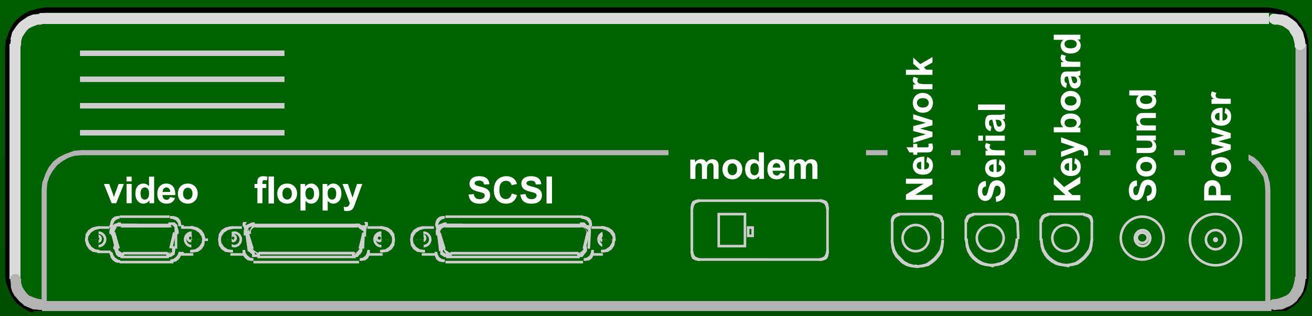
IEEE 1394-1995 High Performance Serial Bus

Michael D. Johas Teener
Chief Technical Officer,
Zayante, Inc.
269 Mt. Herman Rd. #201
Scotts Valley, CA 95066-4000
mike@zayante.com

Background
(the way things are now)



- 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 peripheral bus
- Read/write memory architecture
  - NOT an I/O channel
- Compatible architecture with other IEEE busses
  - Follow IEEE 1212 CSR (Control and Status Register) standard
- Isochronous service

"Isochronous" ??

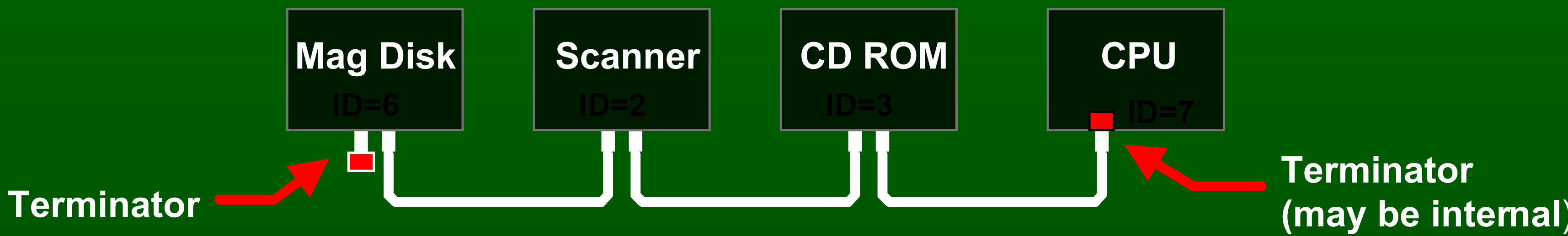
- Iso (same) chronous (time) :
  - Uniform in time
  - Having equal duration
  - Recurring at regular intervals

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

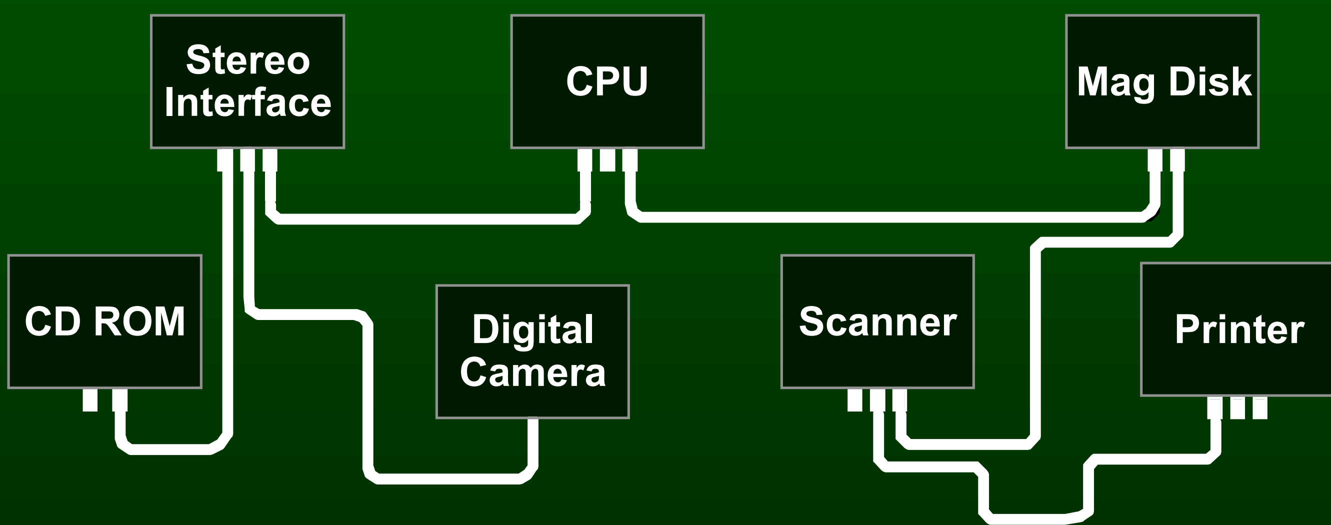
Asynch Vs. Isoch

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

Unsupervised!

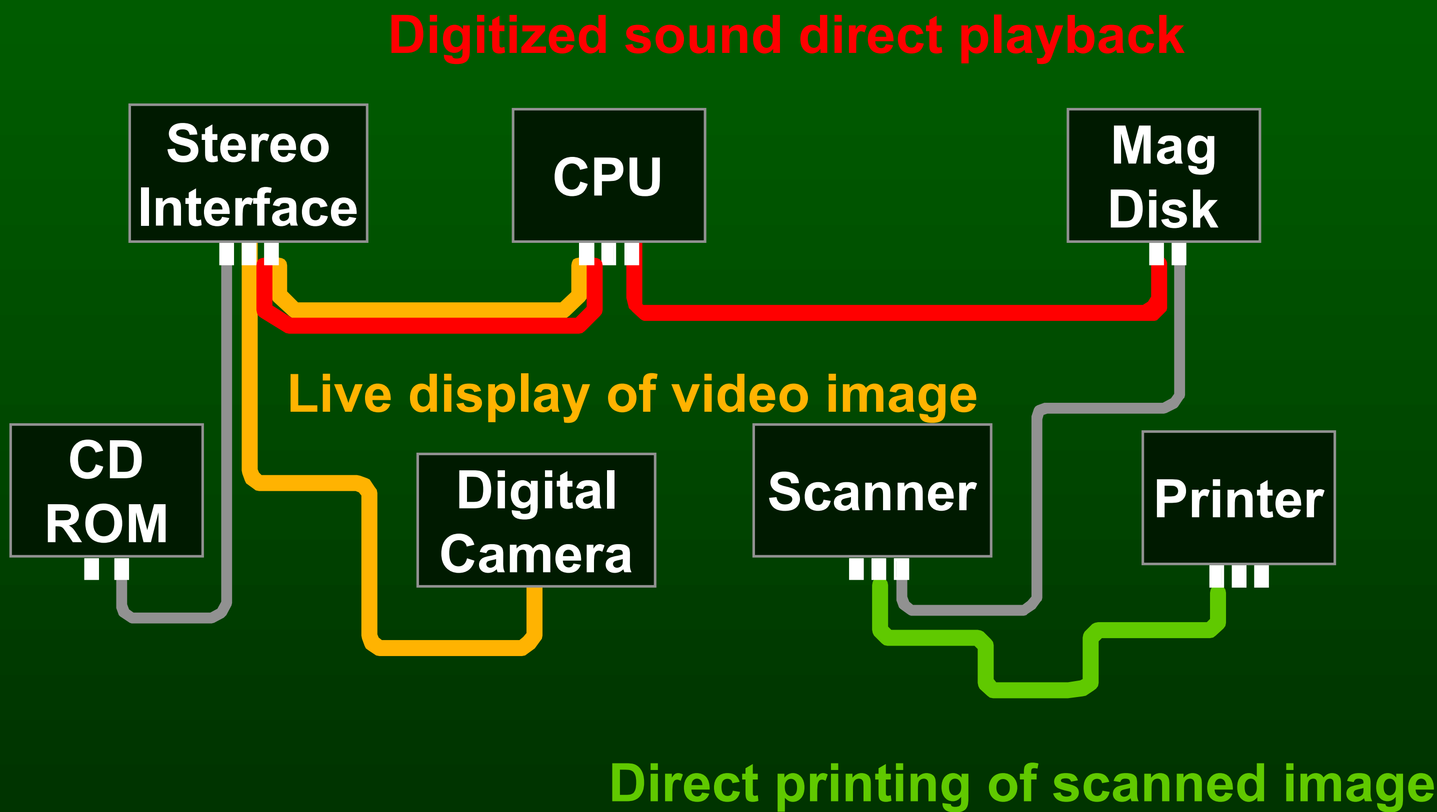


SCSI is typical *"supervised cabling"* — daisy-chain; manual or fixed addresses; terminators at ends; devices with internal terminations must be at one end

