

# ***Testing in a (Brave New) Serial World***

**John Lenthall**

**FuturePlus Systems Corporation**

**tech\_sup@futureplus.com**

# *Why Brave*

- New Serial buses provide features that allow complex system transactions previously usually seen on internal bus.
- Many new possibilities that can create complex scenarios within a system.
- Some of these scenarios may be new to node designers.
- Need to design, debug and test within these scenarios.

# *Why Brave*

- Testing is difficult enough with a parallel bus.
- Serial buses present new challenges.
- However debug and test are a significant part of the product cycle so appropriate choice of tools and methods can directly impact time to market.

# ***IEEE1394***

## **A Sophisticated, High Speed Serial Bus**

- Automatic assignment of node addresses.
- Distributed arbitration mechanism.
- Variable speed data transmission.
- Request/Response model for transactions.
- Block and Quadlet Reads and Writes.
- Isochronous mode provides a low overhead, guaranteed bandwidth service.

# ***IEEE1394***

## **A Sophisticated, High Speed, Serial Bus**

- Peer to Peer Communication Possible.
- Standard defines 3 protocol layers
  - ◆ Physical.
  - ◆ Link.
  - ◆ Transaction.
- Physical Layer supports both cable and Back-plane environments.

# *Challenges of Test*

(or “WOW!, How do I debug this?”)

- Complex system transactions now appear on a serial bus.
- Signaling is:
  - ◆ Low voltage.
  - ◆ Differential.
  - ◆ Uses a data/strobe pair.
- Protocol includes sequential elements and temporal elements (e.g. gaps).

# *Challenges of Test*

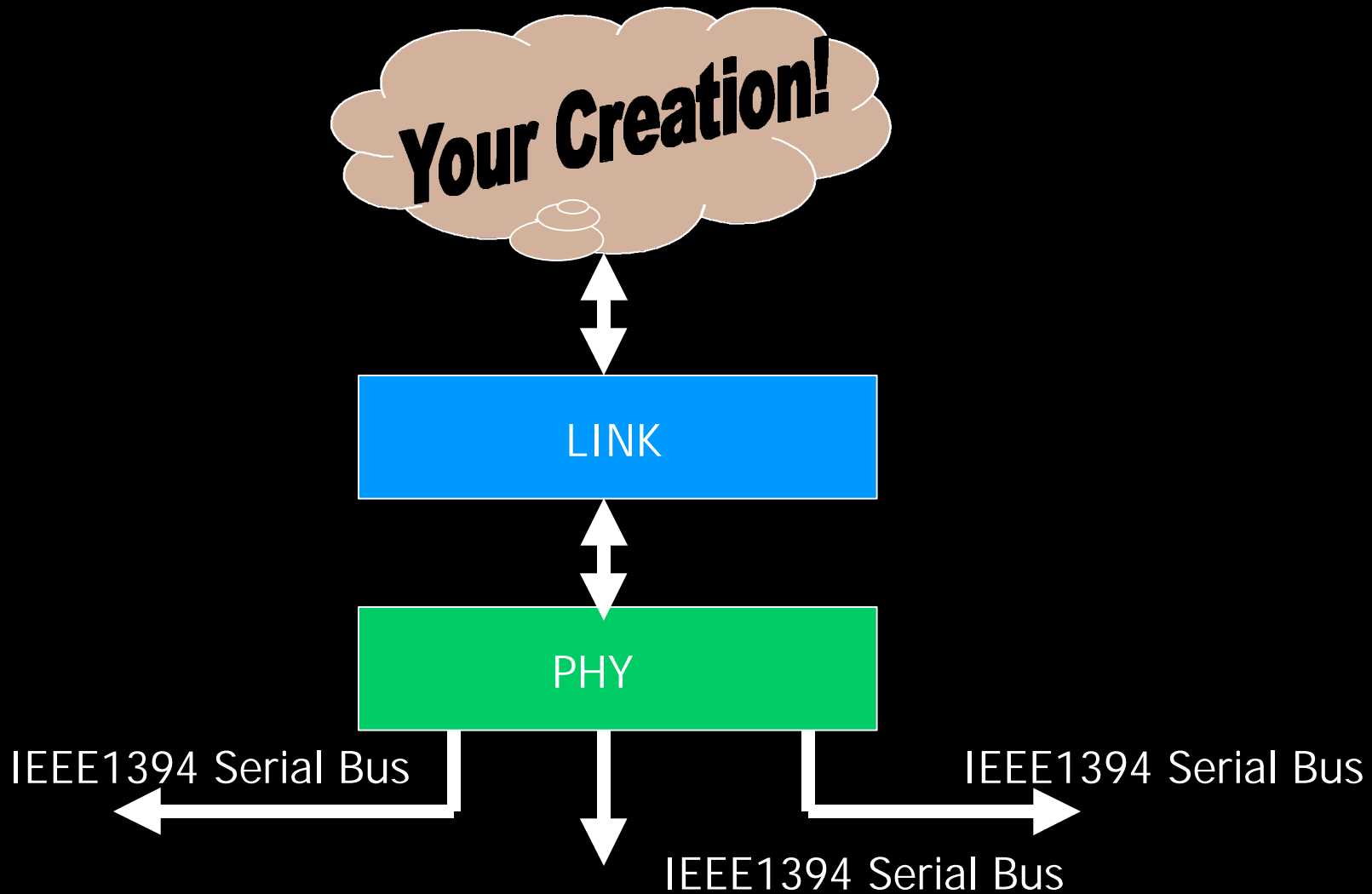
(Or “WOW! How do I debug this?”)

- Speed signaling and arbitration share data medium but are “overlaid”.
- Large amounts of data can occur on bus because of:
  - ◆ Large Address Space.
  - ◆ High Speeds.
  - ◆ Peer to Peer communication.

# *Challenges of Test*

- “Conventional” test equipment (logic analyzers, DSOs etc.) alone not adequate:
  - ◆ Not suited for the signaling methods.
  - ◆ Requires significant interpretation of signals and translation to protocol components.
  - ◆ May be difficult to attach.

# *Basic IEEE1394 Node*



# PHY

- A mixed signal device that provides:
  - ◆ Electrical interface to the IEEE1394 bus.
  - ◆ Arbitration services.
  - ◆ Speed signaling.
  - ◆ Determination of connection status.
  - ◆ Detection of bus “events”.
  - ◆ Interface to LINK.

- A digital device that provides:
  - ◆ A packet delivery service that transmits and receives correctly formatted IEEE1394 packets.
  - ◆ CRC generation and checking.
  - ◆ Provides interfaces to:
    - ◆ A “host bus”
    - ◆ PHY
    - ◆ Transaction layer (asynchronous packets)
    - ◆ Application (isochronous packets)
  - ◆ May provide **Cycle Master** function for isochronous transfers.

# *Reasons To Look* “Out of Spec” Behavior

- Electrical Signaling e.g.
  - ◆ Jitter, Skew.
  - ◆ Voltage Levels.
- Protocol Timing Problems e.g.
  - ◆ ARB Reset gap.
  - ◆ Reset gap.
  - ◆ Sub Action gaps.
  - ◆ ACK gaps.
  - ◆ Cycle Timing.

