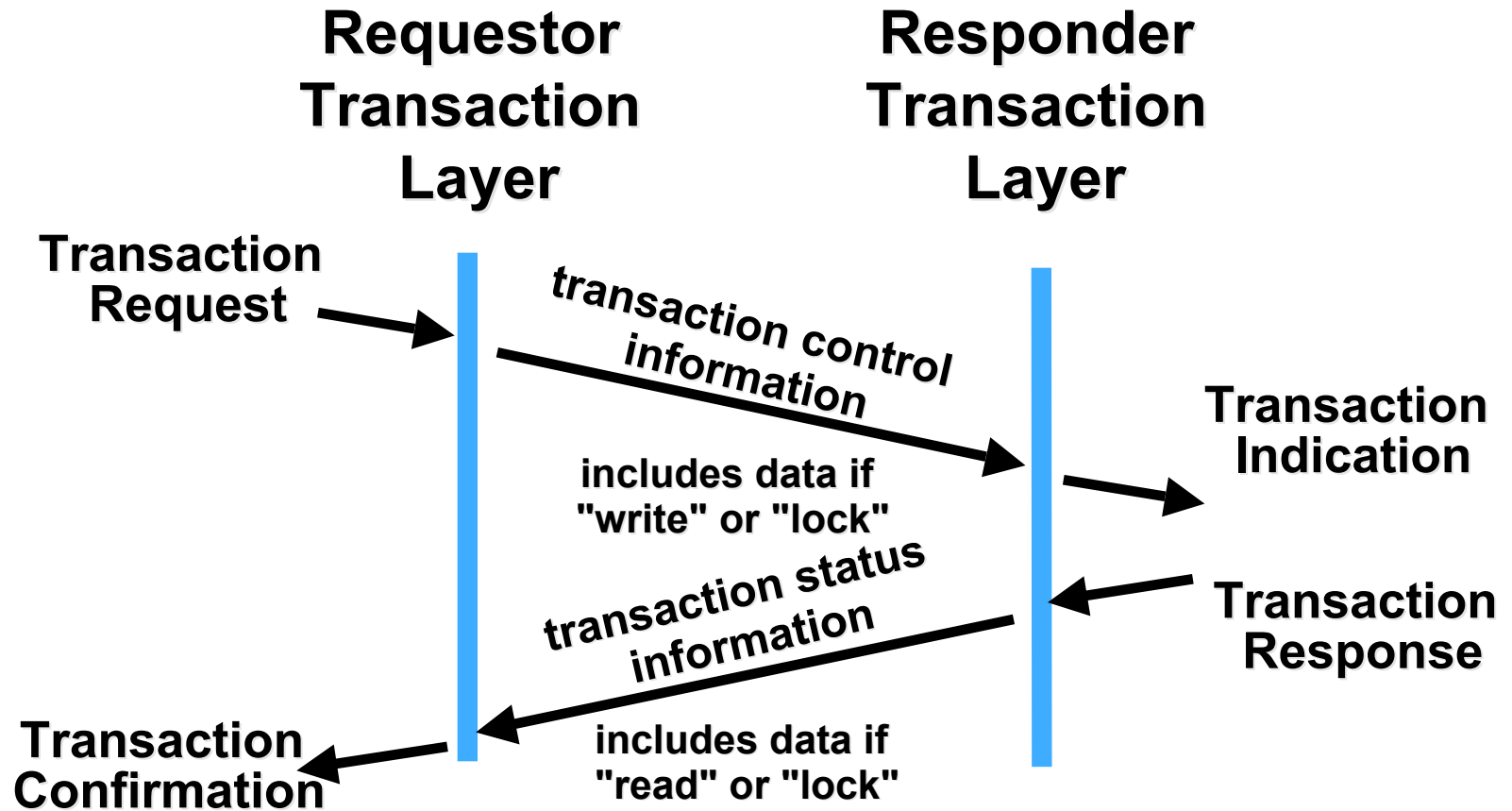


- Cycle start phase has NO correlation with the cycle number
 - Odd/even just used for request pipelining

