IEEE-1394 A standard to interconnect distributed systems

Romualdo Santamaria

Mindready Solutions Inc.

2800 avenue Marie-Curie, Saint-Laurent QC, Canada H4S 2C2

Email: romualdo.santamaria@mindready.com

Abstract: With its high-speed deterministic communication, its low cost and its guaranteed bandwidth, the IEEE-1394 high-performance serial bus (also known as FireWire®) is ideal for interconnecting the components of modular distributed systems such as computers and I/Os. Mindready's paper will cover the following points: design of a typical industrial control system, heterogeneous computer systems and their limitations, isochronous data transmission, node synchronization, IEEE-1394 parameters and IEEE-1394 within the transport of digital data.

1. INTRODUCTION

Every so often, a new technology comes along that greatly simplifies the way to implement distributed systems. The IEEE-1394 high-performance serial bus (also known as FireWire®) is one of such technologies. Originally conceived by Apple Computer, IEEE-1394 has become the de facto serial bus for:

- Consumer electronics market
- PC peripherals
- Home automation systems

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35599-3_29](https://doi.org/10.1007/978-0-387-35599-3_29)

- Multimedia and entertainment systems

However, this new serial bus technology is also intended to revolutionize the way distributed system developers and system integrators design and build. With its high-speed deterministic communication, its low cost and its guaranteed bandwidth, IEEE-1394 is ideal for interconnecting the components of modular distributed systems such as computers and I/Os.

Today's modular systems - with their highly complex interconnection requirements - have challenged distributed systems developers with obstacles such as:

- Distributed shared memories and shared peripherals
- Synchronization between devices
- Controlled latency and jitter
- Topology restrictions
- ID switches and terminators
- Expensive cables and boards
- Connectors and bridges
- Bandwidth limitations
- Hot plugging and plug and play capabilities

2. THE DESIGN OF A TYPICAL INDUSTRIAL CONTROL SYSTEM

To appreciate the impact of using IEEE-1394 in distributed computing, let's examine the design of a typical distributed industrial control system.

A typical industrial control system usually requires:

- A network of computers capable of performing real-time calculations, based on analog or digital I/O input provided by one or more lower-level industrial computers
- A control console that allows an operator to monitor and control the entire distributed system

Let's use a car manufacturer as an example. This manufacturer needs to implement multiple robotic arms – all working in tandem on an assembly line.

A network of industrial computers – the arm's nerve center, controls each arm. These systems typically include several processor boards and I/O

boards and work as a distributed system. It repeatedly monitors the signals from the arm that encodes parameters such as:

- The current position of the arm
- The current speed of the arm
- Pressures being sensed by the arm
- Video images of the arm, and whatever it is handling

Based on the values of these parameters, it calculates corrections in real-time and periodically sends new values back to the arm, thus ensuring accurate execution of the command stream.

Needless to say, all the arms in this multiple-arm system must be perfectly synchronized, in real-time, to do their job. Data transfers and calculations must be performed within a specific time limit in order to guarantee the required system behavior. This can only be accomplished with a synchronized communication path — between all of the modules — that provides zero tolerance for error, as well as minimizes or controls latency and jitter. The slightest unpredicted “hiccup” in the communication system will adversely impact the behavior, with possibly catastrophic results.

3. HETEROGENEOUS COMPUTER SYSTEMS POSE SPECIAL PROBLEMS

Heterogeneous computer systems with different operating systems pose another challenge. An Ethernet connection can be used to provide a communication path between an industrial distributed computer system and the operator console – usually a Sun workstation (running on Solaris) or a PC (running on Windows NT)., This approach however, has several disadvantages. Ethernet is not deterministic and extensive development time might be required to ensure that the needs of the real-time system are satisfied.

Modular real-time systems often collect data from various types of computer hardware platforms. For example, the data entering in a system might be originating from any of the following subsystems:

- An IP (Industry pack) mezzanine board
- A PMC mezzanine board
- A VMEbus board
- A CompactPCI board
- A Personal Computer

- A SPARC workstation

If these subsystems are housed in multiple chassis, then bus-to-bus bridges are needed to interconnect them. While these bridges perform satisfactorily, they are expensive (typically \$5,000 per bridge) and this cost is further increased by the need to create protocol software to support communication between the subsystems.