# *Reasons To Look* “Out Of Spec” Behavior

- Protocol Behavior e.g.
  - ◆ Arbitration (good citizen behavior).
  - ◆ Speed Signaling.
  - ◆ Handling of Retries.
  - ◆ Concatenation of different speed packets (P1394a).

# *Reasons to Look* Functional “Bugs”

- Unexpected Behavior.
- “Broken” Behavior.
- No Behavior!
- For example:
  - ◆ Interaction of multiple, multi-threaded transactions.
  - ◆ Buffer management.

# *Reasons to Look* Performance Characterization

- Latency for ACKs.
- Latency for Responses.
- Understanding gap timings.
- Traffic pattern analysis e.g.
  - ◆ Distribution of packet types.
  - ◆ Addresses.
  - ◆ Channels.

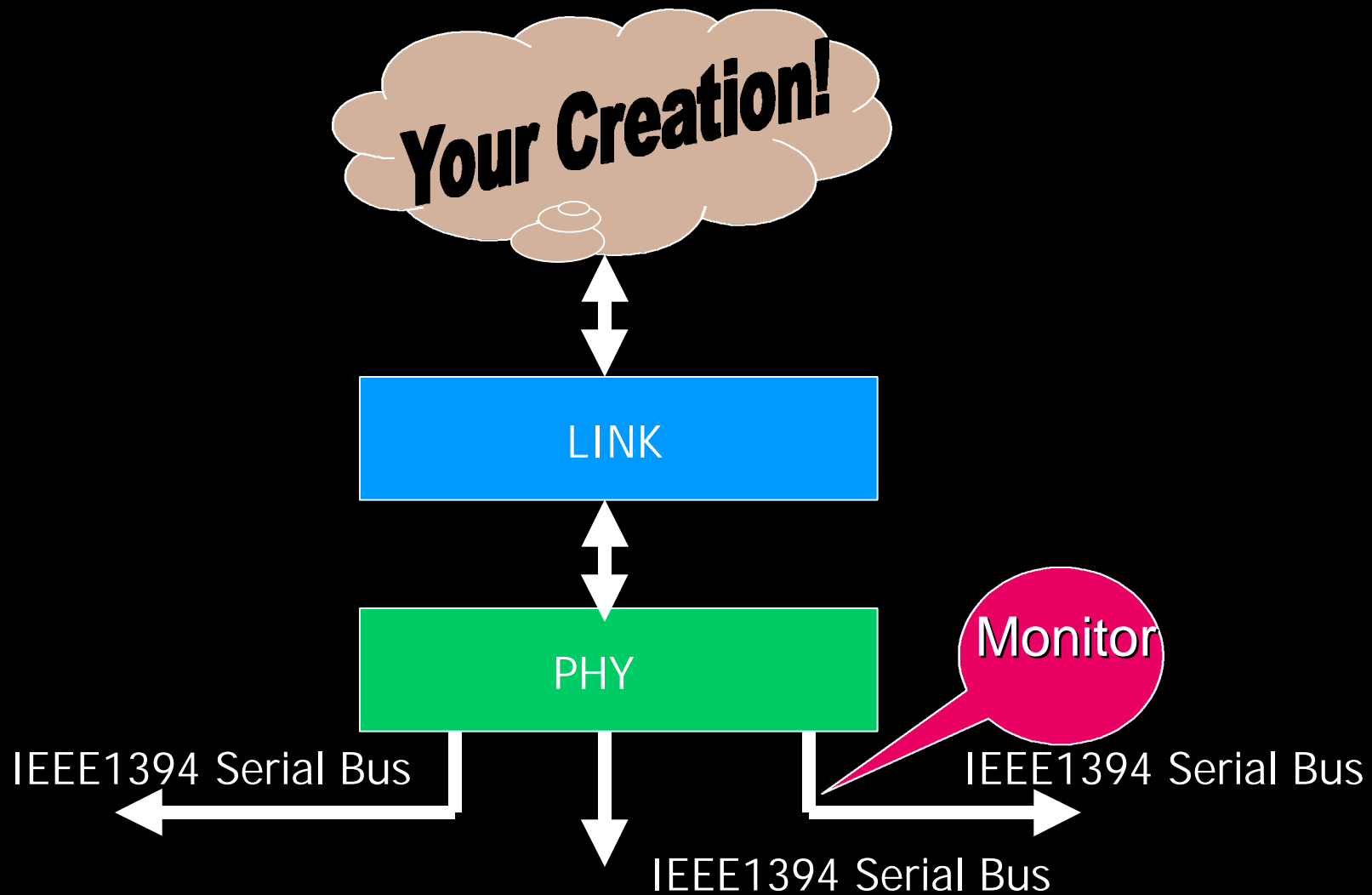
# *Where Will You Spend Time?*

- Trying to figure out how your “cloud” and the rest of the bus are interacting.
- Probably tracking packets and or transactions through a node and across the bus.
- Trying to correlate events & traffic in one domain with events & traffic in another.

## *What to look at*

- In general you need to monitor along the path of a transaction. Ideally at least 2 points.
  - ◆ **Reminder:** path of the transaction crosses a number of protocol & signal domains.
- Where might you look with IEEE1394?
  - ◆ On the bus.
  - ◆ Between the PHY & LINK.
  - ◆ Behind the LINK.

# Monitoring the bus



# *Monitoring on the bus*

## ■ Advantages

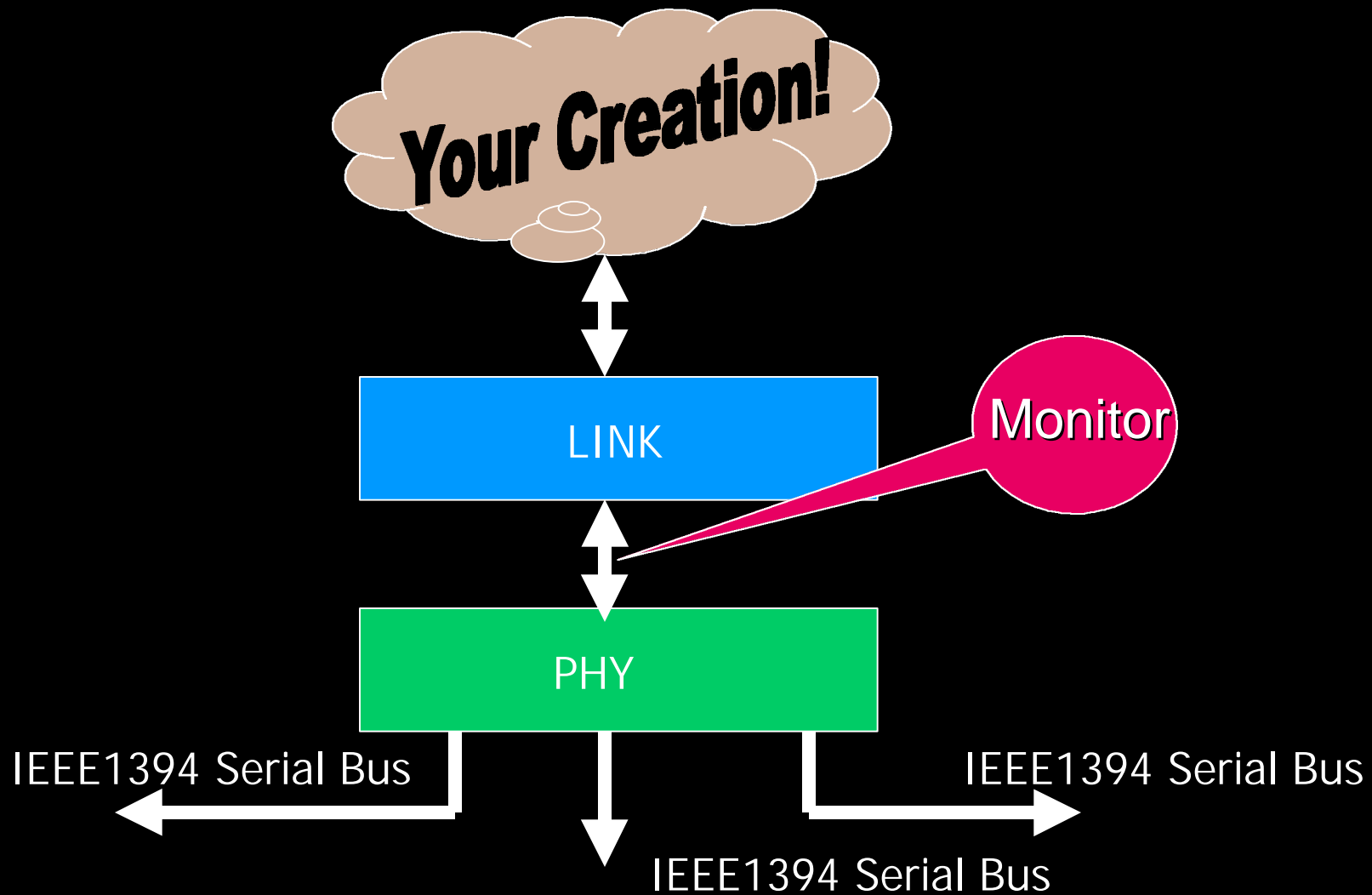
- ◆ Physical layer parameters visible.

