

New Technology for 1394: P1394a and P1394b

Michael D. Johas Teener

Chief Technical Officer,

Zayante, Inc.

269 Mt. Herman Rd. #201

Scotts Valley, CA 95066-4000

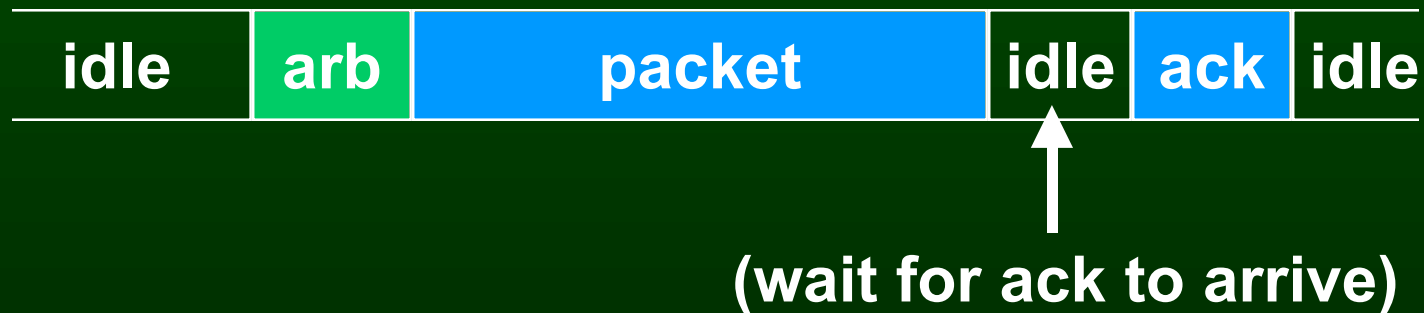
mike@zayante.com

Outline

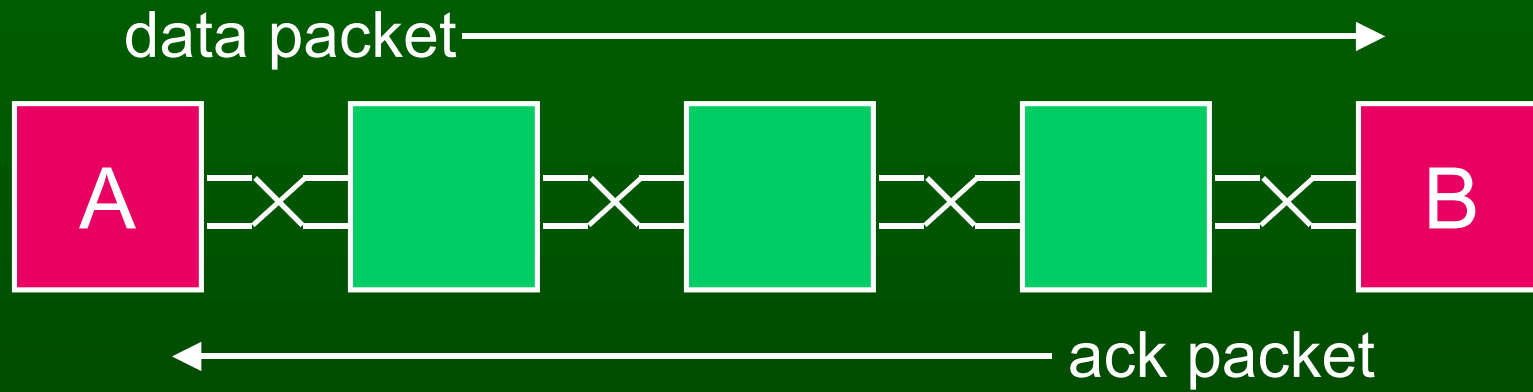
- 1394a: improvements in efficiency
 - ◆ Arbitration acceleration
 - ◆ Reset improvements
 - ◆ Suspend/resume
 - ◆ Miscellaneous
- 1394b: the next step in speed
 - ◆ New connection model
 - ◆ Additional media
 - ◆ Arbitration improvements
 - ◆ Miscellaneous

