Communication Systems and Protocols

Exercise: FireWire

IEEE Std 1394-2008

- Three primary applications have driven the design and architecture of the serial bus:
  - **an alternate for a parallel backplane bus**
  - **a low-cost peripheral bus and**
  - **a bus bridge between architecturally compatible 32-bit buses.**

- Version:
  - The description provided here is based on IEEE Std 1394-2008

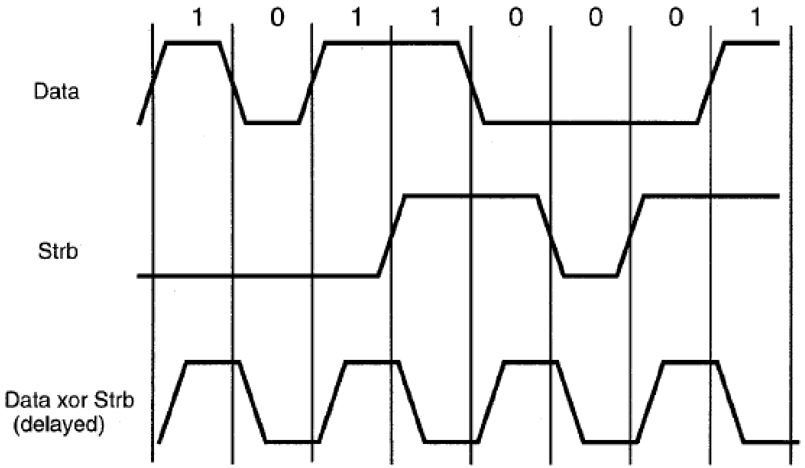
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Data Transmission: Data-Strobe (DS) encoding

- During packet transmission, there is only a single node transmitting on the bus, so the entire media can operate in a half-duplex mode using two signals: **Data and Strb.**

- NRZ data are transmitted on Data and is accompanied by the Strb signal, which changes state whenever two consecutive NRZ data bits are the same, ensuring that a transition occurs on either Data or Strb for each data bit.

- A clock that transitions each bit period can be derived from the exclusive-or of Data with Strb as shown in Figure



DS Encoding

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable environment

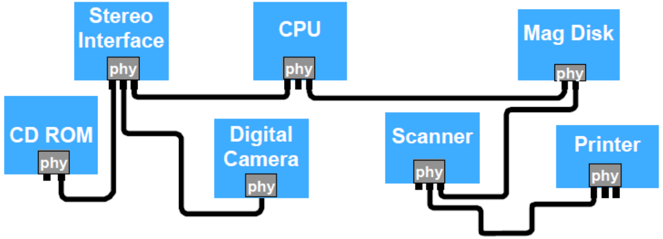
- Here we focus on the arbitration in the cable environment.

- First describe the cable environment in detail:

  - The cable environment is a network of nodes connected by point-to-point links called physical connections.

  - The physical connection consists of a port on the PHY of each node and the cable between them.

  - **The primary restriction is that nodes have to be connected together as an acyclic graph (no loops).**



Cable Environment

Fig Source : https://grouper.ieee.org/groups/802/802_tutorials/04-July/NewTechIntroTo1394.pdf

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

System Configuration

- The cable arbitration takes advantage of the point-to-point nature of the cable environment by having each node handshake with its immediate neighbors to determine ownership of the media.

- **Prior to normal operation the system has to be configured.**

- System configuration is done in three phases
  1. **Bus initialize**
  2. **Tree identify**
  3. **Self-identify.**