## ■ Disadvantages

- ◆ Differential, low voltage signaling.
- ◆ Difficult to interpret packets & transactions.
- ◆ Instrumentation may interfere with the bus.
- ◆ Serial format means large capture memory needed.

To track a transaction it is necessary to monitor the bus

# Monitoring PHY-LINK



# *Monitoring PHY-LINK*

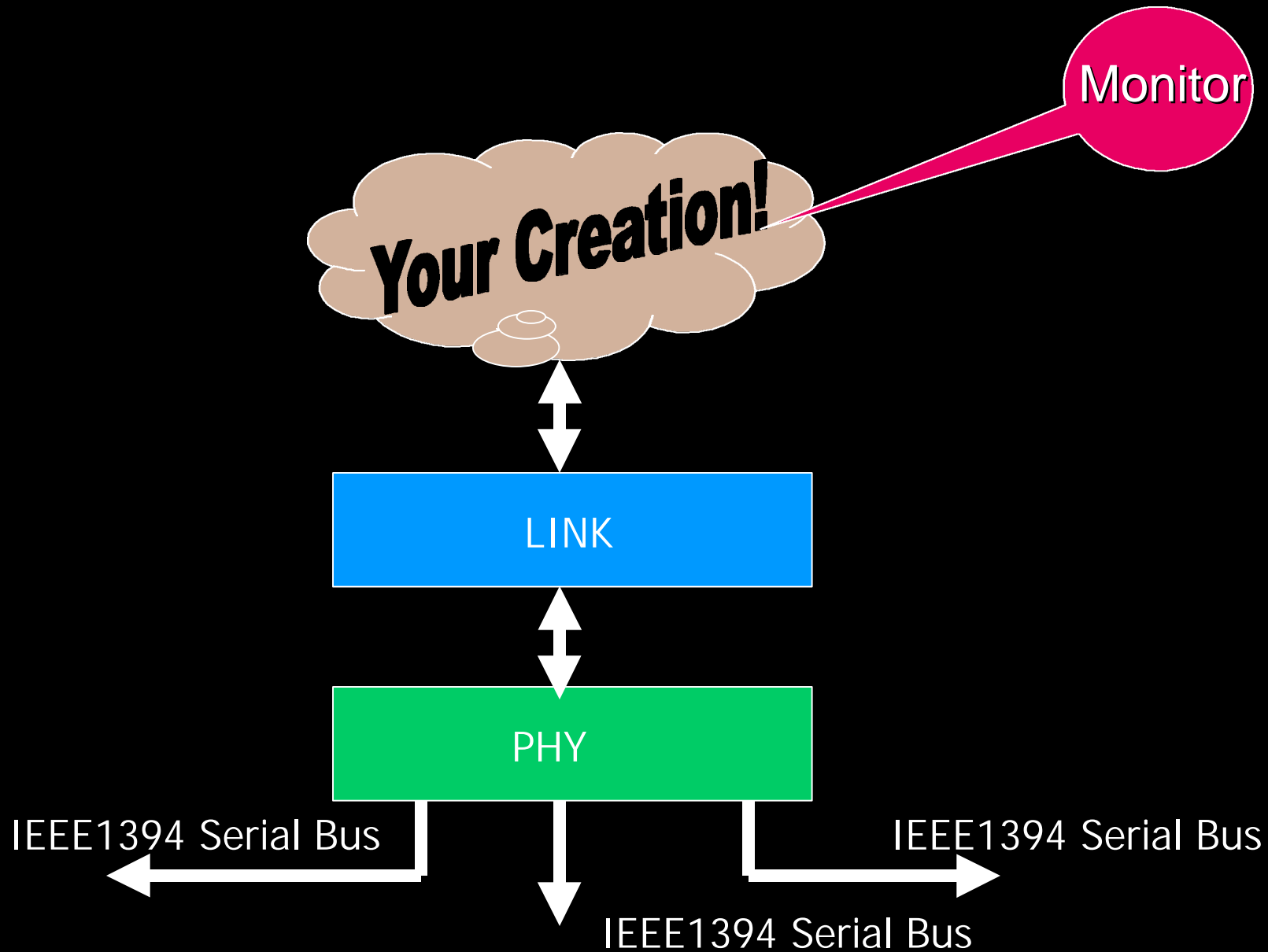
## ■ Advantages

- ◆ PHY provides indication of bus events and status.
- ◆ PHY recovers data.
- ◆ There is a defined PHY-LINK interface (P1394a; informative in IEEE1394-95).

## ■ Disadvantages

- ◆ Packets partially serial.
- ◆ Instrumentation may require special care if galvanic isolation used.
- ◆ Sequential Protocol.

# Monitoring LINK-Host interface



# *Host-LINK INTERACE*

## ■ Advantages

- ◆ Easy to see packets.
- ◆ Easier to see how the “local” system is interacting with IEEE1394 interface.
- ◆ “Standard Logic level” signaling.
- ◆ Probably familiar protocol on host side e.g. PCI.

## ■ Disadvantages

- ◆ Can't see other IEEE1394 bus activity.
- ◆ Farther away from IEEE1394 bus.

# So?

- Following a transaction or packet end to end crosses multiple signaling and protocol domains.
  - ◆ Need to observe more than one point.
  - ◆ Have the capability on hand & support in the design to monitor each domain.
  - ◆ Need the ability for an event in one domain to trigger data capture in another domain.
  - ◆ Need to be able to refine data set to be captured.