Background: 1394 link operation

- Wait for idle bus
- Arbitrate
- Send packet
- Responder sends ack



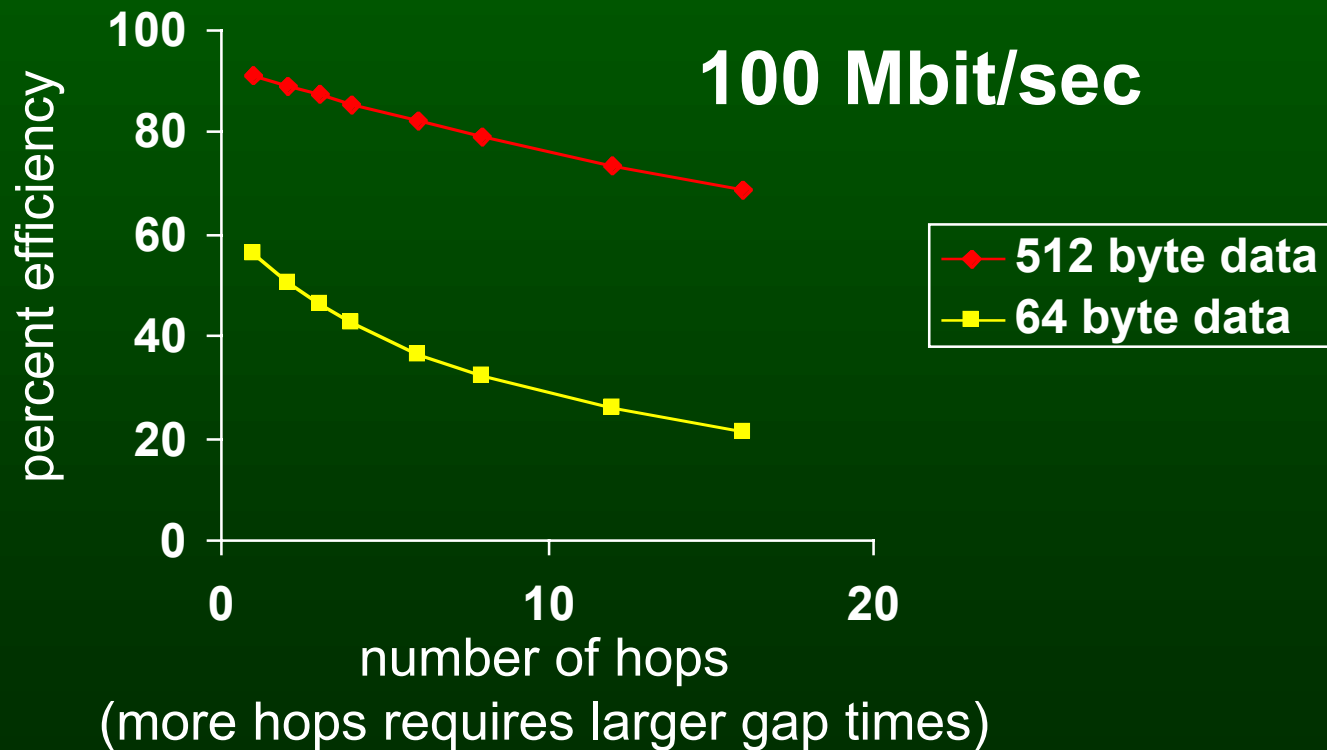
Bus timing for asynch arbitration



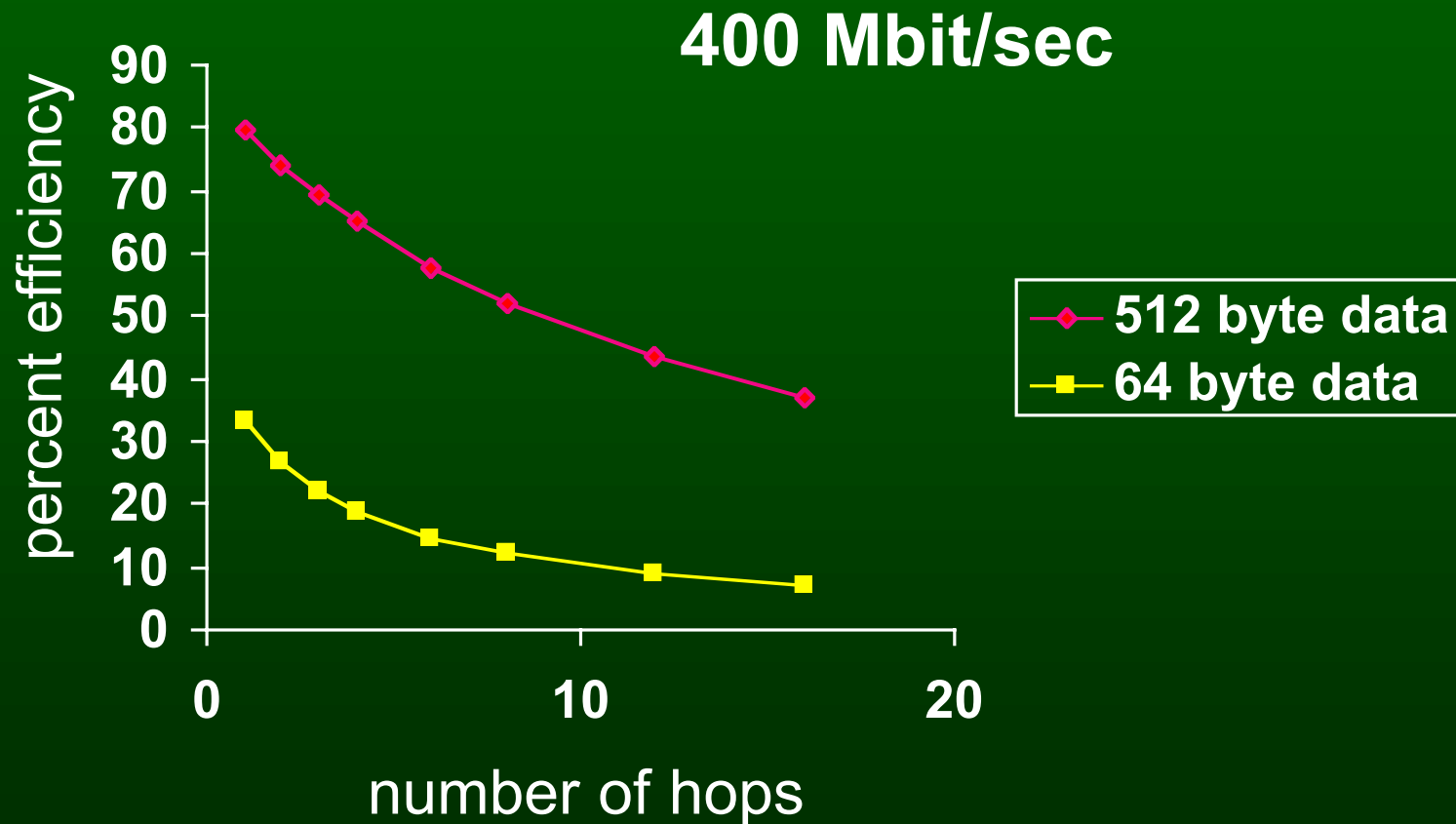
- Subaction gap time is greater than round trip delay time for most distant pair of nodes on a 1394 bus
- For asynch arbitration, waiting a Subaction gap time after passage of last packet before start of new arbitration insures that new arbitration will not interfere with an ack