Attributes of 1394 isochronous service

- Instantaneous jitter for packet delivery is about 200 μ s worst case
 - Most applications provide about 250 μ sec buffers
- Long term drift is determined by cycle master
 - 100 ppm clock accuracy required
 - Better specifications coming from digital studio studies

Transaction layer



Multiple transaction types

- Simplified 4-byte (quadlet) read and write are required
- Variable-length block read and write are optional
- Lock transactions optional
 - Swap, Compare-and-swap needed for bus management

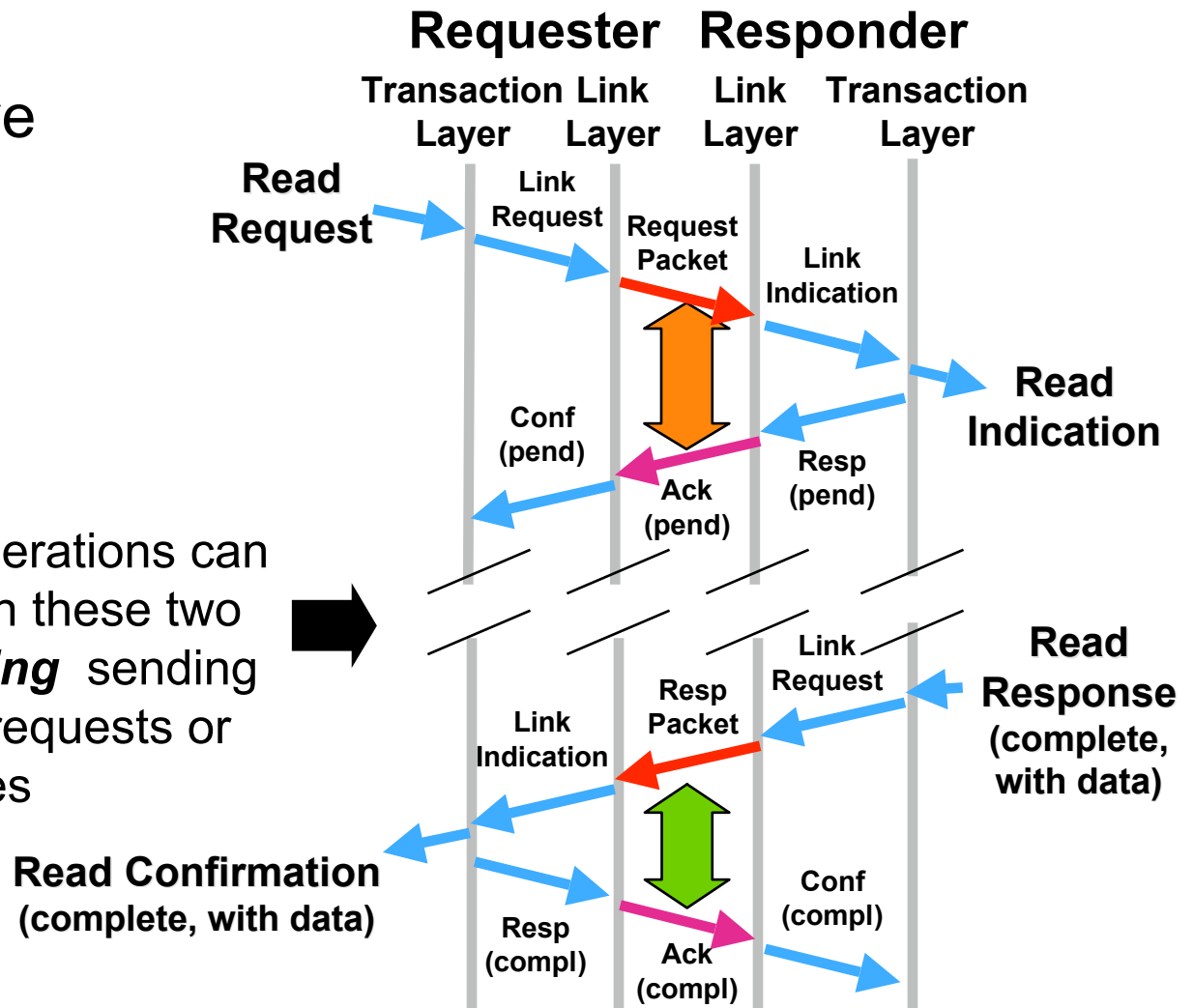