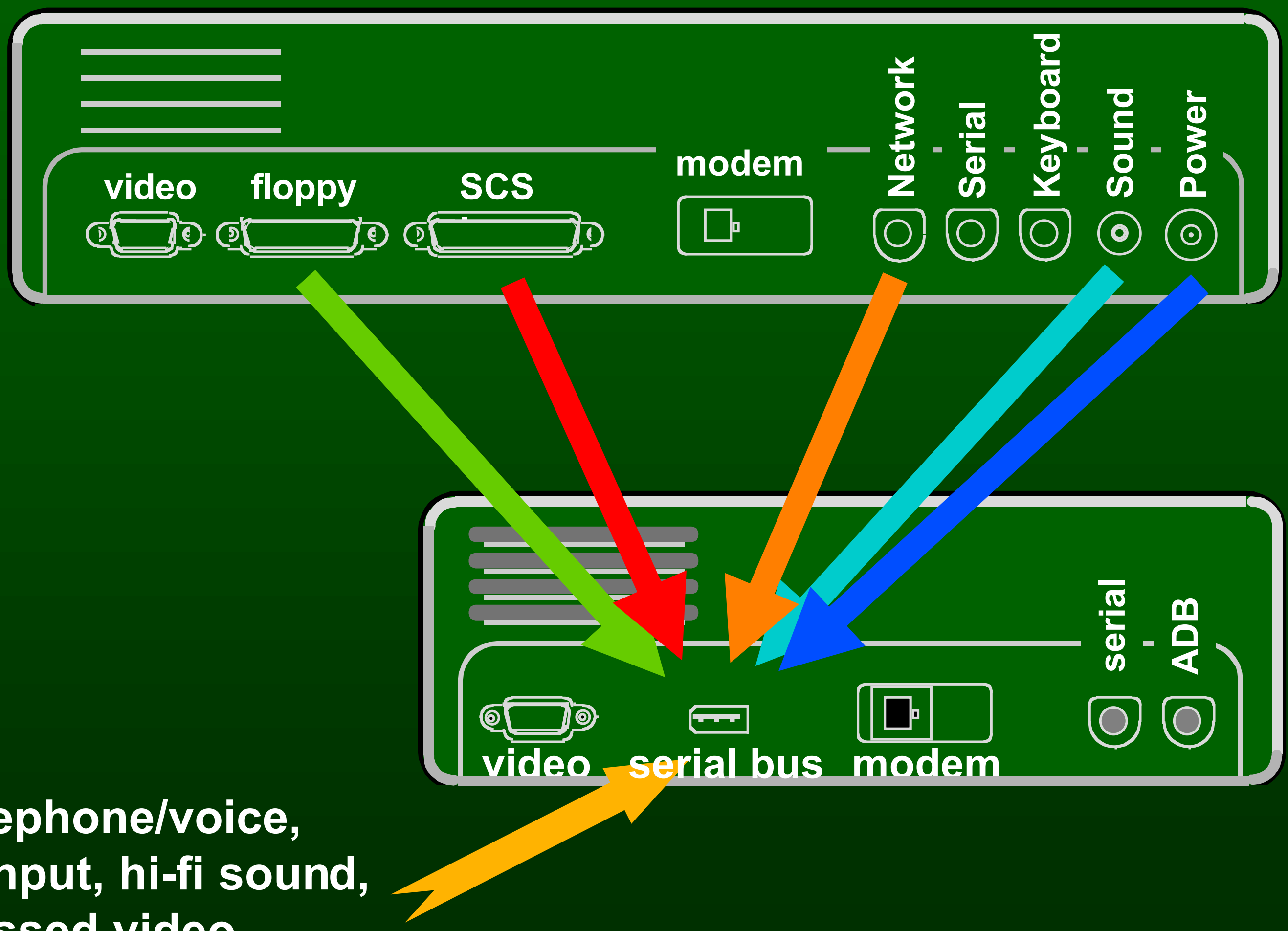


Serial Bus is *"unsupervised cabling"* — "non-cyclic network"; automatic address selection, no terminators, locations are arbitrary

Data paths (peer-to-peer)



Clean up the desktop cable mess!



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

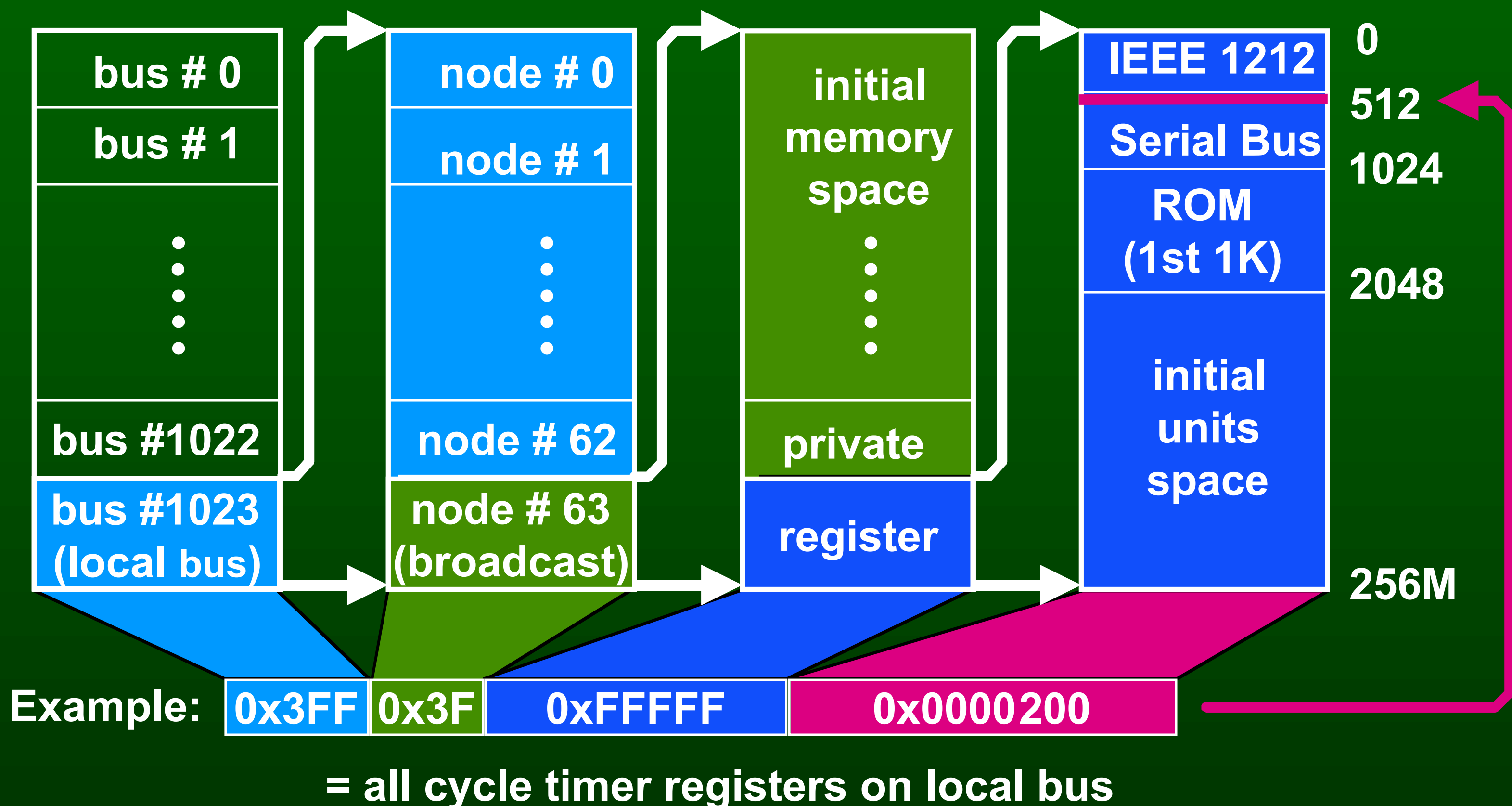
Protocols

- IEEE 1394-1995 High Speed Serial Bus
  - "Memory-bus-like" logical architecture
  - Serial implementation of 1212 architecture
- IEEE 1212-1991 CSR Architecture
  - Standardized addressing
  - Well-defined control and status registers
  - Standardized transactions
- X3T10 Serial Bus Protocol-2 and IEC 61883
  - SBP-2 integrates DMA into I/O process
  - IEC 1883 defines control and data for A/V devices

Some terminology

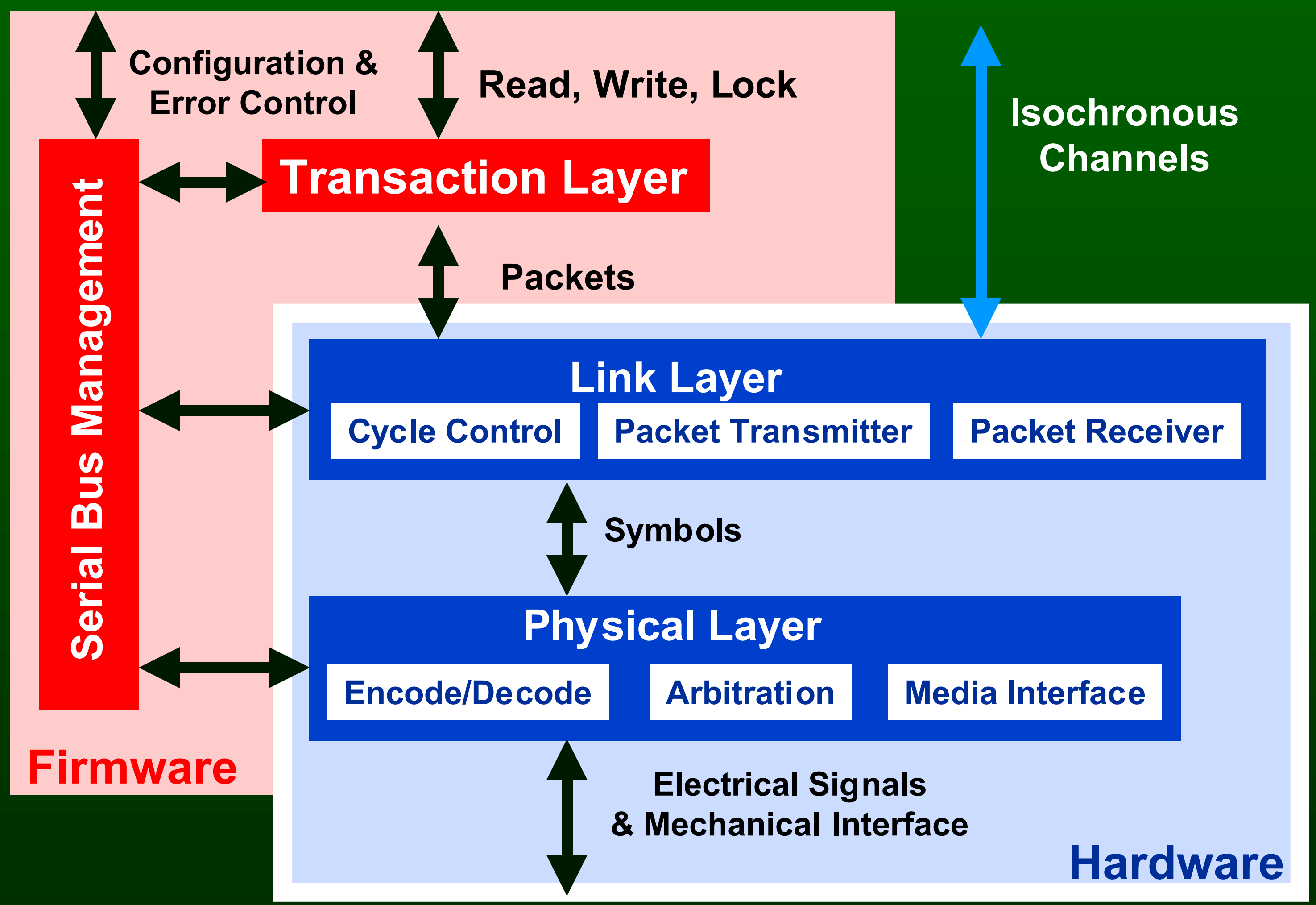
- "quadlet" - 32-bit word
- "node" - basic addressable device
- "unit" - part of a node, defined by a higher level architecture ... examples:
  - SBP disk drive (X3T10 standard)
  - A/V device - VCR, camcorder (IEC 61883 standard)