In addition to the hardware and software costs of interconnecting heterogeneous platforms, all activities undertaken jointly by two or more platforms must be synchronized. This makes the development of distributed real-time applications a difficult task, even for the most experienced developers.

4. WHAT IEEE-1394 PROVIDES

IEEE-1394 provides a solution to these problems with:

- A scalable architecture
- A flexible peer-to-peer architecture
- A low cost implementation
- A high-speed deterministic mean of communication

The fact that IEEE-1394 is a bus that allows direct read/write access to memory in remote nodes is a major advantage allowing the systems to be a big pool of distributed memory and peripherals. However, it is the synchronized channel scheme of communication, added to the bus functionality that makes IEEE-1394 the perfect communication media for distributed real-time applications.

IEEE-1394 is a real bus with functionality similar to the PCI bus, which gives 1394 an advantage in comparison with communication links like Ethernet. 1394 does not have collisions and every asynchronous action is acknowledged signalling to the sender that the receiver has successfully accepted a message. Since there are typically several nodes on a IEEE-1394 bus, some arbitration method must be provided to determine which node has priority when multiple nodes are waiting to use the bus. This arbitration is provided by the Physical Layer, which is typically implemented in hardware.

The arbitration algorithm that is provided by the physical layer is called Fair Arbitration. This form of arbitration serves a very important function in real-time systems by ensuring that no single node is allowed to monopolize the bus.

The arbitration is controlled by idle periods on the wire. Depending on the length of idle periods, the nodes can determine the state of the bus.

The fairness protocol is based on the concept of fairness interval. A fairness interval consists of one or more periods of bus activity, separated by short idle periods called subaction gaps, and is followed by a longer idle period known as an arbitration reset gap. At the end of each subaction gap, bus arbitration is used to determine the next node to transmit.

When using fair arbitration, an active node can initiate sending a packet once in each fairness interval. Once a node sends in a fairness interval, an internal arbitration flag is cleared disabling the node from arbitrating again during the same interval. Once all the nodes who need the bus have successfully won arbitration and been disabled, an arbitration reset gap occurs. This ends the fairness interval. At that point, the internal arbitration flag of all the nodes is reset, a new fairness interval begins, and all nodes are enabled to arbitrate again.

5. ISOCHRONOUS DATA TRANSMISSION

IEEE-1394 also provides a *synchronous* form of communication, called *isochronous* data transmission. Isochronous data transmissions are given priority over *asynchronous* data transmissions.

Isochronous data transmissions are initiated once every 125 μ secs by the cycle master, which transmits a special *cycle start packet* to trigger all the waiting isochronous data transmissions from other nodes. After transmitting this special packet, all nodes waiting to send isochronous packets start arbitrating for transmission until all nodes finish sending their pending isochronous packets.

All isochronous packets have to be transmitted within the 100 μ sec in the 125 μ sec period.

If the transmission of isochronous packets takes less than 100 μ sec, then the cycle master accepts asynchronous packets for the remainder of the cycle.

Isochronous transmission was originally included in the IEEE-1394 standard to support multimedia applications - such as a live video feed - where uninterrupted transport of time-critical data and just-in-time delivery reduce the need for costly buffering. Fortunately, isochronous transmission is also ideal for interconnecting real-time computer systems because it provides automatic synchronization between the local clocks in the various nodes and guarantees a predetermined communication bandwidth without extra programming.

6. NODE SYNCHRONIZATION

The cycle master (driven by its own internal 8 kHz clock) attempts to start a new cycle every 125 μ secs. However, the beginning of any particular cycle might be delayed slightly if an asynchronous packet is still being sent when the time for the new cycle arrives. Thus, there might be some "jitter". However, the *average* cycle time will still be 125 μ secs. All nodes supporting isochronous communications have a timer running at 24.576 MHz. Each time a cycle starts, the cycle master node writes the value of its internal timer into the timer of all other nodes. This provides synchronization between all nodes. The cycle start message provides the timing information if a particular cycle start is delayed, this way all nodes know by how long, and can therefore manage and control resulting jitter.

7. IEEE-1394 PARAMETERS

There are parameter values that must be specified locally in each node before IEEE-1394 packets can be exchanged between nodes. These parameters include the IEEE-1394 node number, IEEE-1394 bus number, and IEEE-1394 response flag.