Efficient media usage

- Split transactions required
 - Transactions have request and response parts
 - Bus is never busy unless data is actually being transferred
- Request and response can be unified two ways
 - "Read" and "Lock" can have concatenated subactions
 - "Write" can have immediate completion

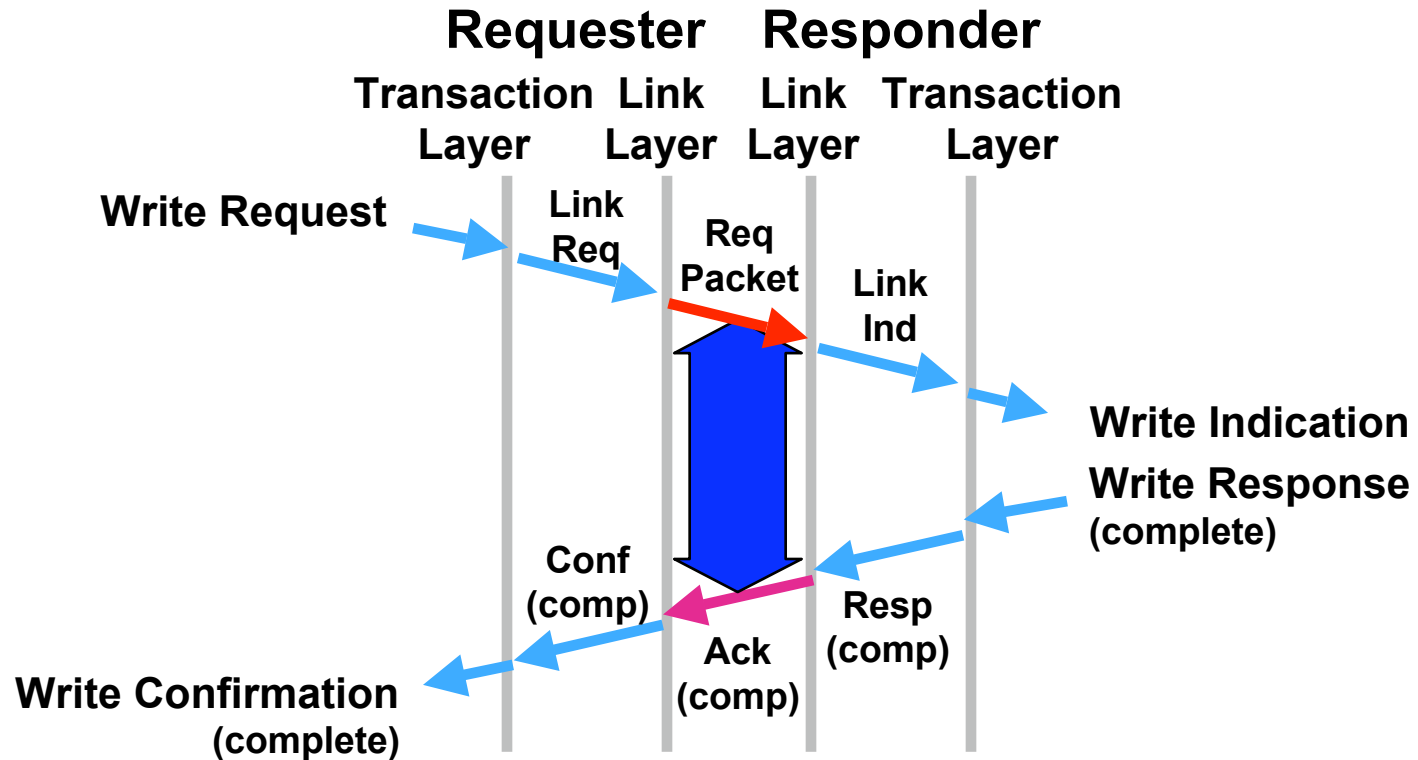
Split transaction

- Request and response have separate subactions

Other Link-Layer operations can take place between these two subactions, *including* sending other transaction requests or responses