IEEE 1212 addressing

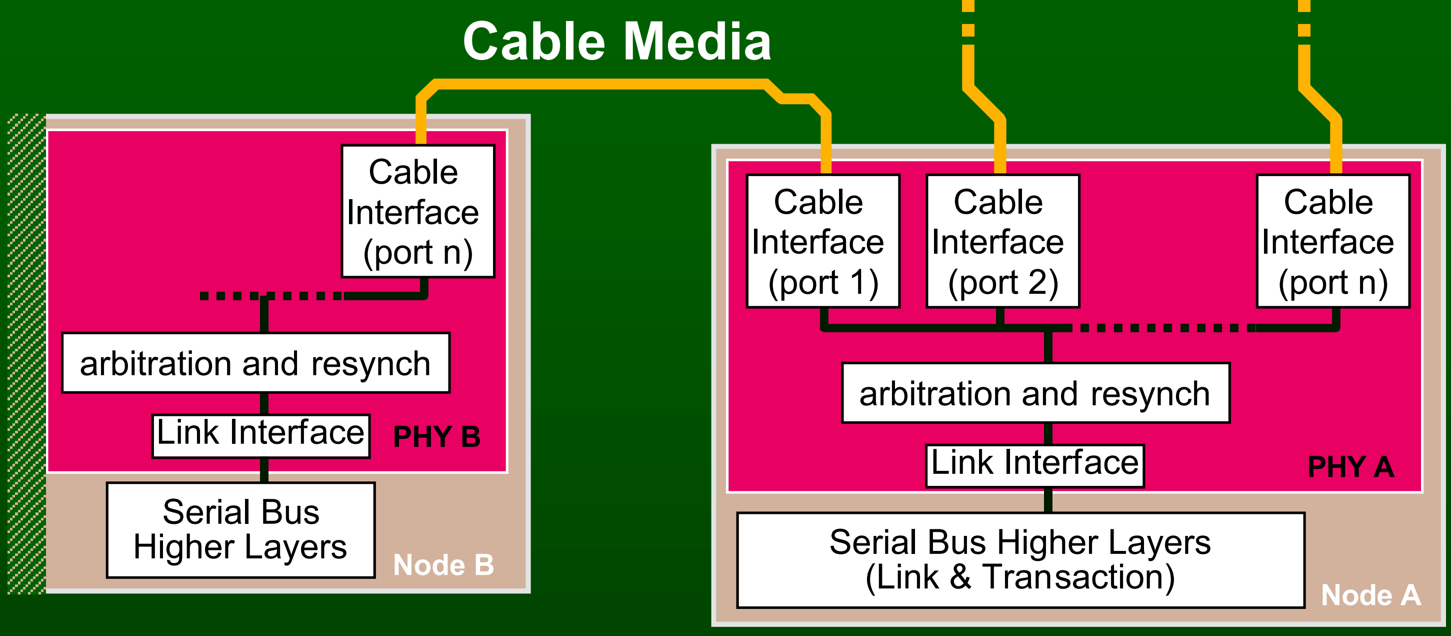


- 1394 uses "64-bit fixed" addressing

IEEE 1394 protocol Stack

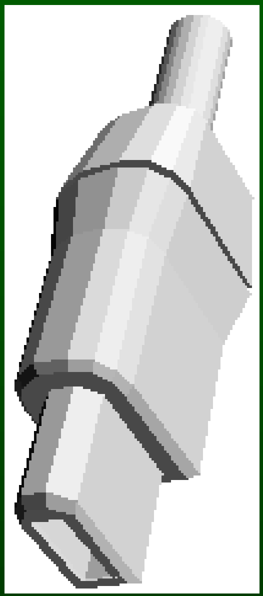


Cable interface



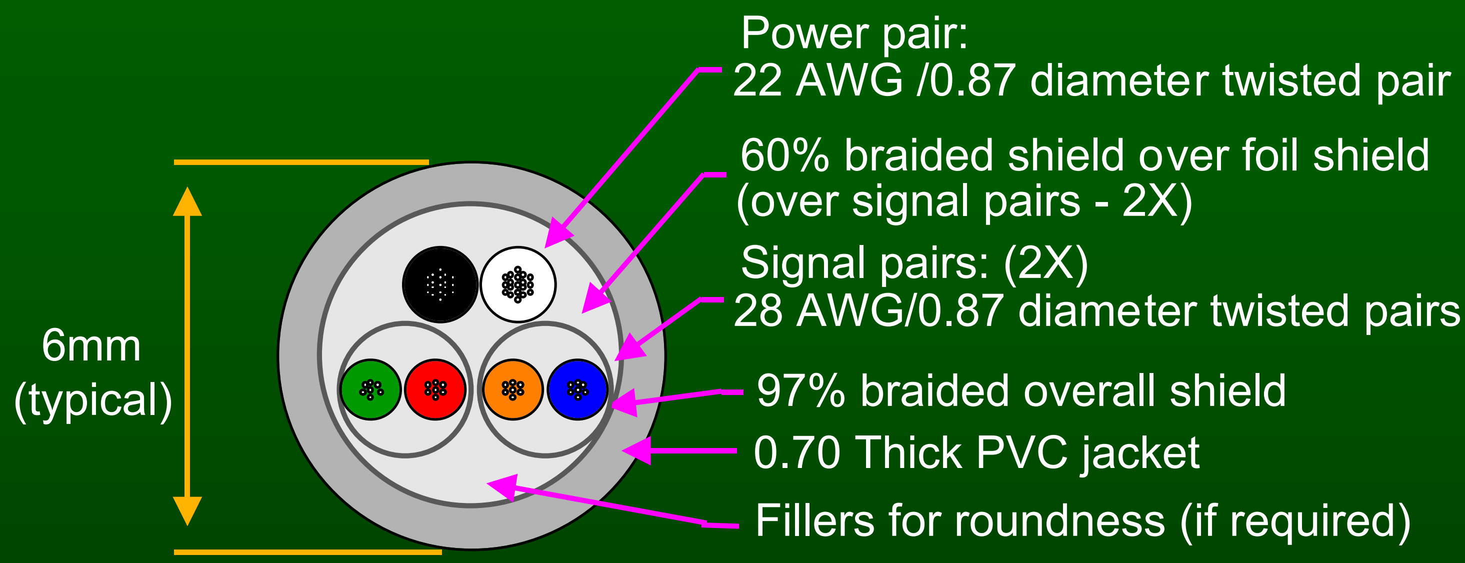
- PHY transforms point-to-point cable links into a logical bus
- Cables and transceivers are bus repeaters

Cable media



- 3-pair shielded cable
  - Two pairs for data transport
  - One pair for peripheral power
- Small and rugged connector
  - Two sockets in the same area as one mini-DIN socket
- CMOS transceiver
  - 220 mV differential
  - 4 ma drive

Cable media example



- Capable of operation at 400 Mbit/sec for 4.5 m
  - Slightly thicker wire allows 10 meter operation
- p1394b encoding allows 800 Mbit/sec for 4.5 m
  - ... perhaps even 1.6 to 3.2 Gbit/sec

Cable interface features

- Live attach/detach
  - System protected from power on/off cycling
  - Higher layers provide simple management

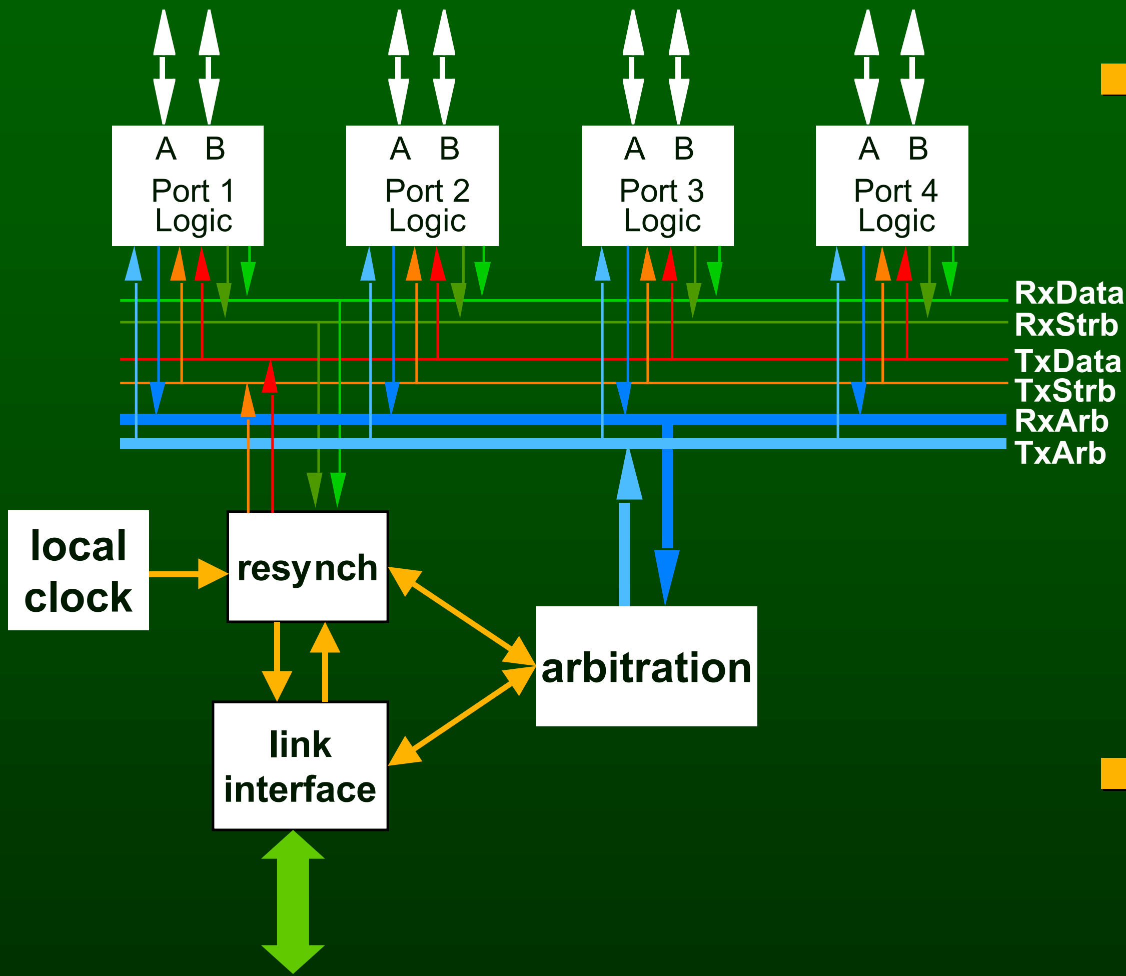
Peripheral power

- 8-40 VDC carried by cable
  - 1394 TA defining tighter standards
  - 20-30 VDC recommended for power sources
- Total available power is system dependent
  - Node power requirements must be declared in configuration ROM
- Cable system allows up to 1.5 A per link
  - Nodes can either source or sink power
  - Multiple power sources on one bus provide additional flexibility