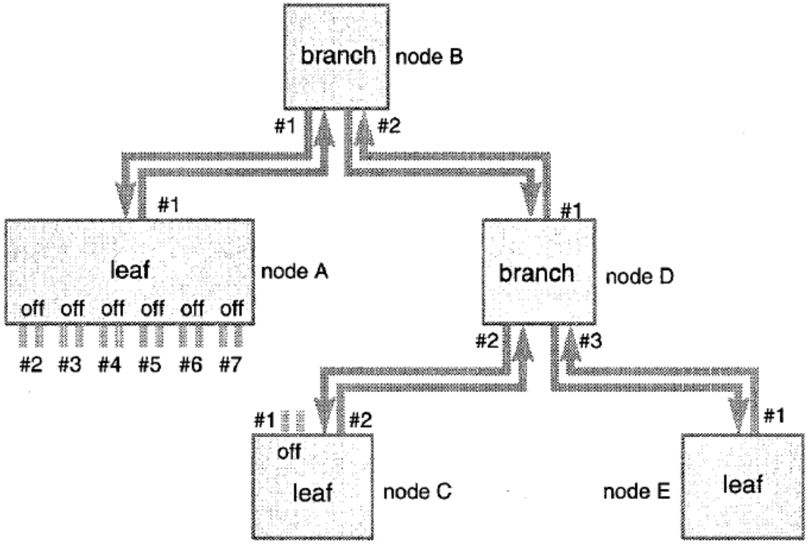
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

1. Bus Initialize

- Whenever a node joins the bus, a bus reset signal forces all nodes into a special state that clears all topology information.

- After the bus initialization process, the only information known to a node is whether it is a
  - **Branch:** more than one directly connected neighbor or
  - **Leaf:** only a single neighbor or
  - **Isolated:** unconnected

- The eventual root may be either a branch or a leaf!



After Bus Initialization

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

- The tree identify process translates the general network topology into a tree, where one node is designated a root and all of the physical connections have a direction associated with them pointing towards the root node.

- The direction is set by labeling each connected port as a
  - **Parent:** connected to a node closer to the root or
  - **Child:** port connected to a node further from the root

- Any unconnected ports are labeled “off” and do not participate in further arbitration processes.

- Any loop in the topology is detected by a timeout in this process.

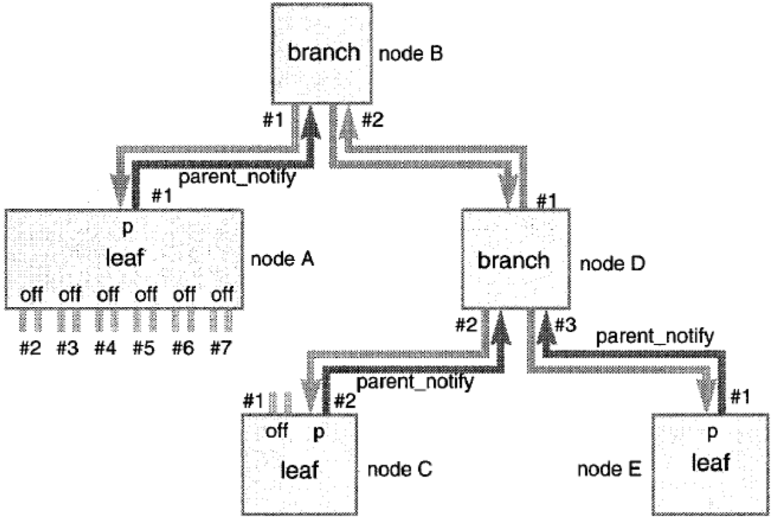
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

- The first step is for all leaf nodes to notify their probable parents. This is done by sending a *parent_notify* packet (signal)

- In this example, nodes A, C, and E send a *parent_notify* to their single connected port.

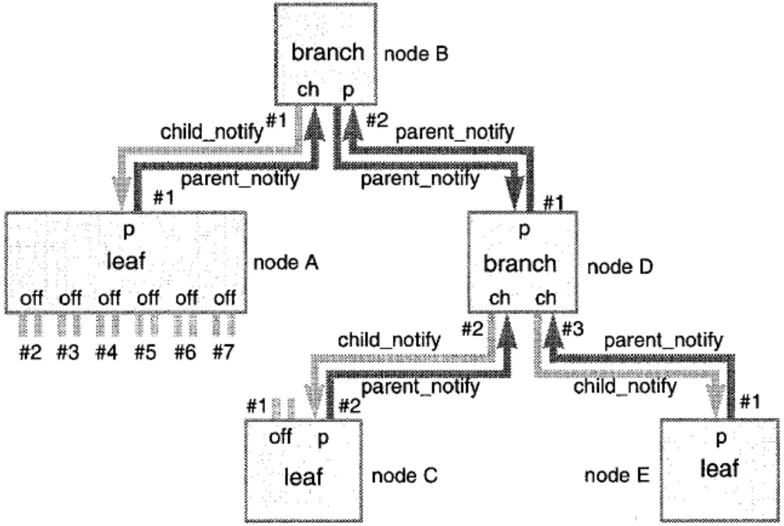
- This is the start of the parent-child handshake process.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