Efficiency effects of gap time

- OK for large packets, terrible for small ones



Gets worse for high speeds



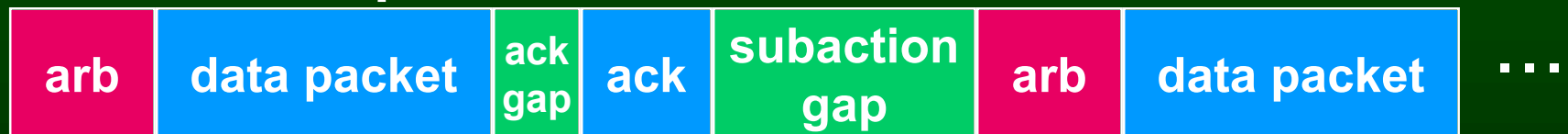
Solutions from P1394a

- ack acceleration
- fly-by arbitration
- priority arbitration

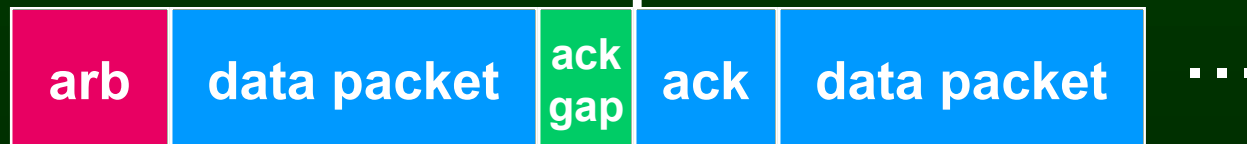
Ack acceleration

- The reason for waiting a Subaction gap before starting arbitration is just to insure that arbitration will not interfere with an ack
- Therefore, if the last packet on the bus was an ack then there will be no subsequent ack, and a new round of arbitration may start immediately

Standard Implementation



Ack Accelerated Implementation

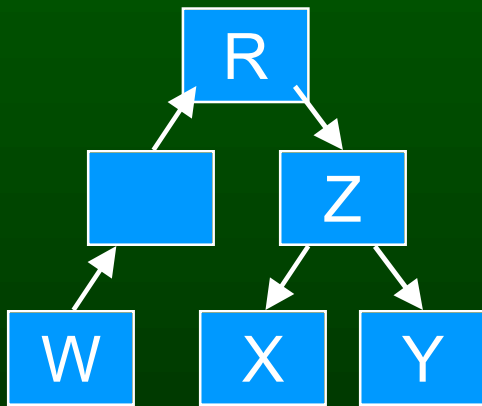


Ack acceleration considerations

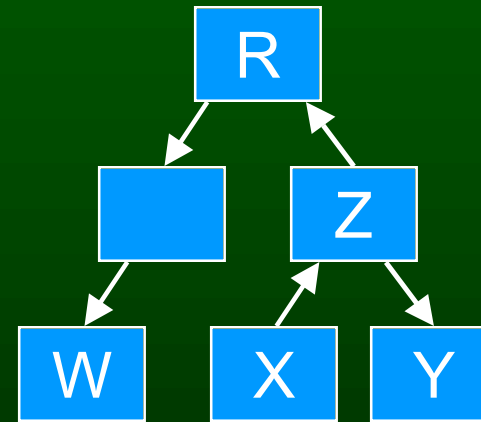
- Packet concatenation must not tie up bus for so long that the root cannot send Cycle-Start packet on time. This means that new links cannot use ack arbitration when cycle start is expected.
- 1394a Links and PHYs must support both “request” (1394-1995) and “accelerated_request”

Using bus topology

- From the standpoint of a single node, receiving a packet is the same whether the receive port is a child or parent port
- But from the bus standpoint, there are significant differences