Physical layer

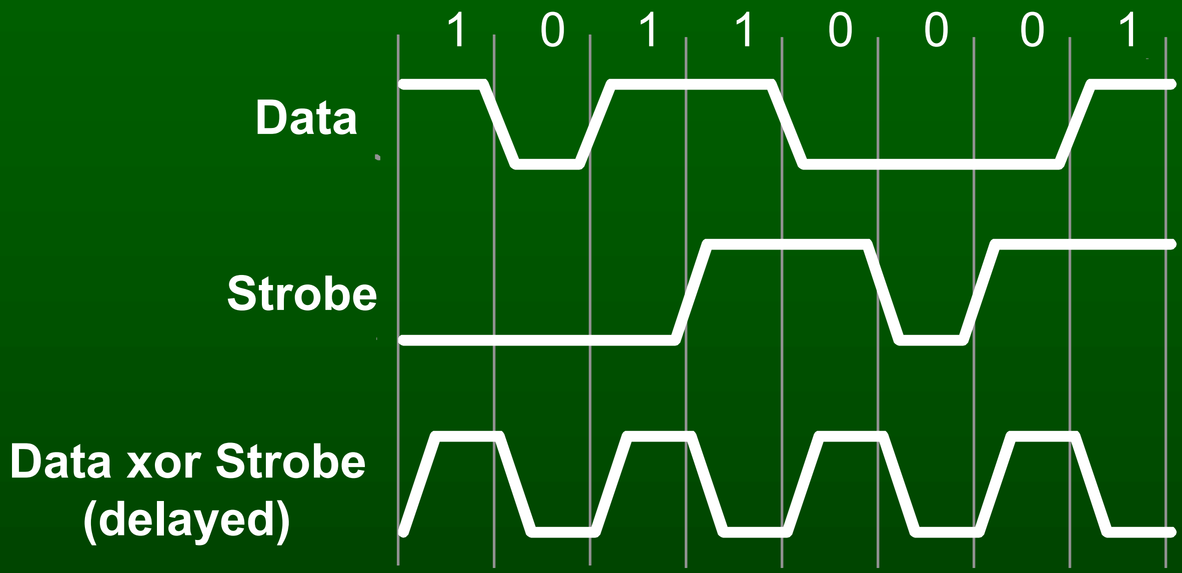
- 98.304 Mbit/sec half duplex transport
  - Data reclocked at each node
  - 196.608 (2x), 393.216 (4x) Mbit/sec growth paths
    - 1394b provides 8X, 16x, 32x rates
- Data encoding
  - Data and strobe on separate pairs
    - 1394b uses 8b10b encoding full duplex
  - Automatic speed detection
- Fair and priority access
  - Tree-based handshake arbitration
  - Automatic assignment of addresses

Example cable PHY IC



- Two twisted pairs for data: TPA and TPB
  - TPA is transmit strobe, receive data
  - TPB is receive strobe, transmit data
  - Both are bidirectional signals, both are used in arbitration
- Reclocks repeated packet data signals using local clock

Data-strobe encoding

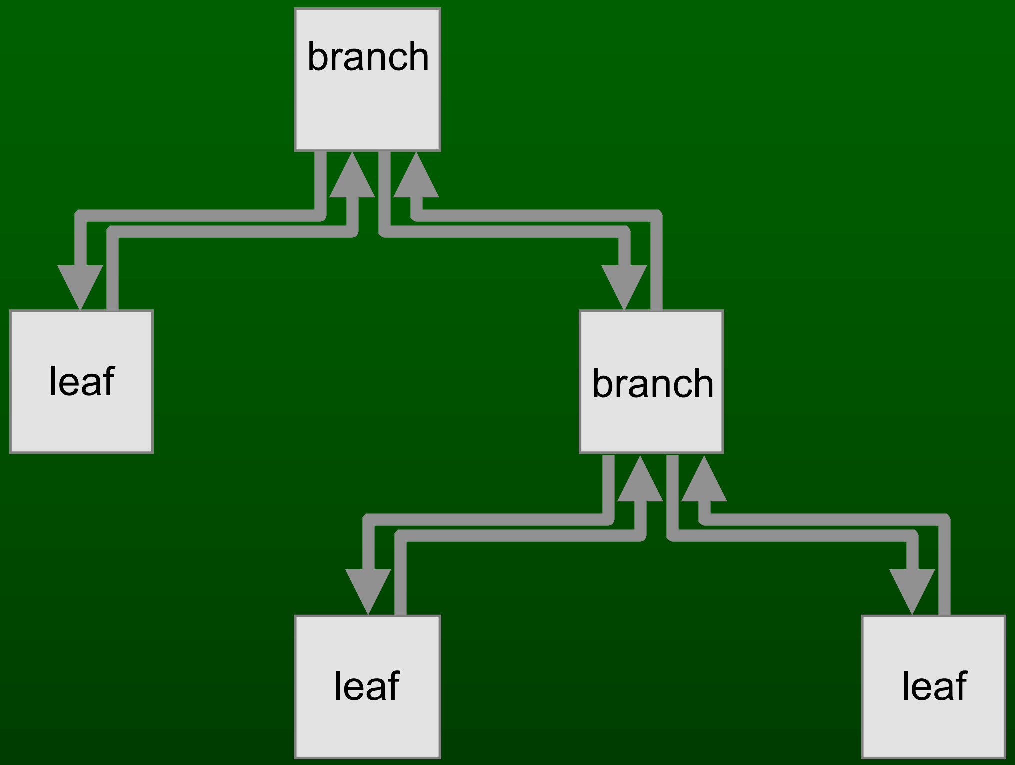


- Either Data or Strobe signal changes in a bit cell, not both
  - Gives 100% better jitter budget than conventional clock/data

Cable arbitration phases

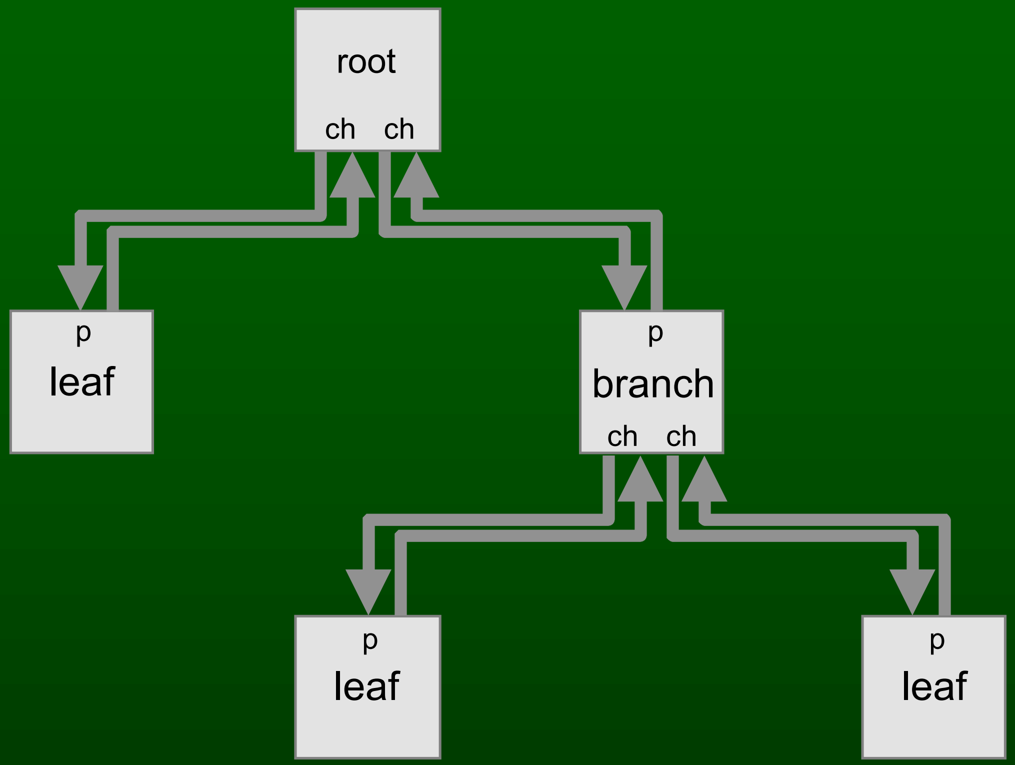
- Reset
  - Used whenever reconfiguration needed
  - Live insertion & new cycle master are examples
- Tree Identification
  - Transforms a simple net topology into a tree
- Self Identification
  - Assigns physical node number (Node ID)
  - Exchange speed capabilities with neighbors
- Normal Arbitration
  - Root has highest priority

Tree identification #1



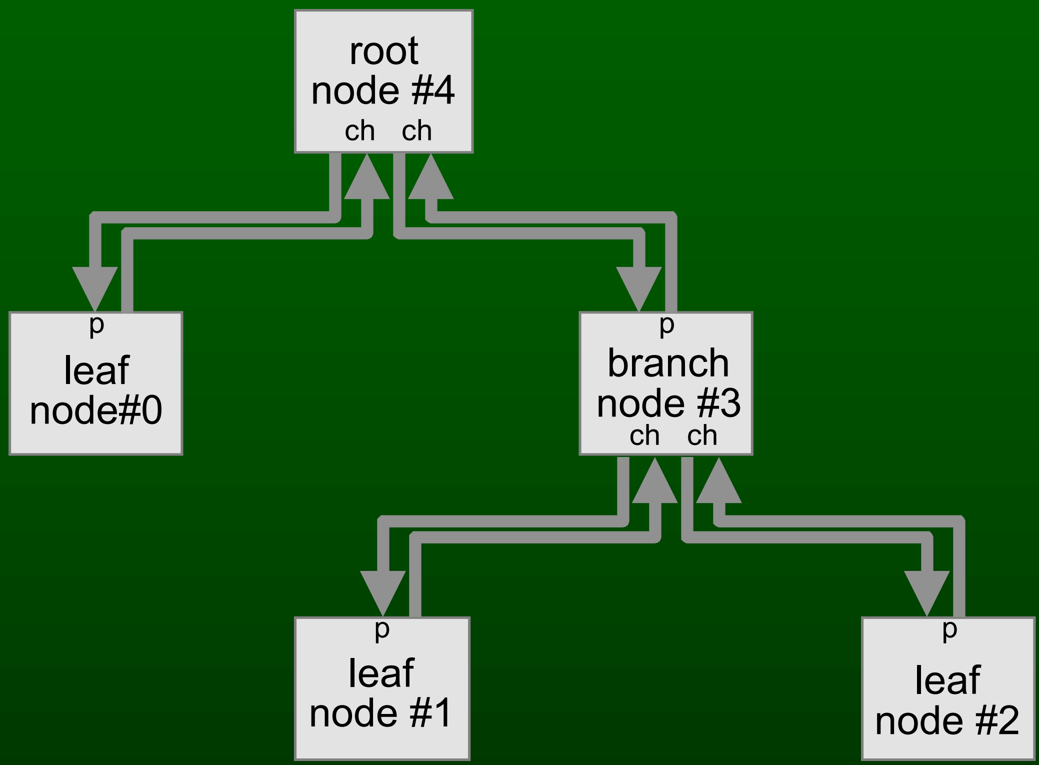
- After reset, each node only knows if it is a leaf (one connected port) or a branch (more than one connected port)

Tree identification #2



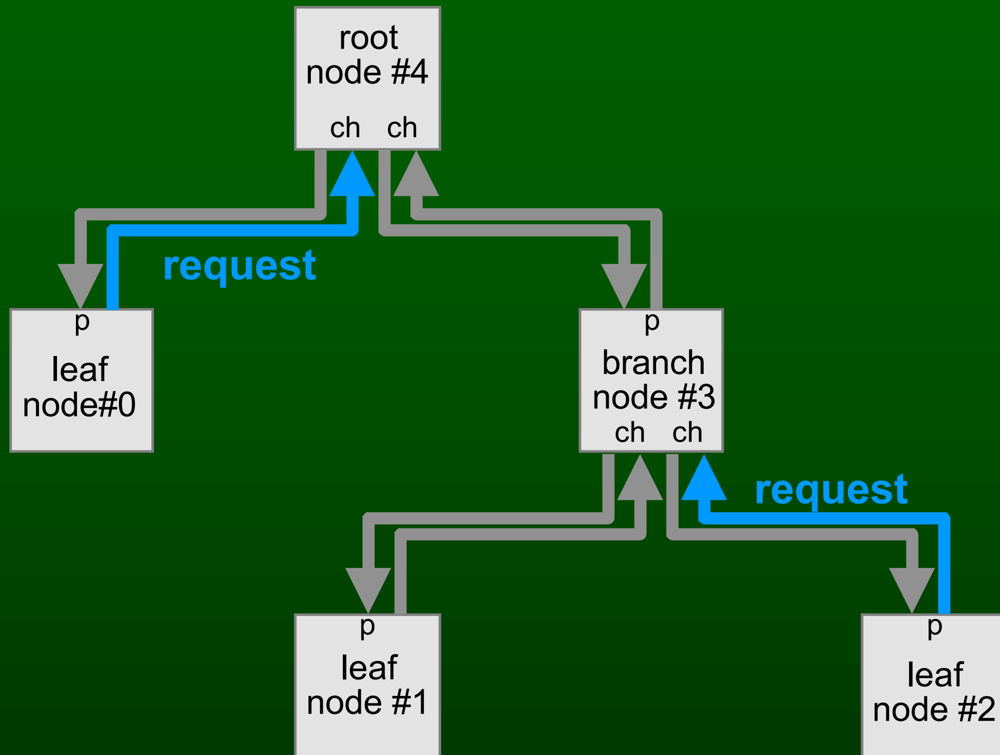
- After Tree ID process, the Root node is determined and each port is labeled as pointing to a child or a parent
  - Root assignment is “sticky”, will normally persist across a bus reset.