- The nodes internally recognize the *parent_notify* signals and mark the ports receiving them as child ports.
- If these nodes have one unidentified port, they send *parent_notify* up to their probable parents.
- At the same time the nodes send down *child_notify* signals to their child ports.
- Example here: Nodes B and D have one port remaining that is connected but not yet identified as child or parent. Therefore a *parent_notify* signal is sent.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

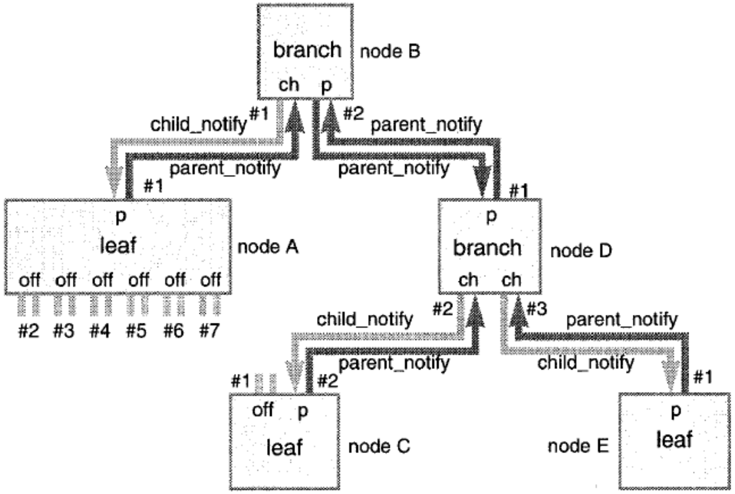
- When the leaf nodes receive the *child_notify* signal, that port is assigned as their parent port. Once assigned, their *parent_notify* signal is withdrawn.

- **Nodes having only child ports are assigned as the root.**

- It can occur that multiple nodes are receiving only *parent_notify* signals. This leads to a process called “root contention”

- Example:
  - Both nodes B and D discover that they are receiving the *parent_notify* port
  - Since one of the two nodes has to become the parent of the other, this collision of intentions starts a process called “root contention”.

- This is resolved by withdrawing the *parent_notify* signals. A timer is assigned on each node with a random duration and the signals are resent after the time elapses.



Start of root contention

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

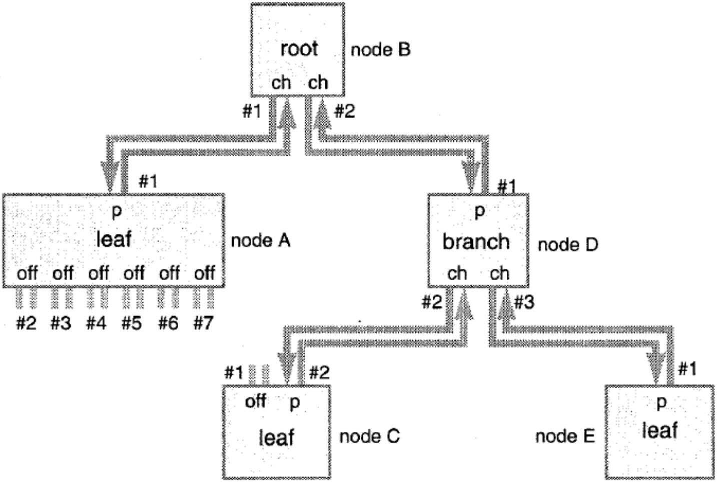
- When a node has all ports labeled as children, it takes the root function for itself.