# *A Tool to Help*

- A class of tools that addresses some of these issues is bus analyzers.

# ***What is a Bus Analyzer?***

- Typically provides capture and display of bus packets & events in a readable format.
- Usually provides some form of triggering and filtering capability.
- May detect error conditions.
- Simplifies acquisition and interpretation of bus activity.
- A IEEE1394 bus node that monitors the bus.

# Sample Bus Analyzer Display

| State Number | FUTUREPLUS SYSTEMS c 1998  | ADDR     | DATA     | STAT | DATA     | STAT | DATA     |
|--------------|--|----------|----------|------|----------|------|----------|
| Decimal      | IEEE 1394 BUS TRANSACTIONS REV 0.3   | Hex      | Hex      | Hex  | Hex      | Hex  | Hex      |
| 62. 0        | ARB Reset Gap  | 00000000 | 00000000 | 0981 | 00000000 | 0981 | 00000000 |
| 1            | WR_REQ_DATA_BLOCK @S400;<br>TLABEL= 01H; RETRY CODE= retry_1<br>DST_ID;Bus= 000H;Node= 03H   | 00000000 | 00030410 | 8B01 | 00030410 | 8B01 | 00000000 |
| 2            | SRC_ID;Bus= 000H;Node= 05H   | 00000000 | 00050607 | 8B02 | 00050607 | 8B02 | 00000000 |
| 3            | DST_OFFSET= 060708090A0CH  | 00000000 | 08090A0C | 8B03 | 08090A0C | 8B03 | 08090A0C |
| 4            | DATA LENGTH= 12  | 00000000 | 000C0000 | 8B04 | 000C0000 | 8B04 | 00000000 |
| 5            | HDR CRC: 10111214H   | 00000000 | 10111214 | 8B45 | 10111214 | 8B45 | 10111214 |
| 6            | DATA= FFFEFDFFCH   | 00000000 | FFFEFDFF | 8B00 | FFFEFDFF | 8B00 | FFFEFDFF |
| 7            | DATA= FBFAF9F8H  | 00000000 | FBFAF9F8 | 8B00 | FBFAF9F8 | 8B00 | FBFAF9F8 |
| 8            | DATA= F7F60000H  | 00000000 | F7F60000 | 8B00 | F7F60000 | 8B00 | F7F60000 |
| 9            | DATA CRC: D4836204H  | 00000000 | D4836204 | 8BC0 | D4836204 | 8BC0 | D4836204 |
| 61. 10       | ACK Packet: ack_complete   | 00000000 | 1E111213 | 6300 | 1E111213 | 6300 | 1E111213 |
| 11           | SUBACTION Gap  | 00000000 | 00000000 | 1181 | 00000000 | 1181 | 00000000 |
| 12           | RD_REQ_DATA_QUADLET @S400;<br>TLABEL= 02H; RETRY CODE= retry_1<br>DST_ID;Bus= 000H;Node= 01H | 00000000 | 00010840 | A301 | 00010840 | A301 | 00000000 |
| 13           | SRC_ID;Bus= 000H;Node= 04H   | 00000000 | 00040C40 | A302 | 00040C40 | A302 | 00000000 |
| 14           | DST_OFFSET= 0C400A090A08H  | 00000000 | 0A090A08 | A303 | 0A090A08 | A303 | 0A090A08 |
| 15           | HDR CRC: 0C5739ACH   | 00000000 | 0C5739AC | A3C4 | 0C5739AC | A3C4 | 0C5739AC |
| 16           | ACK Packet: ack_pending  | 00000000 | 2D111213 | 6300 | 2D111213 | 6300 | 2D111213 |
| 17           | SUBACTION Gap  | 00000000 | 00000000 | 1181 | 00000000 | 1181 | 00000000 |

# *Typical Triggering*

- Events e.g.
  - ◆ Bus reset.
  - ◆ Cycle Start.
  - ◆ Error events.
  - ◆ Arb Reset.
  - ◆ Sub Action Gaps.
- Header Fields e.g.
  - ◆ Destination Node ID.
  - ◆ Source Node ID.
  - ◆ Channel Number.
  - ◆ Tcode.
  - ◆ Transaction Label.
  - ◆ Retry Code.
- Some combination or sequence.

# *Filtering*

- Reduces data set to be searched and analyzed.
- Can reduce time needed to find activity of interest.
- Optimizes use of acquisition memory.
- Filter (for example) on:
  - ◆ Node addresses.
  - ◆ Transaction Types.
  - ◆ Data payload.

# A “Simple” Example

OR: “Is it really that difficult?”

- Need a simple trigger, say `cycle_start`:
  - ◆ LINK-Local Host Interface:
    - ◆ Instrument the interface.
    - ◆ LINK **may** provide accessible h/w indication of event (e.g. and interrupt) but probably shared with other events.
    - ◆ Next find interrupt response cycle - subject to interrupt response **latency**.
    - ◆ Decode to check that interrupt is `cycle_start`.

# A “Simple” Example

- Need a simple trigger, say `cycle_start`:
  - ◆ PHY-LINK Interface:
    - ◆ Instrument the PHY-LINK interface
    - ◆ First determine PHY is receiving a packet
    - ◆ Next determine speed
    - ◆ Check packet length  $> 2$  quadlets
    - ◆ Finally “parse” packet (possibly 2 bits at a time) to find TCODE for `cycle_start`

# A “Simple” Example

- Need a simple trigger, say `cycle_start`
  - ◆ On the IEEE1394 bus with a ‘scope or similar
    - ◆ Instrument the bus.
    - ◆ monitor TPA/TPA\* and TPB/TPB\* pairs to detect and decode data prefix and speed signaling.
    - ◆ Detect start of packet.
    - ◆ Parse packet (1 bit at a time) for TCODE of `cycle_start`.

# A “Simple” Example

- Need a simple trigger, say `cycle_start`
  - ◆ On the IEEE1394 bus with a bus analyzer
    - ◆ Instrument the bus (connect in the analyzer)
    - ◆ Set trigger “IF `cycle_start` THEN trigger”