Self identification



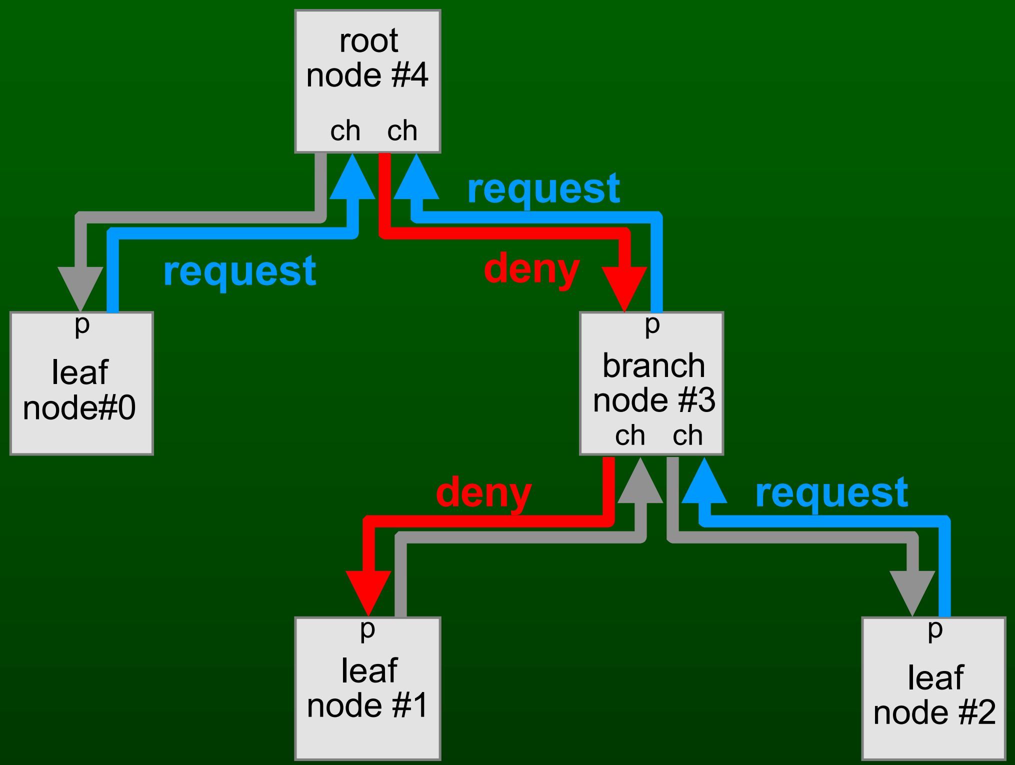
- After the self ID process, each node has a unique physical node number, and the topology has been broadcast

Normal arbitration #1



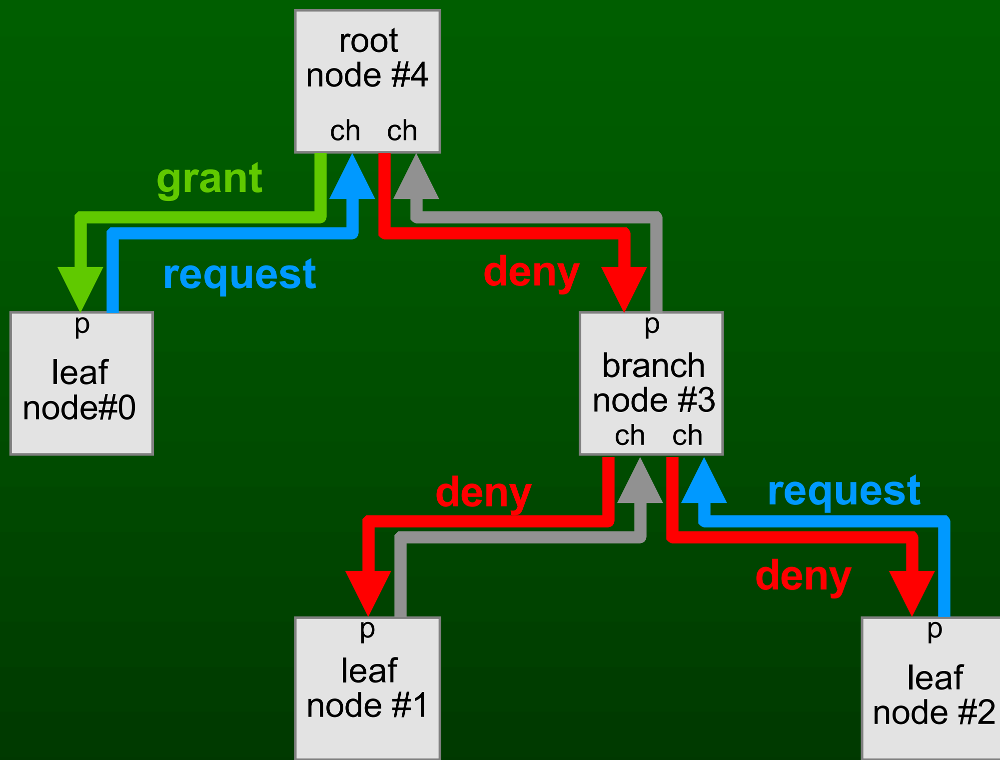
- Suppose nodes #0 and #2 start to arbitrate at the same time, they both send a request to their parent ...

Normal arbitration #2



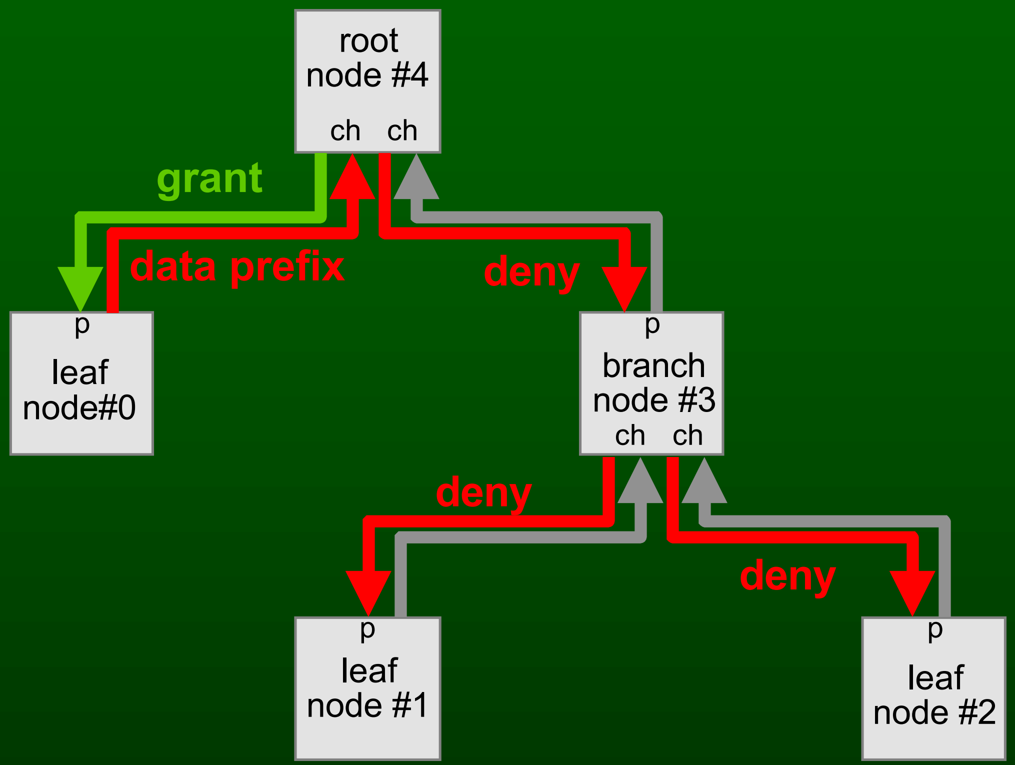
- The parents forward the request to their parent and deny access to their other children ...

Normal arbitration #3



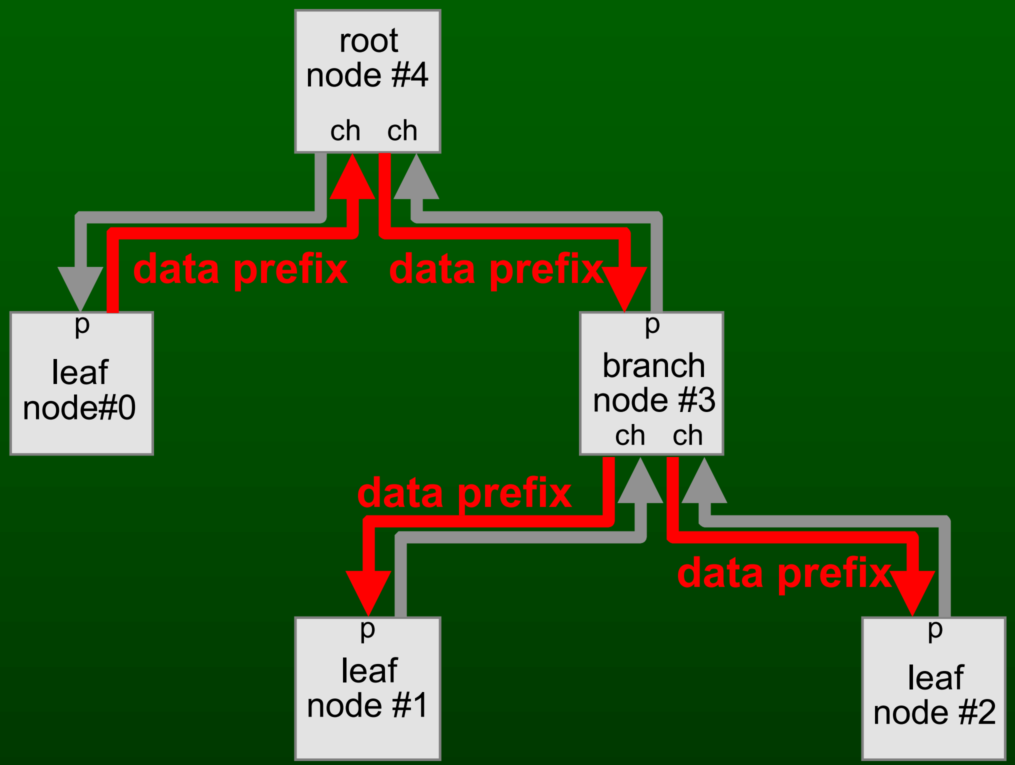
- The root grants access to the first request (#0), and the other parent withdraws it's request and passes on the deny ...