- In this example:
  - Node D resends the signals first due to root contention
  - Node B has all ports as children and is assigned as the root.

- Note that the selection of the root node is not topology dependent. It is completely acceptable that the root node also be a leaf.

- The node that waits the longest after the bus reset to start participating in the tree identify process becomes the root.

- A particular node can be forced to wait a longer time by setting a *force_root* bit



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

3. Self-Identify Process

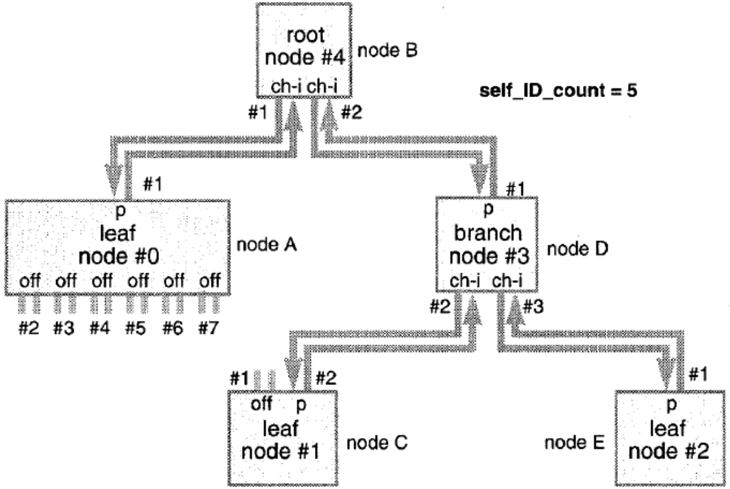
- The next step is to give each node a unique physical_ID

- The self-identify process uses a deterministic selection process, where the root node passes control of the media to the node attached to its lowest numbered connected port first. The child nodes then passes control in a recursive manner

- The first leaf node to receive control, during its self-identify process selects 0. The next node which receives control selects 1 and so on

- The root then passes control to the next lowest numbered port. When the nodes attached to all the ports of the root are finished, the root itself does a self-identify process.

- **Note that each port of the node is individually numbered. There is no particular order to the numbering, it is just a way to give each port a unique label**

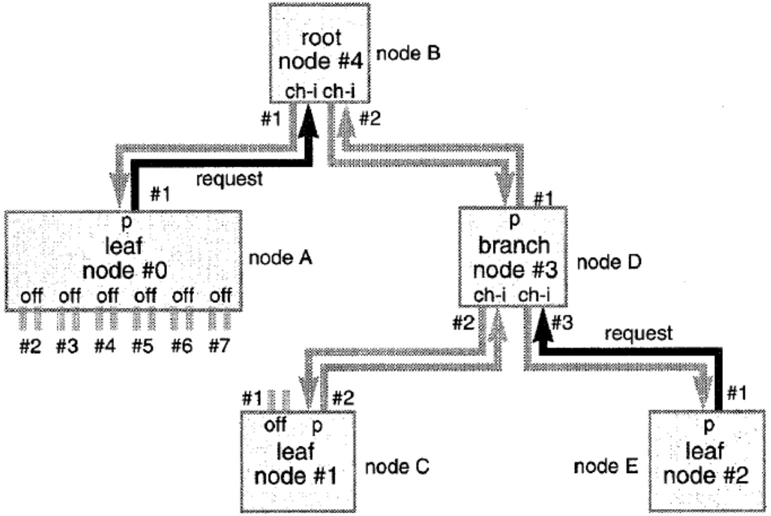


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Arbitration

- Once the self-identify process is complete, nodes can use the normal arbitration method to send packets