Unified transaction



- Only used for write transactions

Bus management

- Automatic address assignment
 - Done in physical layer with self-ID process
 - Root (cycle master) is “sticky” between bus resets
- Resource management
 - Isochronous channels and bandwidth (also “sticky” ... stay allocated between bus resets).
 - Power
- Standardized addresses and configuration ROM from IEEE 1212 architecture

Resource management

- Done with 4 registers, each with compare-swap capability
 - Bus manager ID
 - holds 6-bit physical ID of current bus manager
 - Bandwidth available
 - holds 13-bit count of time available for isochronous transmission
 - Channels available
 - two 32-bit registers with a bit for each of the 64 possible isochronous channels

Compare-swap operation:

- request has “new data” and “compare” values
- responder compares current value (“old data”) at requested address with “compare” value
- if equal, the data at the address is replaced with “new data” value
- in all cases, “old data” is returned to requester

Using compare-swap

- Example: allocate bandwidth

```
test_bw = read4 (addr = bandwidth_available);
old_bw = test_bw + 1; // force entry into loop 1st time
while (old_bw != test_bw) {
    old_bw = test_bw;
    new_bw = old_bw - bandwidth_needed;
    if (new_bw < 0) fail; // all out of bandwidth
    test_bw = compare_swap (addr = bandwidth_available,
        new_data = new_bw, compare = old_bw); }
```

- *test_bw* will be equal to *old_bw* if no other node has altered the *bandwidth_available* register between the time it was read and the time of the *compare_swap*

Where are the bus resource registers?

- On bus reset PHY builds network, assigns addresses, sends self-ID packets
 - power requirements/capabilities, maximum speed rating, port status (child, parent, unconnected)
 - “contender” or not
 - link (higher layers) running or not
- Highest numbered node with both contender and link-on bit is “isochronous resource manager”
 - this is the node that has the four resource manager registers

Automatic reallocation & recovery of resources

- When self_ID completes:
 - all nodes with allocated bandwidth and channels before bus reset reallocate their resources
- after one second:
 - nodes with new bandwidth or channel request may ask for new resources
 - nodes keep resources they had before bus reset!
 - resources allocated to nodes removed from bus are automatically restored!
- Bus manager reallocated the same way

Automatic restart of isochronous operation

- Root assignment is persistent across bus reset
 - Cycle master operation restarts after bus reset if node is still root (normal case)
- Nodes assume that bandwidth and channel allocations are still good
 - Automatically restart sending when receive cycle start
- Only fails if two operating subnets are joined
 - If reallocation fails, node terminates sending
 - If bus over allocated, cycle master detects isoch data sent for longer than 100 μ sec and stops sending cycle starts

How does 1394 help?

- Much better human interface
 - smaller, more rugged connectors with defined usage
 - Hot plugging, no manual configuration
- Excellent real performance
 - High true data rates
 - Direct map to processor I/O model
 - DMA is simple: CPU memory directly available to peripherals
 - example: SBP supports direct scatter/gather buffers

... but even more important

- *Direct support for isochronous data*
 - *THE* current choice for digital consumer video, high-end audio

Getting documentation

- “IEEE 1394-1995 High Performance Serial Bus”, “IEEE 1394a-2000 Amendment 1”, “IEEE 1394b-2002 Amendment 2”
 - IEEE Standards Office +1-908-981-1393,
<http://standards.ieee.org>
- Internet email reflectors
 - “stds-1394@ieee.org”, subscription information at
<http://grouper.ieee.org/groups/1394/c>
- 1394 Trade Association - <http://www.1394ta.org>

Thank You