Normal arbitration #4



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

Normal 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!) ...

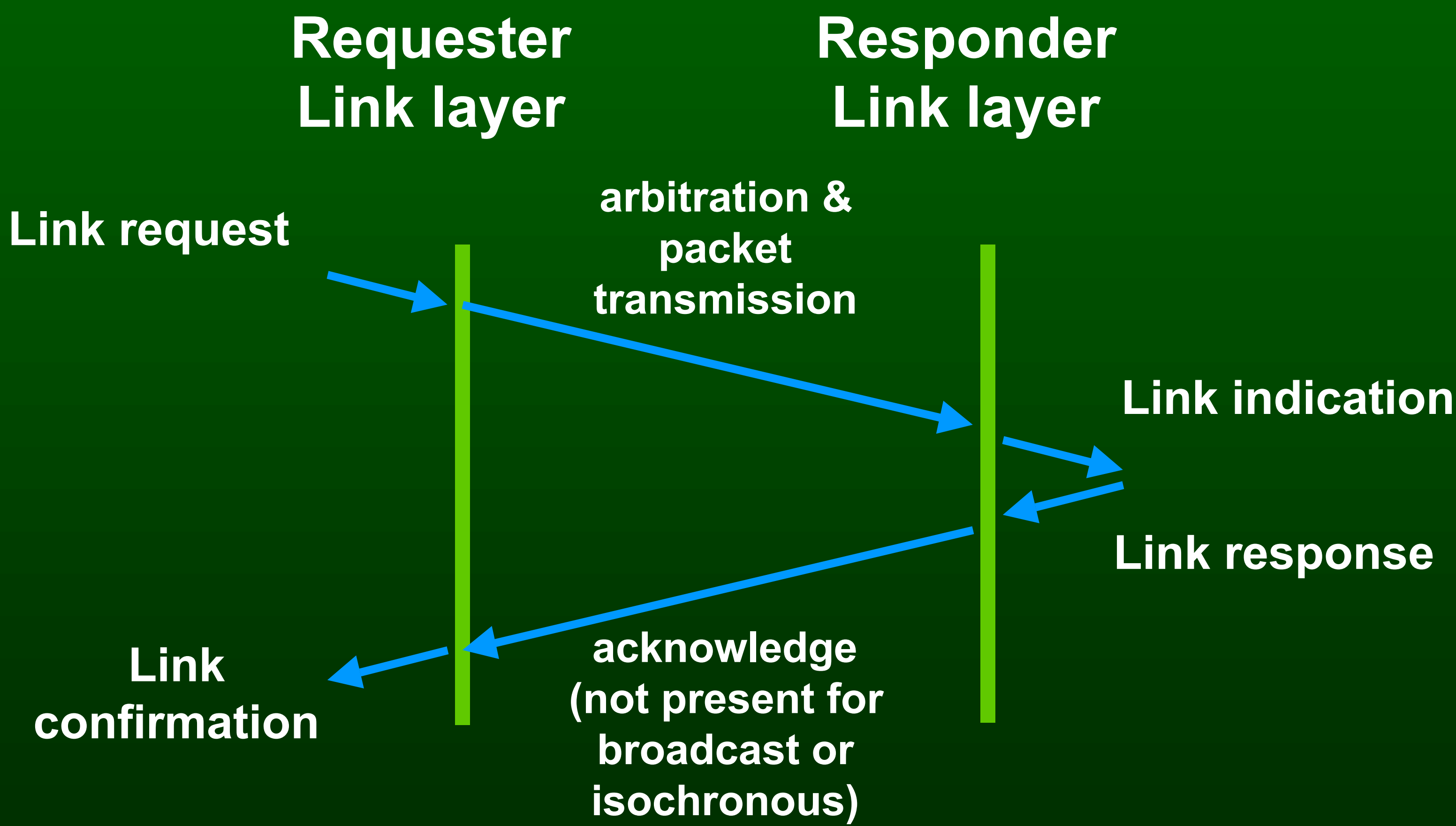
Link layer

- Implements acknowledged datagram service
  - Called a "subaction" of arbitration, packet transmission, and acknowledge
- Flexible addressing using 1212 architecture
  - Direct 64-bit addressing (48 bits per node)
  - Hierarchical addressing for up to 63 nodes on 1023 busses

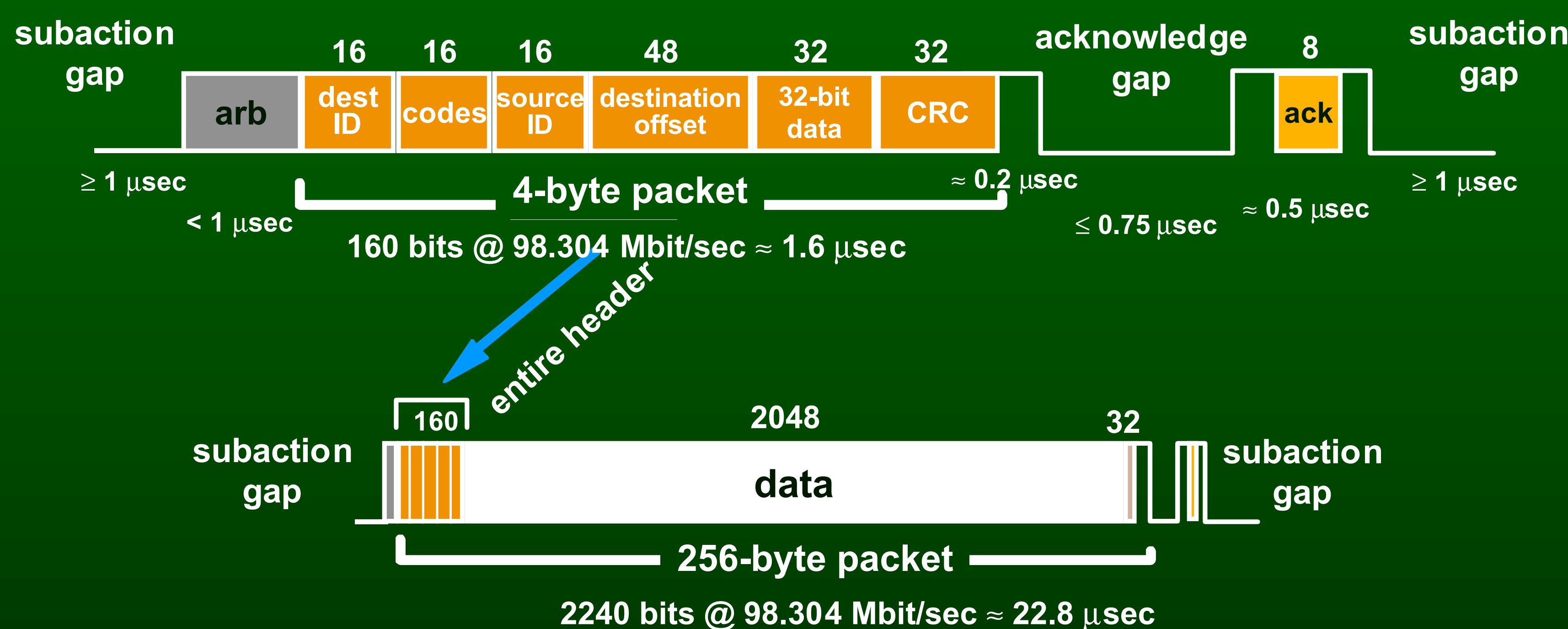
Isochronous transport

- Optional
  - But required for multimedia applications
- Multiple "channels" each 125 µsec "cycle" period
  - Channel count limited by available bandwidth
- Variable channel size up to ≈1000 bytes/cycle
  - Up to ≈2000 bytes/cycle at 196 Mbit/sec