- W starts packet
- X receives on parent port
- X & Y receive at same time

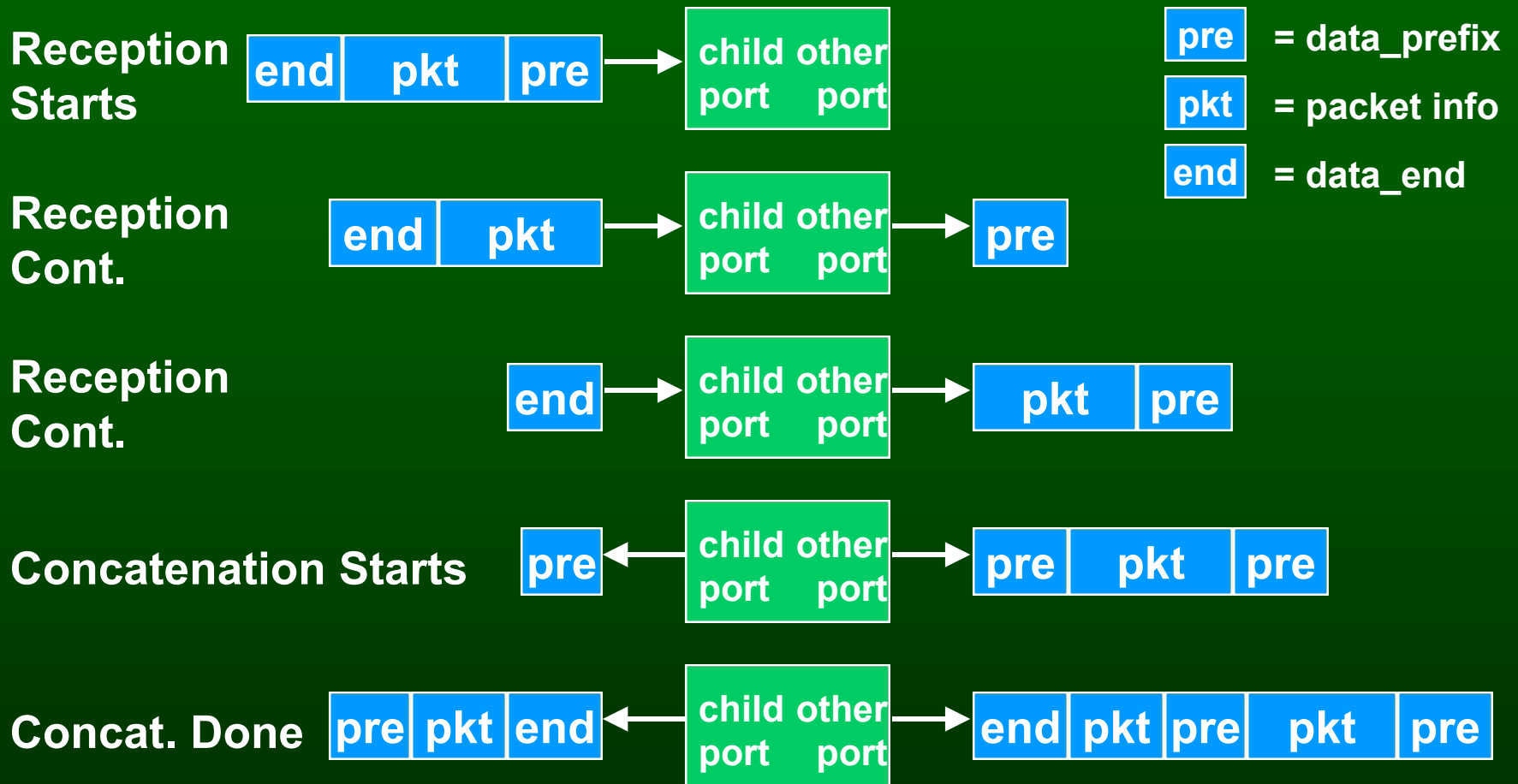


- X starts packet
- Z receives on child port
- No other node receives at exactly same time as Z

Fly-by arbitration outline

- In parent port receive case, other nodes may be receiving packet at exactly same time. Arbitration "tricks" could cause packet collision, since multiple nodes could try "trick" at same time.
- Child port receive case offers unique opportunity to exercise ingenuity!
- Consider what happens if a node receives a packet on a child port, and concatenates to an additional packet on the fly. Two possible opportunities:
 - ◆ Isochronous - result looks like isochronous concatenation
 - ◆ Received packet is an ack - result looks like ack concatenation

Fly-by arb operation



Note that the original receiving child port just sees two packets. Parent port and other child ports see concatenated packet string.