# *More on Bus Analyzers*

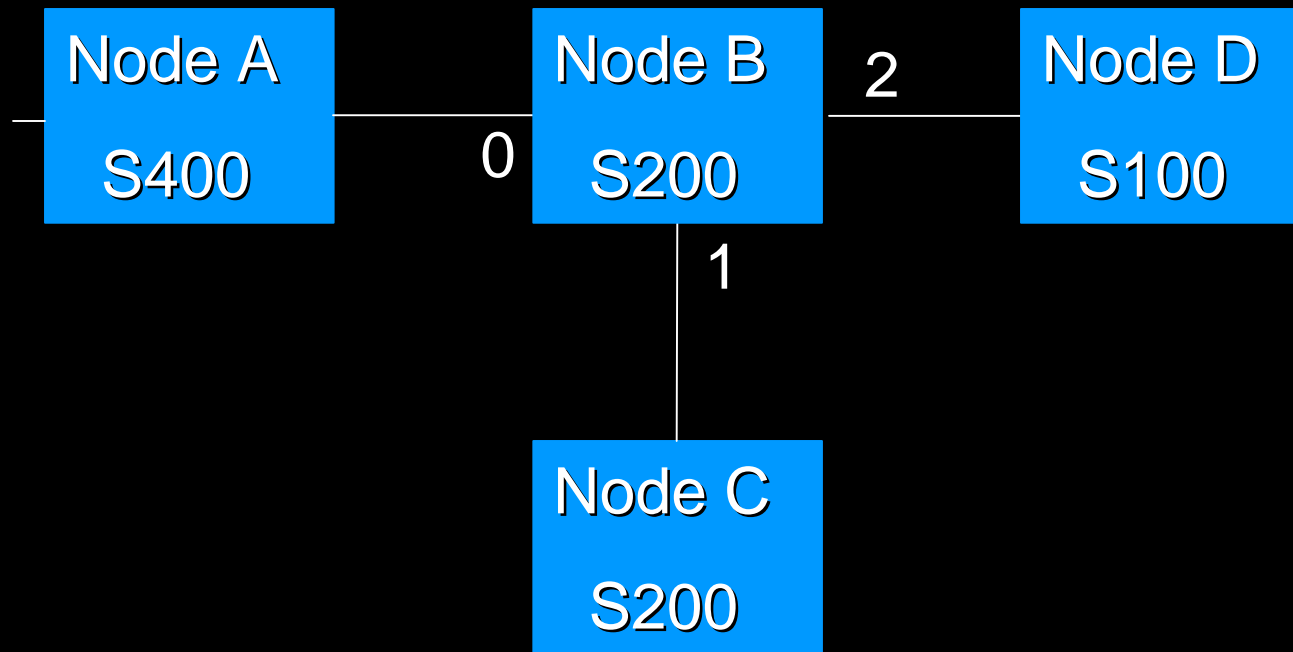
- Probably has a PHY interface
- Provides observability on a bus from the *view point of a node.*

# *Using a Bus Analyzer*

- Topology of bus is important in IEEE1394.
- Need to place analyzer with care to be sure to see all traffic relevant to the problem.
- For example need to consider such “characteristics” as:
  - ◆ Speed traps.
  - ◆ Re-assignment of node IDs.
  - ◆ Effects of arbitration enhancements.