Link layer operation

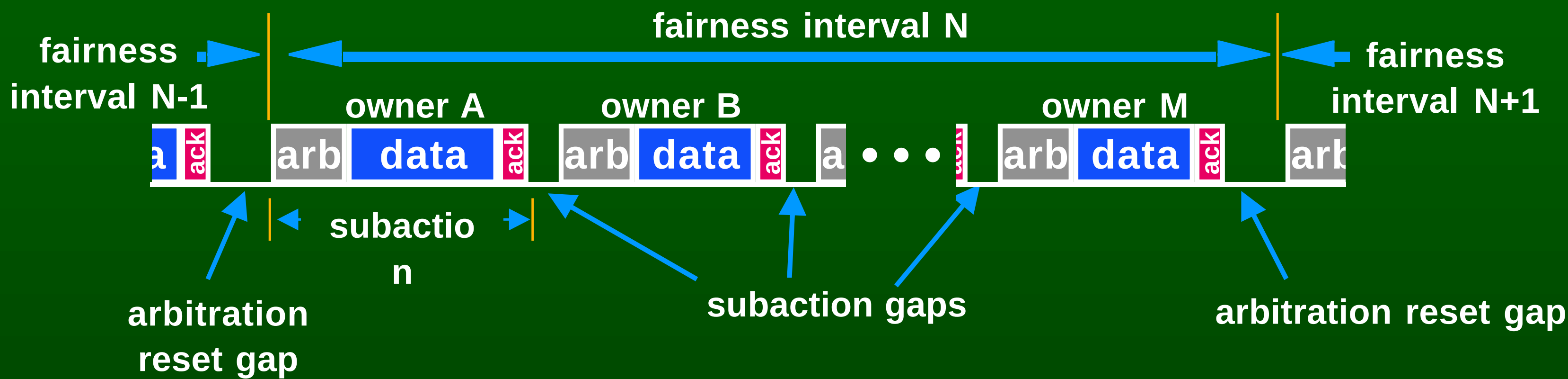


Example packets



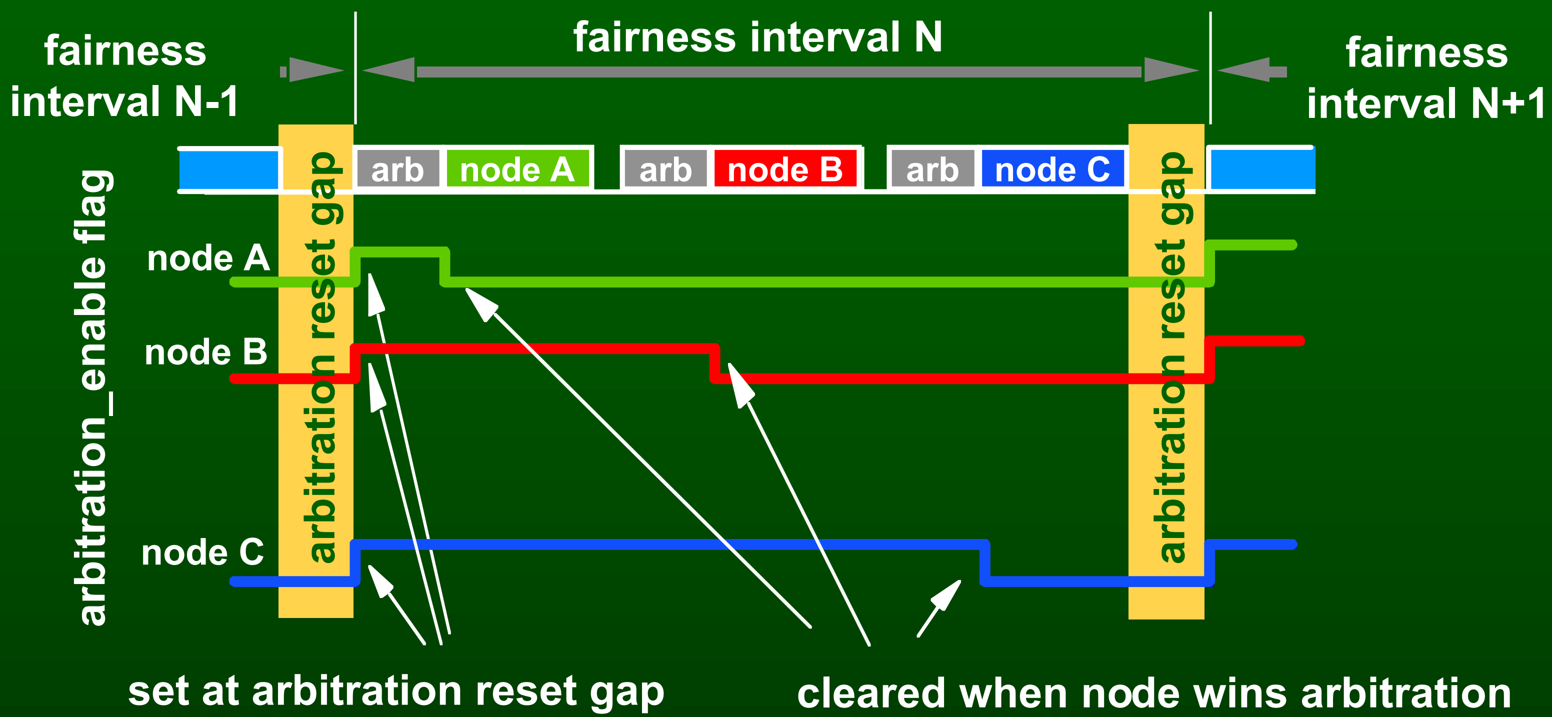
- Actual efficiency very good
  - 10 Mbytes/sec information throughput including all of the SBP disk protocol using 100 Mbit/sec rate (~80%)

Fairness interval



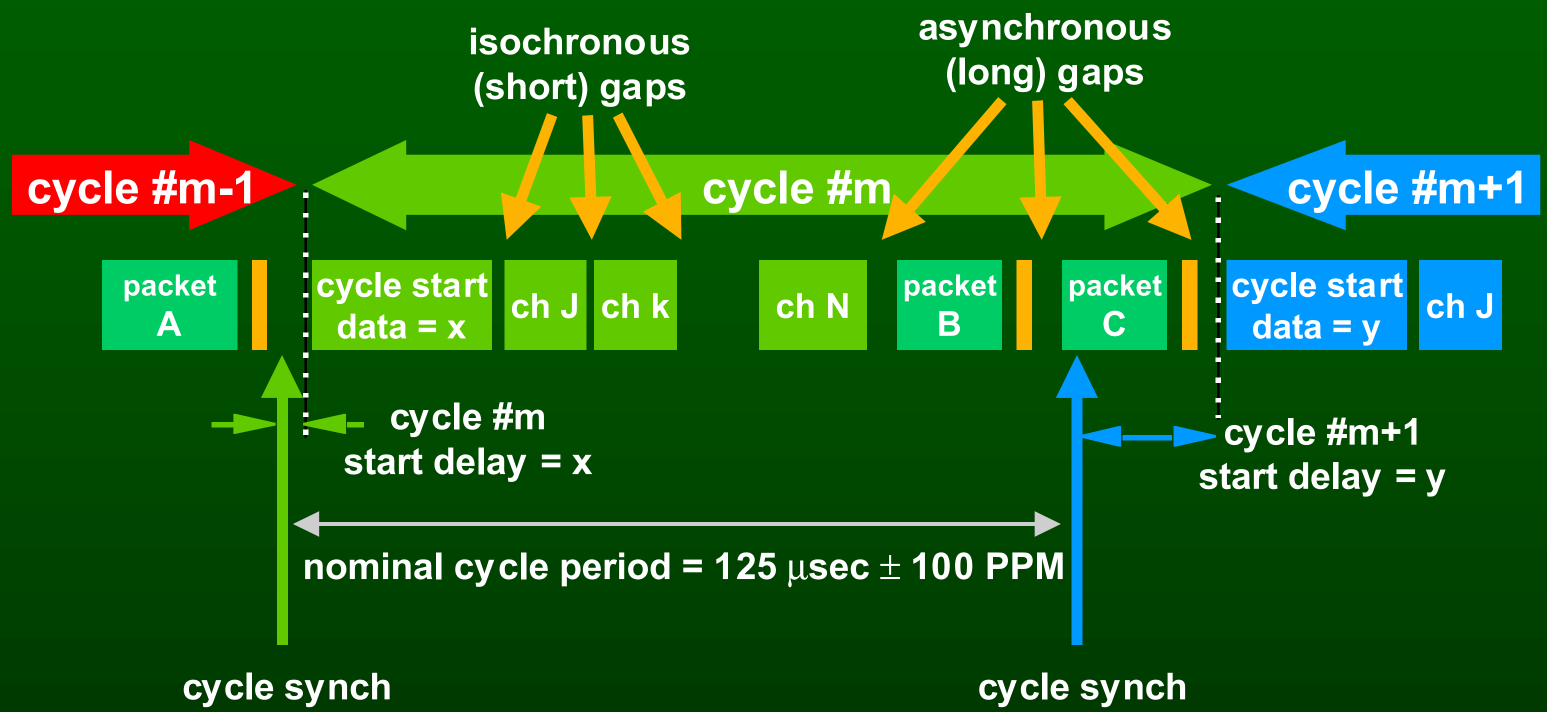
- Fairness Interval is bounded by "arbitration reset gaps"
- Reset gaps are longer than normal subaction gaps

Fair arbitration



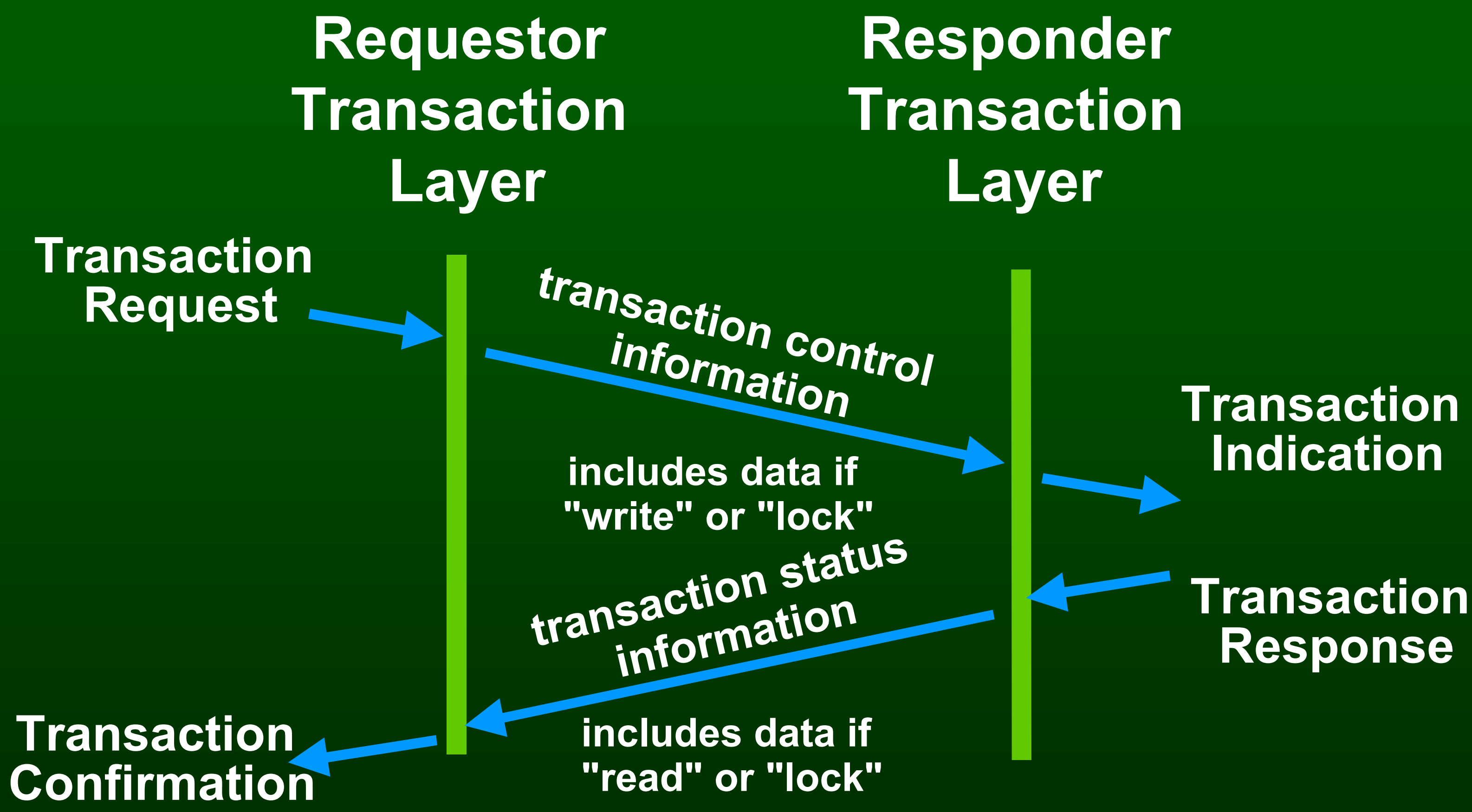
- Each node gets one access opportunity each Fairness Interval
  - special case for isochronous data