Fly-by arb asynch considerations

- Four conditions must be met for asynchronous fly-by arbitration:
 - ◆ Original packet must be received on a child port
 - ◆ Original packet must be an ack (Concatenating onto any other packet would leave multiple nodes on bus waiting for ack's)
 - ◆ Fundamental fairness protocol still applies
 - ◆ Packet concatenation must not tie up bus for so long that root cannot send Cycle-Start packet on time. This means that new links cannot use fly-by arbitration when cycle start is expected.
- New links and PHYs need both “request” (1394-1995) and “accelerated request”

Priority arbitration

- Arb reset gaps are expensive, try to reduce them
- Responses don't use fairness
 - ◆ No need, since there is at most one response for each request ... responses will not keep requests off the buss
- If less than 63 nodes, assign "unused" opportunities for requests to nodes that will most likely need them
 - ◆ Keep the total number of requests in a fairness cycle to 63, same as worst case 1394-1995 system
 - ◆ Same worst case latency as 1394-1995

Software changes needed

- Very little
 - ◆ P1394a PHY and Link hardware will do the right thing automatically
 - ◆ Need new software for assignment of extra arb attempts
- For very special cases, need to support new arbitration request types
 - ◆ Diagnostics only!

Backwards compatibility

- P1394a arbitration accelerations are totally interoperable with existing hardware and software.

Reset changes for P1394a

- Fast “arbitrated” reset
 - ◆ worst case reduced from 300 μ sec to about 100 μ sec
 - ◆ typical reduced from 165 μ sec to 5 μ sec
 - ◆ if even one “old” PHY exists in network, then old timing applies, but backwards compatibility is complete
- Connection debounce
 - ◆ wait for connection status to stabilize for 1/3 second before reporting connection
 - ◆ After first loss of connection, wait for 1/3 second of stable “no connect” before reporting new connection

Suspend/Resume

- Want to reduce power on the bus
 - ◆ Each PHY consumes at least 16 ma for bias plus 100-200 ma for clocked logic
- Use special low level current source to signal “disconnect” separately from “no bias”
- “No bias” but not “disconnect” means that the port is suspended
- Suspend can be initiated by special PHY packets
- Non-trivial implementation, difficult to test

Additional changes (1)

- PHY “pinging”
 - ◆ Measure time from sending request to a PHY until response is received
 - ◆ More accurate gap setting (compensates for varying cable lengths and PHY delays)
- 4-pin connector
 - ◆ much smaller ... but no power and much more difficult FCC Class A qualification for S200/S400 (current in the shield)
 - ◆ do not use for computer peripherals

Additional changes (2)

- PHY-Link interface standardized
 - ◆ Only “informative” in 1394-1995, now “normative”
 - ◆ Much improved specification
 - ◆ Includes improved PHY register set
 - ◆ Now only 16 ports per PHY allowed (down from 27)
- Minor inconsistencies removed

Hardware design changes needed

■ Link

- ◆ Minor interface change to PHY interface to support different queuing of requests
- ◆ Minor changes to interfaces to DMA to support different arbitration requests

■ PHY

- ◆ Moderate changes to state machines
- ◆ Small change to port interface to support new connect detect current

P1394b: the next step

- Full backwards compatibility with 1394-1995 and P1394a
- Extend speed ranges to 800, 1600 and 3200 Mbit/sec
- Support long distance systems
 - ◆ 50m, 70m, 100m

New connection model

- high speed PHYs communicate using continuously transmitted full duplex 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 Bristol