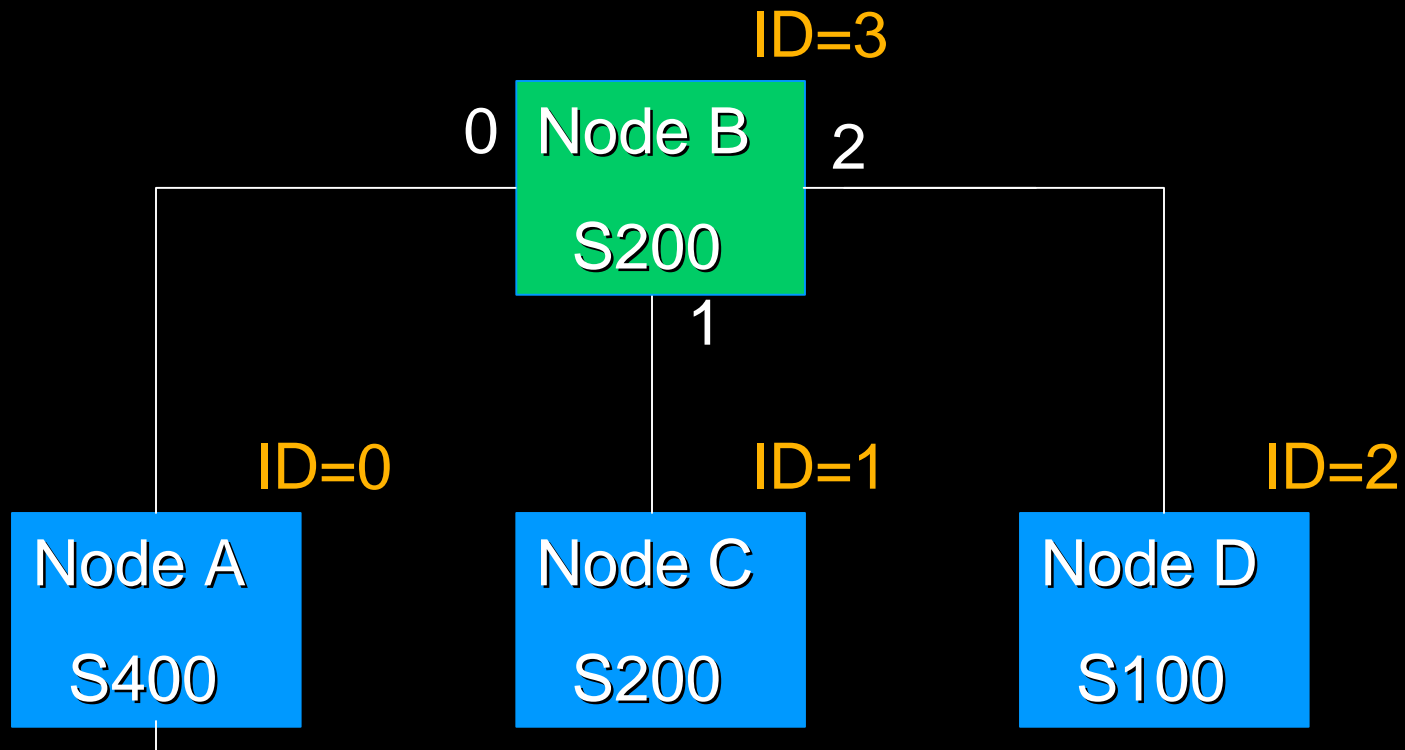
# Using a Bus Analyzer

## A Simple Bus Structure



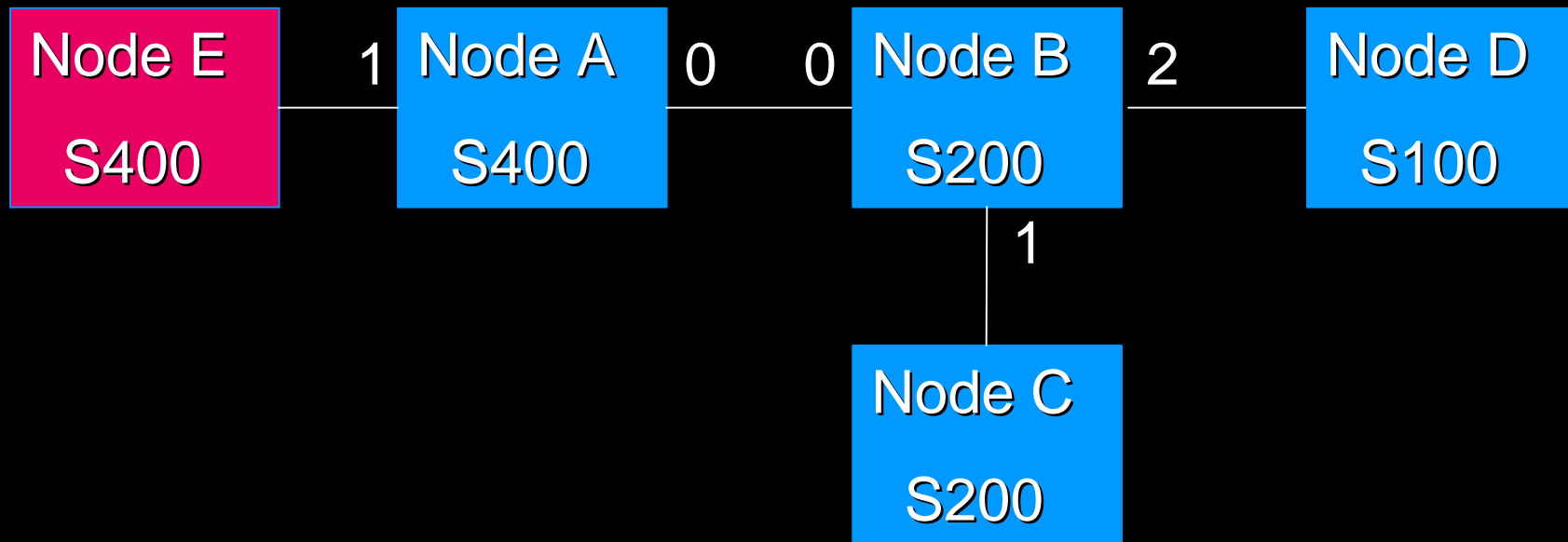
# Using a Bus Analyzer

## Topology after Bus Configuration



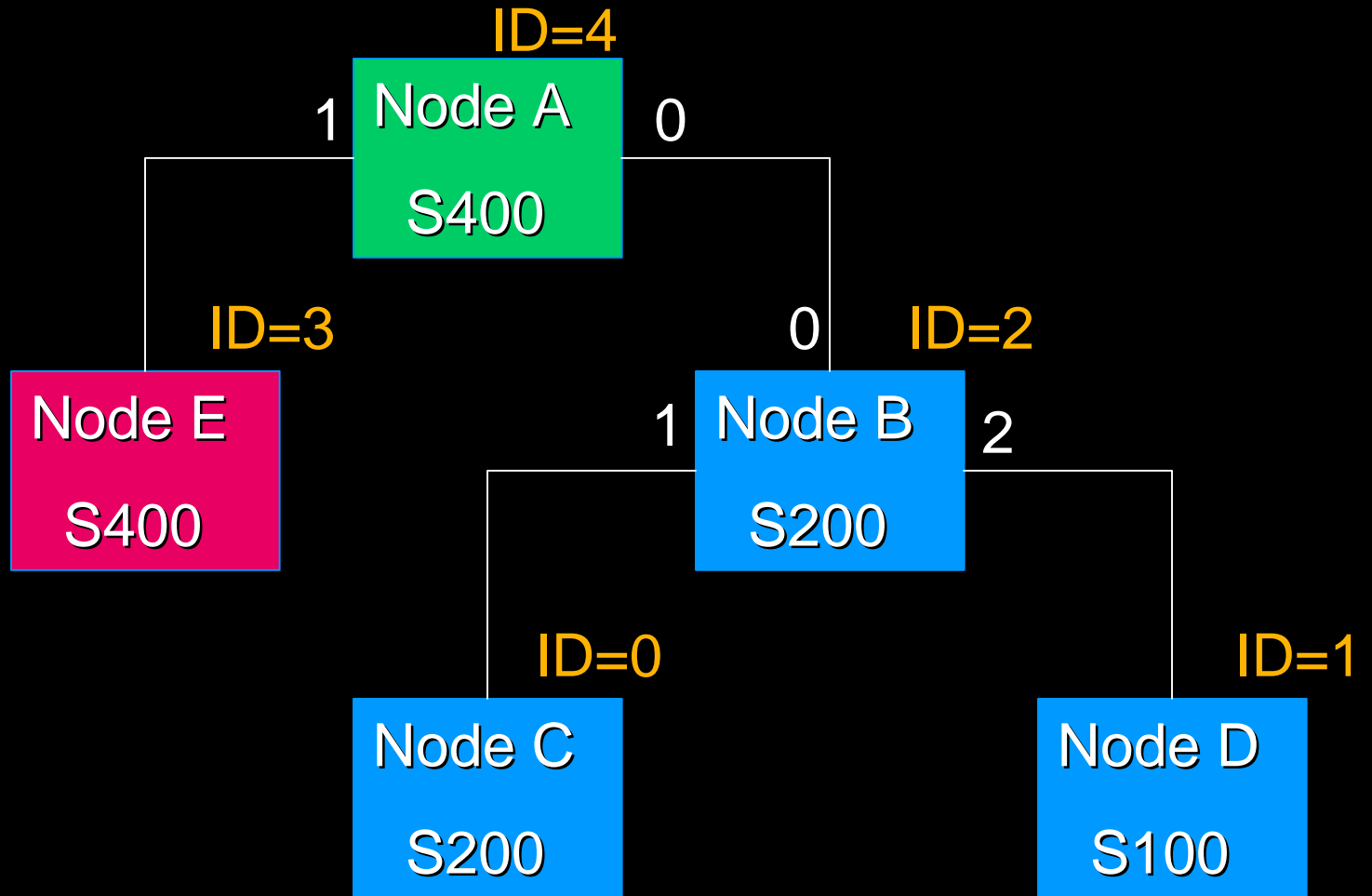
# Using a Bus Analyzer

New Structure - Add a node



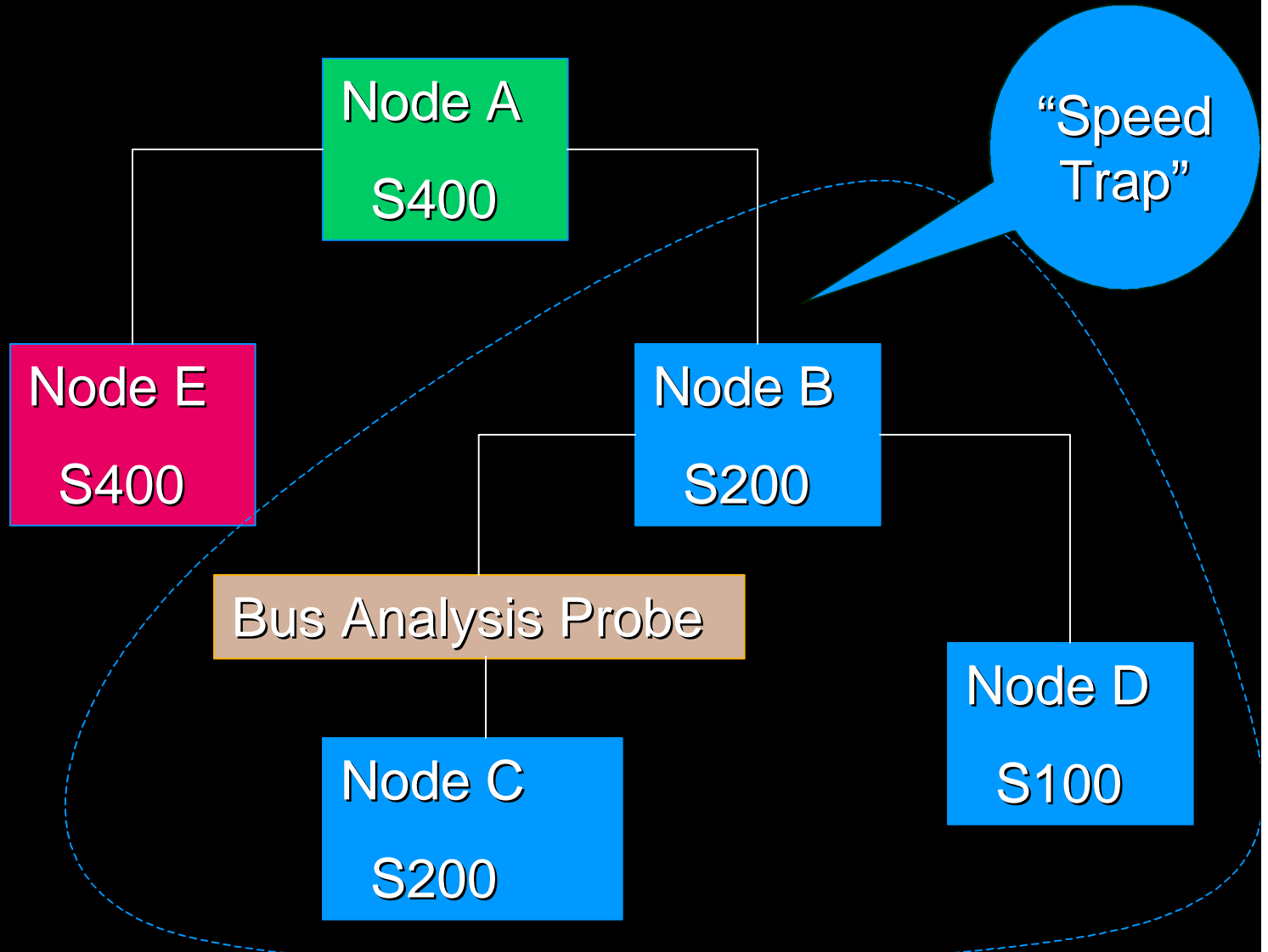
# Using a Bus Analyzer

## New Topology



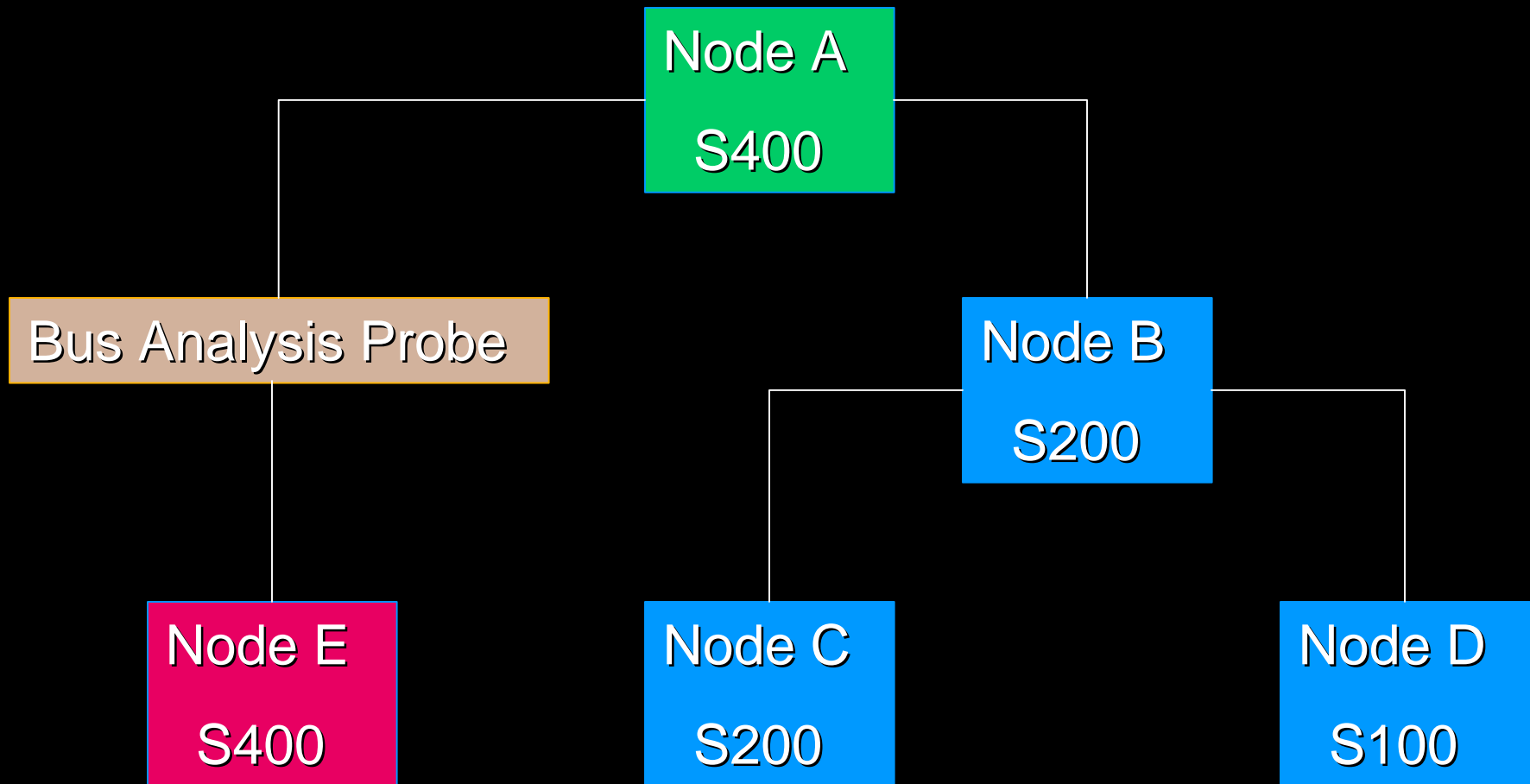
# Using a Bus Analyzer

## Restricted Observability



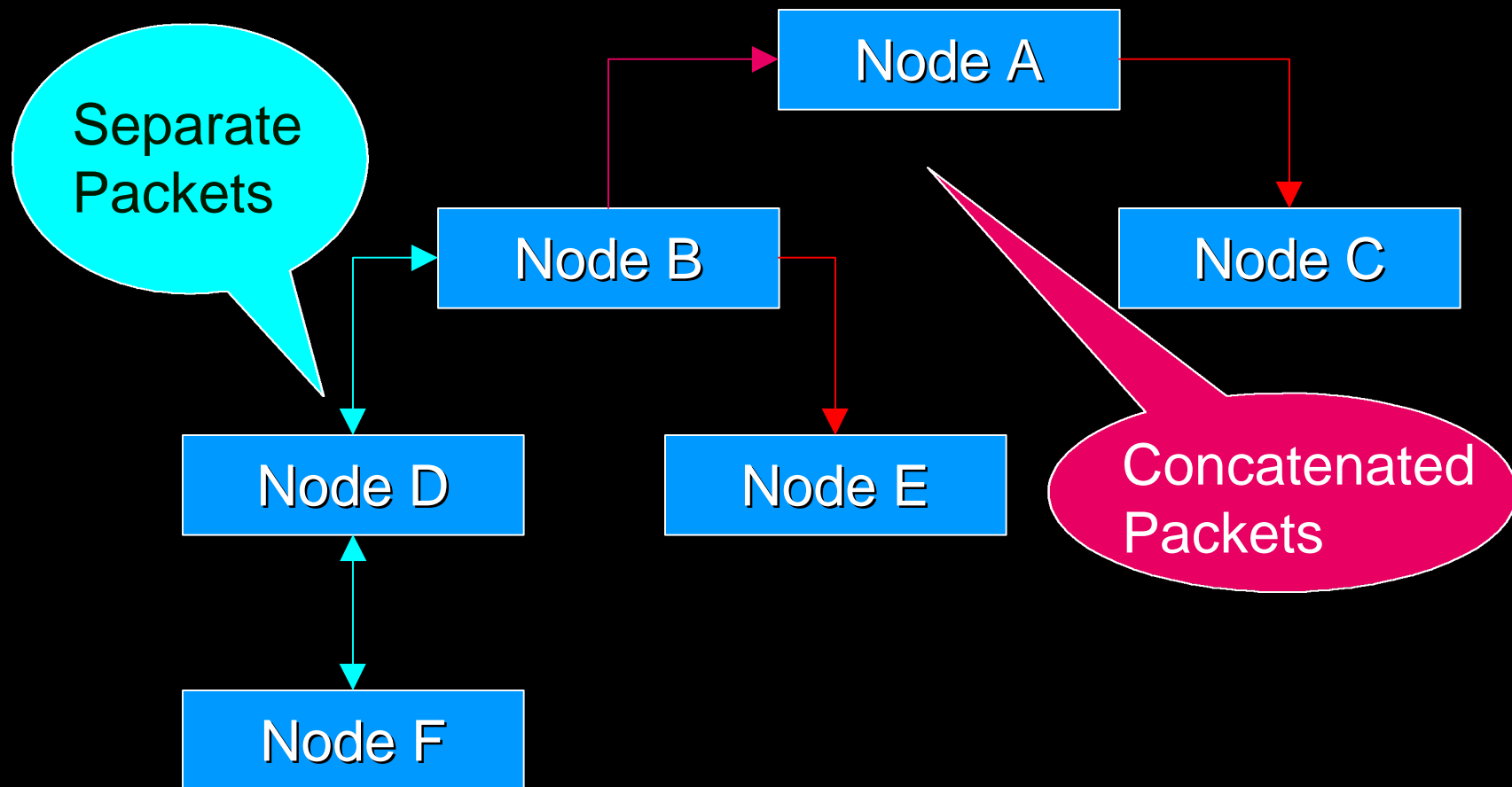
# Using a Bus Analyzer

Improved Observability