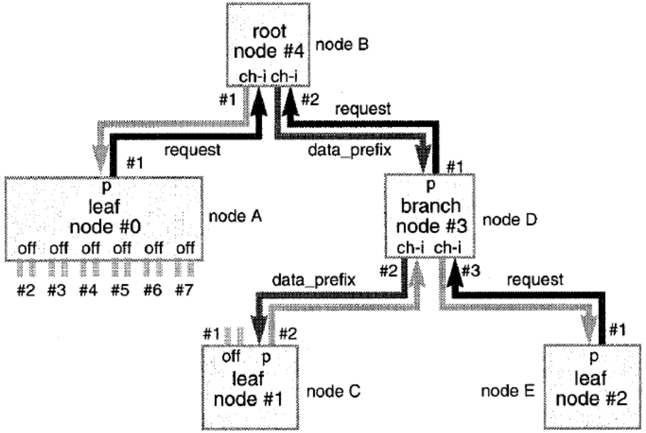
- Example:
  - Node A and E begin arbitrating at the same time by sending **request** to their parents

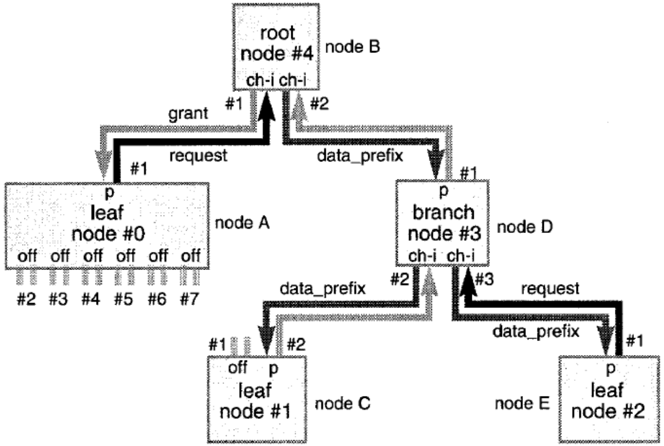


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Normal Arbitration

- Node D forwards the request and denies access to its other children by sending a *data_prefix*.
- At the same time, node B receives request from node A and denies access to its other children.
- Since node B is root, it does not forward the request further and grants access to Node A.

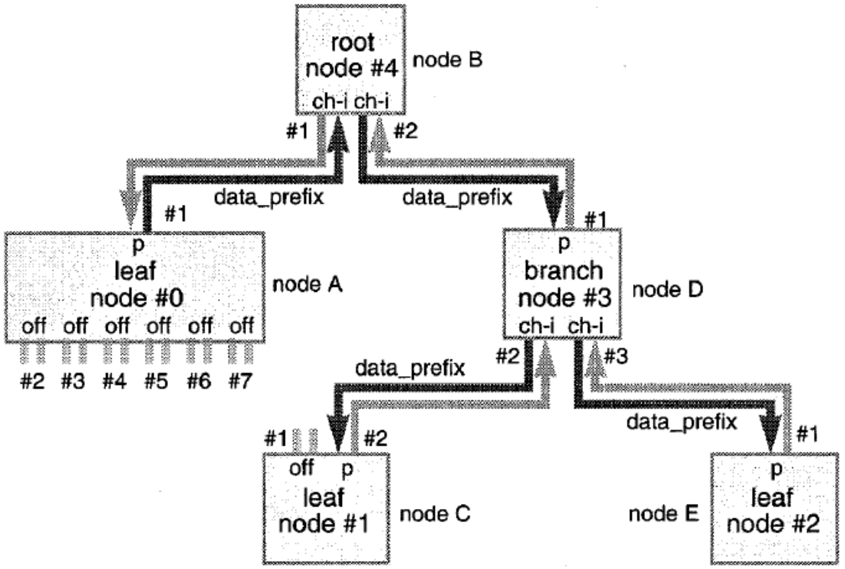




Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Normal Arbitration

- The granted node sends a *data_prefix* signal to all other nodes to warn other nodes that data is about to be sent.
- When the parent (in the example it is the root) receives this *data_prefix* signal, it stops sending the *grant*.
- At this point, the physical connections between all the nodes are now in the same state and pointed away from the node that won the arbitration.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Fair Arbitration

- The normal cable and backplane arbitration methods guarantee that **only one node will be transmitting at the end of the arbitration period.**

- These methods only provide a strict priority access; the node with the highest natural priority will always win.
  - Example: For cable environment, closest node to the root always wins

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Task 1: FireWire

Time Allocated 10 min

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233