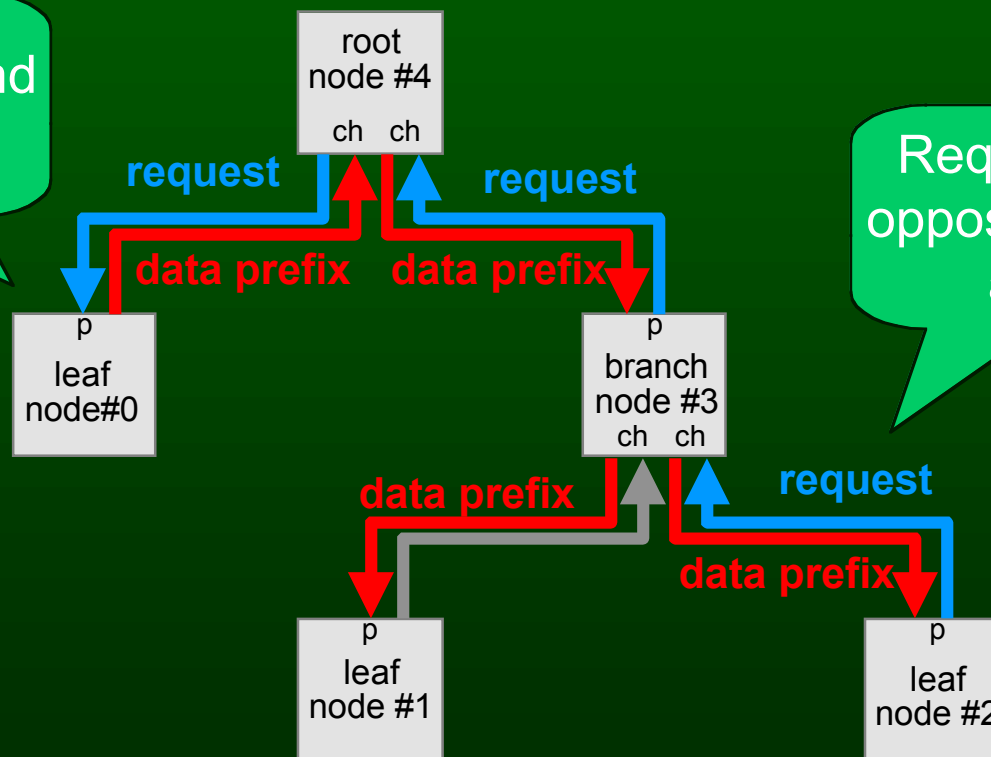
Additional media

- Good news for copper
 - ◆ Minor (and backwards compatible) changes to 6-pin connector and media work for 4.5m up to S1600
- Long distance at S800 and above requires optical fiber
 - ◆ 50 micron graded index OK
- Long distance at S100 can be done via UTP5\ul style="list-style-type: none;">- ◆ same as 100 Mbit Ethernet

Arbitration improvements (1)

- Full duplex connections means that arbitration can be pipelined!

Last node to send is the "boss"



Requests go in opposite direction as data

Arbitration improvements (2)

- General rule: send requests continually, and in the direction that packets come from
- Sender is the “BOSS” if there is no other implicit winner of arbitration
 - ◆ if sending a directed asynchronous packet, then the sender of the ACK is the implicit winner ... Will become the BOSS.
- Many fallback methods, error recovery straightforward
 - ◆ More cool technology, David LaFollette of Intel

Miscellaneous

- Startup methods are complex
 - ◆ May be connected via optical (no DC bias), so no direct connection signal
 - ◆ use “tones”
 - ◆ May connect using copper
 - ◆ Always use “beta” mode if possible
- Virtual node IDs
 - ◆ If bus_ID is specific (not 0x3FF), then phy_ID is “virtual” ... assigned by management
 - ◆ virtual IDs are persistent across bus resets

Changes needed for P1394b

- Link - P1394a plus:
 - ◆ Very fast system interface -> 100Mbyte/sec
-> 400 Mbyte/sec
 - ◆ Added speed codes, pipelined arbitration
- PHY - P1394a plus:
 - ◆ extensive changes in low level transmitter and receiver
 - ◆ clock recovery much more complex
 - ◆ timing margins much more difficult to meet

How to get ready for 1394a and b

- **Keep to the standard!**
 - ◆ New standards are only compatible with the old
 - ◆ Non-standard extensions will likely conflict with the A and B enhancements
- **Drivers should use standard system software, if possible**
 - ◆ In addition, for peripheral designers there are multiple sources of standard firmware
 - ◆ Those sources will track the standards for you
- **Be public with any proposed protocols**
 - ◆ Most protocols developed in isolation have been disasters
 - ◆ 1394 is remarkably subtle, ask the experts first!

About Zayante

Announcing the merger between two of
the most well known IEEE 1394
technology start-up companies

Macro Designs, Inc
&
Zayante, Inc.

Zayante's Mission

- To be the leading provider of total 1394 interoperability solutions
 - ◆ silicon IP components
 - ◆ software IP components
 - ◆ reference systems
 - ◆ tools and testing services
- Enabling vendors of consumer devices to unleash the full potential of IEEE 1394 and related technologies