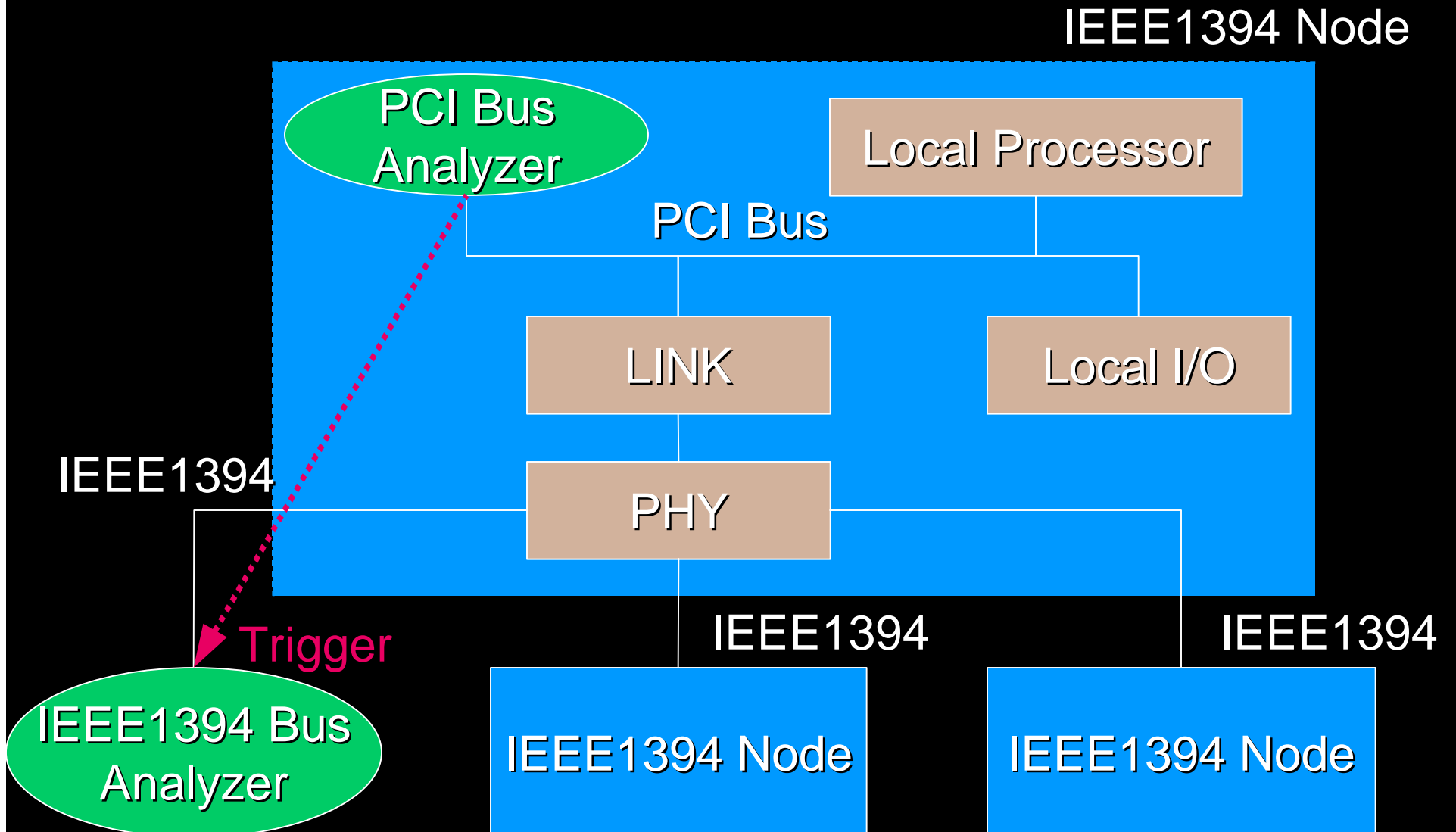


# Using a Bus Analyzer

Different views of the same transactions



# Cross Domain Analysis



# *Summary*

- IEEE1394-95, P1394a and soon P1394b are complex interconnects.
- Understanding of system dynamics is increasingly important.
- Thorough testing of both nodes and systems will be a challenge.
- Understanding the protocols, your design and the systems is still necessary.
- May need to look at multiple domains.

# Summary

- Tools such as Bus Analyzers help to reduce “time to insight” on a problem by:
  - ◆ Providing straightforward interface to the bus.
  - ◆ Displaying bus “packets” in a readable form.
  - ◆ Providing protocol specific data acquisition controls e.g.
    - ◆ Triggering
    - ◆ Filtering
    - ◆ Pattern searching
  - ◆ Aid in detecting protocol violations.

# *Resources*

- FuturePlus Systems produces a number of products that can assist in test of IEEE1394 based systems.
- For information:
  - ◆ <http://www.futureplus.com>
  - ◆ FuturePlus Systems Corporation
    - ◆ 6455 N. Union Ste 202
    - ◆ Colorado Springs, CO 80918
    - ◆ (719) 278-3540