Cycle structure



- The cycle start is sent by the cycle master, which must be the root node

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

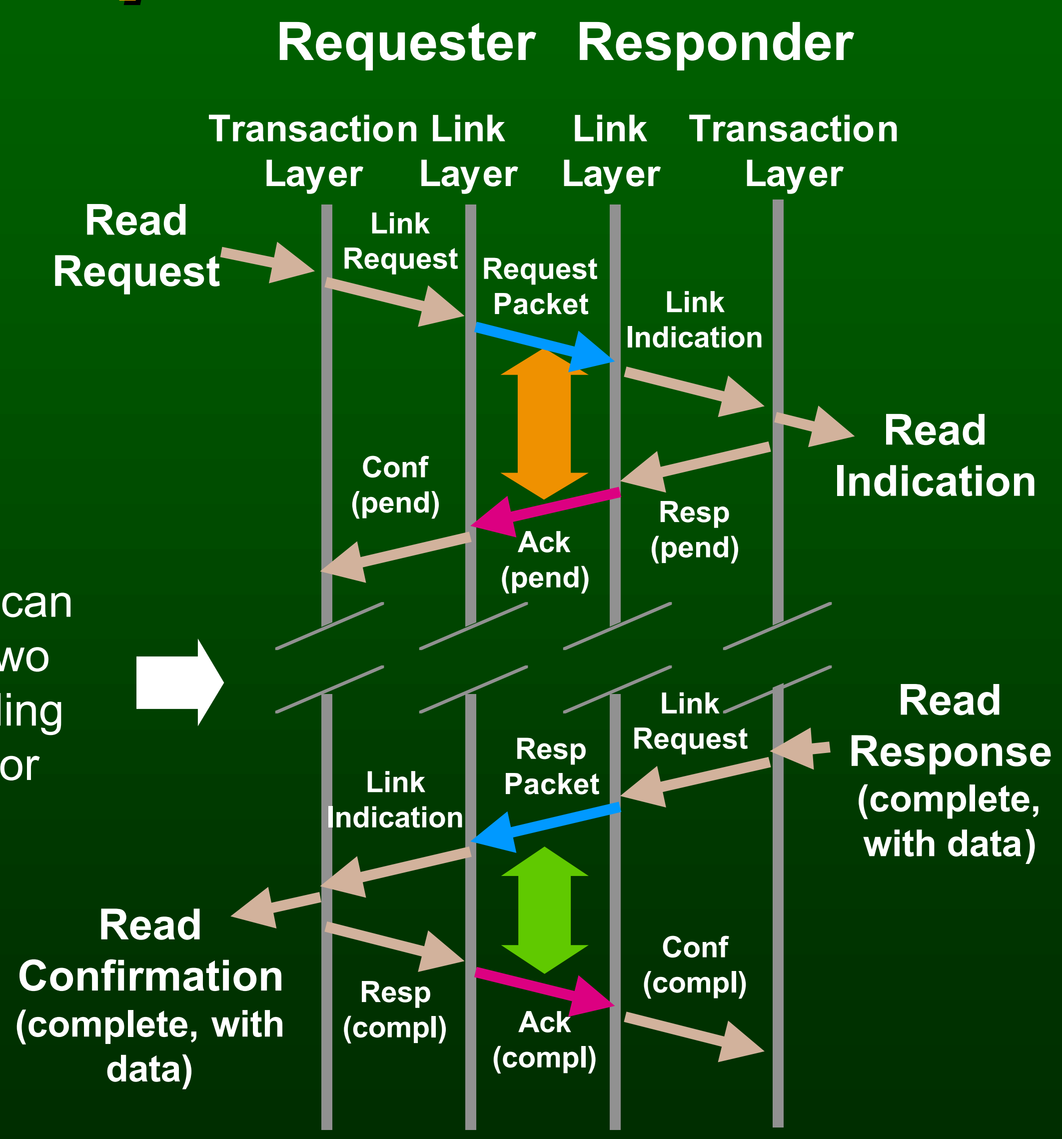
Efficient media usage

- 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

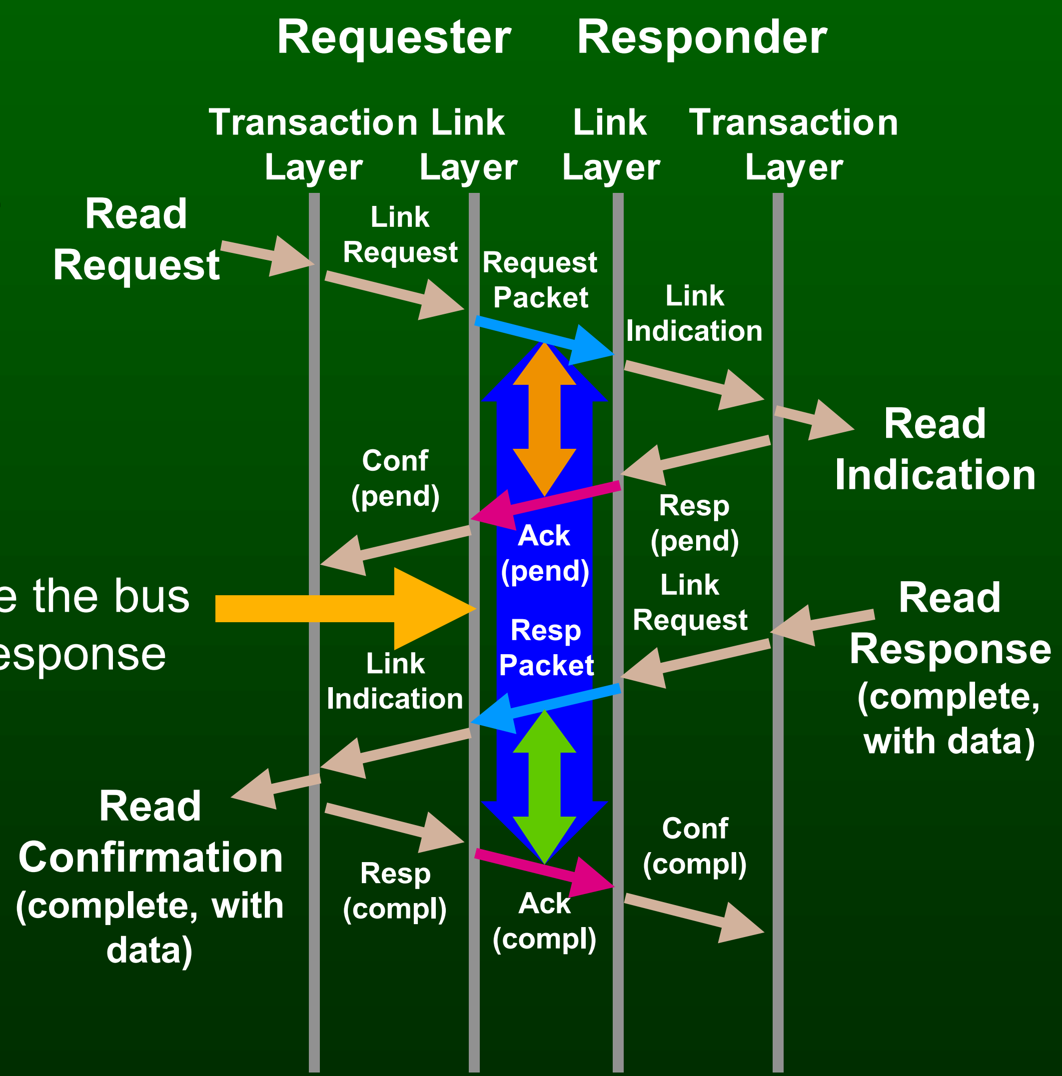
Other Link-Layer operations can take place between these two subactions, ***including*** sending other transaction requests or responses



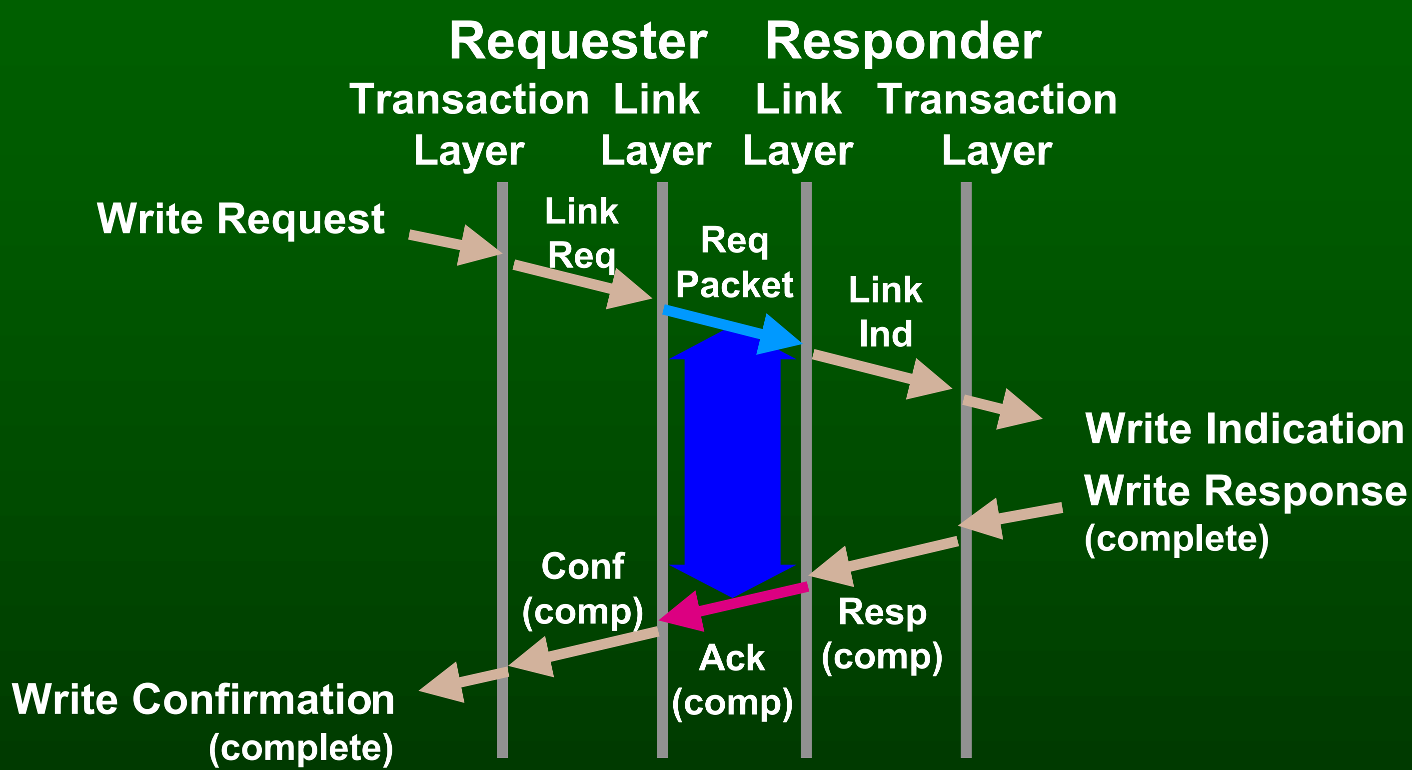
Concatenated transaction

- Used if responder is fast enough to return data before ack is completed

the responder does not release the bus after sending the ack, sends response packet within 1.5µsec



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 µsec and stops sending cycle starts

Futures

- Gigabit rates and fiber (P1394B high speed)
  - 800 Mbit/sec - 3.2 Gbit/sec
- Fast reset (P1394A)
- Support for very low power (P1394A)
  - “suspend-resume”
- Redundant gap removal (P1394A)
  - “Accelerated ACK”, fly-by concatenation
- Bridging issues (P1394.1)
  - for > 63 devices, or for isolation of high-bandwidth local traffic

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

- It's inexpensive
  - For computers, it’s already almost as cheap as single-ended 8-bit SCSI
    - will be cheaper since it's silicon-intensive
  - Much less expensive for peripherals and consumer electronics
- Direct support for isochronous data
  - **THE** choice for digital consumer video, high-end audio
  - Media servers get cheaper

Getting documentation

- "IEEE 1394-1995 High Performance Serial Bus"
  - IEEE Standards Office +1-908-981-1393, http://www.ieee.org
  - P1394a balloting under way (first round closed June 11, 1998), so new version will be available in 1999.
  - P1394.1 and p1394b drafts available via internet, see http://www.zayante.com/p1394b
- Internet email reflectors
  - "`stds-1394@majordomo.ieee.org`" (p1394a) and "`stds-1394-1@majordomo.ieee.org`" (p1394.1) … send "help" to "`majordomo@majordomo.ieee.org`"
  - "`p1394b@zayante.com`", subscription information at http://www.zayante.com/p1394b
- 1394 Trade Association
  - http://www.1394ta.org