The IEEE-1394 node number

A node number is a unique number (between 0 and 63) that is associated with each node on the bus. This number is dynamically re-configurable, allowing the software in each node to change its node number at any time. Node number 63 is used to broadcast within a single 1394 bus.

The IEEE-1394 bus number

The IEEE-1394 bus number allows nodes on different IEEE-1394 buses to communicate with each other. Each IEEE-1394 bus is assigned a unique

number between 0 and 1023. Bus number 1023 is the local bus for each 1394 bus.

The IEEE-1394 response flag

The IEEE-1394 response flag tells each node in the system whether it must send a 'response packet' for every read request packet, or write request packet that it receives.

8. IEEE-1394 REVOLUTIONIZES THE TRANSPORT OF DIGITAL DATA

In a distributed industrial control application (such as the robot controller mentioned earlier) the developer can use IEEE-1394's isochronous transmission to:

- Synchronize the timers of all nodes on the bus with the cycle start timer of the master node
- Guarantee data delivery
- Guarantee on-time delivery
- Guarantee data transmission at a pre-determined minimum rate

This is all accomplished without any extra programming. This way, isochronous transmission reduces latency and controls jitter, and the data exchange intervals are controlled natively on the bus, with no extra programming.

The developer can interconnect up to 63 devices on each IEEE-1394 bus in a peer-to-peer fashion, communicating digitally at up to 400 Mbps. This peer-to-peer communication allows each node's internal address to be mapped in the local address map of each other node on the same IEEE-1394 bus.

The result is a much simpler integration of multiple heterogeneous computer platforms, which is often necessary to provide enough processing power for today's demanding real-time applications.

In addition, no expensive bridges are required. All nodes can exchange data over the IEEE-1394 bus by simple read and write operations. For example, suppose an IP (Industry Pack) mezzanine board is plugged into a mezzanine site on the IEEE-1394-to-VMEbus adapter board. Its I/O registers

would then be mapped to the local address space of each other node on the IEEE-1394 bus (such as a SUN workstation or another VMEbus rack) without any special software drivers or protocols, or any special interface hardware.

This eliminates the need for any dedicated bridges, and simplifies the integration of different devices in real-time systems. The design time, the project costs are significantly reduced, and the versatility of the resulting distributed real-time system is increased. Systems can also be upgraded and/or expanded very easily, reducing concern about what additional equipment might be needed at some future time.

The IEEE-1394 serial bus allows for all communications to be performed over thin, easy-to-handle 6 conductor cables. Cable length is limited to 4.5 meters by the arbitration mechanism and the signal propagation delay. However, high quality cabling and/or repeaters can be used to extend that distance.

IEEE-1394's flexible cabling topology allows for much easier installation than standards requiring rigid cabling topologies, such as point-to-point and hubbed architectures. Tree and daisy chain topologies are supported. As a practical matter, the topologies are limited only by the number of ports on the bus-switching chip, which can be up to 6 ports.

Finally, IEEE-1394 interfaces are designed to allow hot plug and play. For example, suppose one of the robotic arms in the distributed system is in need of service – possibly even removal and replacement. With IEEE-1394, the manufacturing process can continue uninterrupted while this is being done. There would be no need to shut down the entire production line just to add or remove that one node.

9. SUMMARY

Without a doubt, IEEE-1394 is revolutionizing the way data is being transported in distributed real-time systems, and as such, is changing the way developers and systems integrators design, build and interconnect these systems.

The IEEE-1394 specification currently supports speed of up to of 400 Mbps, and is expected to reach speed of up to 3.2 Gbps with the recent advent of 1394b, an enhanced version of 1394. This provides an easy evolutionary path—without any changes in the application code—for users

who are currently developing real-time applications with ever-increasing bandwidth requirements. Because IEEE-1394 is rooted in mass-market solutions, chips and cables are widely available at a low cost. These same low-cost components can be used in IEEE-1394-based real-time systems.

Currently deployed particle acceleration and flight simulation systems have demonstrated that IEEE-1394 provides:

- An easily scalable architecture
- A flexible peer-to-peer topology
- A low cost implementation
- High-speed, deterministic communication

Based on this demonstrated performance and relatively low cost, IEEE-1394 is poised to become the de facto standard for interconnecting heterogeneous real-time computer systems.