

# Combining AS5643 and IEEE 1394b – the Rise of the Machine

By Kenneth Sanders, Lockheed Martin (ret.) and Richard Mourn, FlightWire Technology

For years, IEEE-1394 (FireWire) has been a successful standard, however it wasn't until IEEE-1394b-2002 and its key new features such as allowing looped topologies and 8b10b encoded data, that when coupled with SAE Standard AS5643 created a deterministic, robust, and redundant system architecture suitable for aerospace and defense type applications. The combination of the two open standards meets jitter, data latency and data coherency requirements for a hard real-time networks such as those needed in an aerospace vehicle management, avionics, and mission or associated display systems. Standard AS5643 has been written and developed without much context given as to how these system types have evolved, matured and have been qualified for their various applications, and the evolutionary paths taken by these systems differed from 1394's path. But when they converged, the combination has benefited the vehicles themselves, along with enabling new approaches to test equipment design and vehicle maintenance.

Several steps have been taken to reach the current development state of these systems and a standard such as AS5643 can support and enable new ones.

## **How the vehicle system embedded control architecture began; data latency vs coherency**

Because embedded computing started as analog systems with a CPU stuck in the middle, data conversion was always an issue, time-consuming and riddled with data latency challenges. Along with standard dynamic compensation techniques, data latency was compensated for by over sampling existing analog signals by some integer multiple of the primary processor iteration rate. Regardless of the approach used and the complexities introduced because of those approaches, some data was always older than desired. When these control complexities were combined with physical redundancy, additional issues arose with data coherency and time skew between the physical channels. Physical and dissimilar redundancy added another layer of complexity to safety-critical embedded systems when trying to solve the problem of interchannel differences that occur due to data coherency issues.

There is a clear distinction between data latency and data coherency in these types of systems. Data latency is introduced by the "self" channel's introduction of time delay when processing its own inputs and outputs and forming its own view of the world and data coherency is the time delay introduced by the other branches (e.g. right, left, and/or opposite) or other subsystems, as the case may be, where that data is primarily used for comparative purposes to detect and isolate failures within the system. Over time, there were primarily three methods for controlling the magnitudes of data coherency. First was the use of some centralized timing mechanism whereby all branches would slave their internal clocks and stay in lock step with one another. While there are systems that operate this way today, overall, this method is not preferred. The introduction of failure modes in the timing mechanisms themselves can lead to common mode failures and even if those failure modes can be handled, the failure of the centralized timing mechanism leads the system to asynchronous operation, which has to be safely adjudicated.

Second was the attempt at operating a redundant system totally asynchronously. Given the iteration rate of the system and some other factors, the net effect was that interchannel differences could be so large as to require very wide tolerances on voting algorithm trip levels. The result in system operation was to allow failures to persist longer than desired and, depending on the platform, this persistence could be catastrophic. Again, there are systems that operate this way today also. Designers overcame some of the persistence issues described by acknowledging that certain events either external to the system (e.g. a configuration change) or internal to the system (e.g. a mode change) needed to be aligned and recognized by all redundant branches before subsequent actions are taken. For the purposes of this article, that is called “event” synchronization. Given this past experience, it is clear that a practical redundant system can either be time synchronous or event synchronous (or some hybrid of either) but not exclusively asynchronous. This leads to the third way of handling data coherency issues by defining the beginning of each iteration cycle as an “event” and allowing the majority of branches to recognize that “event” before continuing with the next frame’s processing. This approach does not synchronize the branches perfectly since independent clocks are still involved in determining the “event” but can achieve time synchronization within a few microseconds depending on the transmission rates or method of interchannel communication. While the details of how this is accomplished is proprietary to some systems, the concept of defining a “Top of Frame” (TOF) is important, and is discussed below.

## **SIDEBAR**

### **What was the initial role of 1553;**

The introduction of serial data links to safety critical redundant systems occurred for two reasons. First, there was a need to provide cross-channel data communications primarily to support redundancy management functions - and analog interfacing became impractical. These interfaces remained dedicated point-to-point serial links - as with 1394 - for almost 30 years. Second, there was a need to connect this uniquely redundant system to the outside world, to implement outer loop controls, mission management systems, display systems and improved sensors. The evolution of these type systems occurred at a faster pace than embedded control systems and for most of the 1980s and 1990s the interface of choice for military applications was MIL-STD-1553.

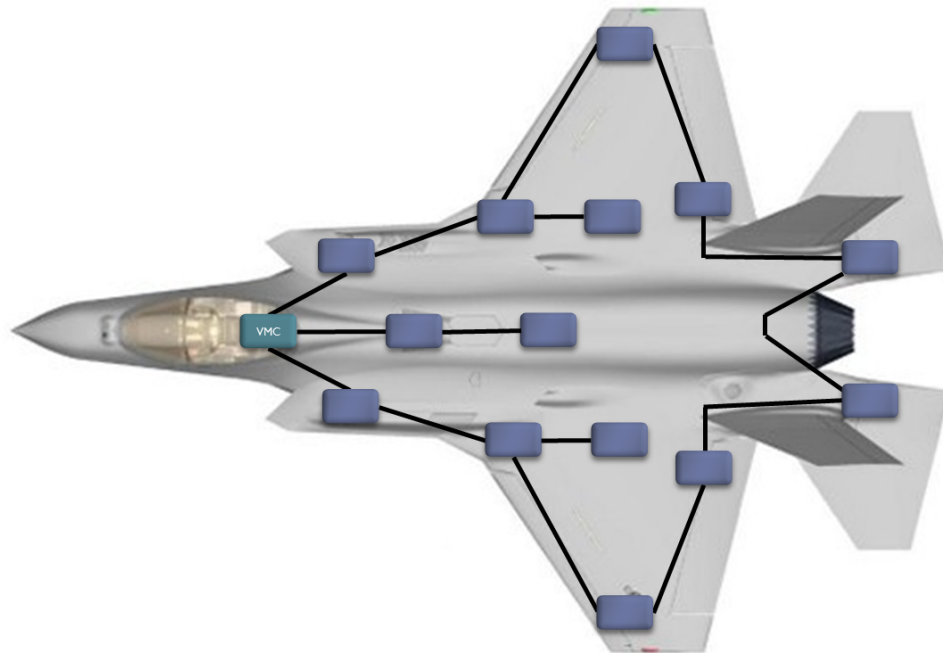
These two paths of system development could not be more different. Their primary computational rates were driven by entirely different considerations and the results were not only different but were not even integer multiples of one another. Tying these eclectic systems to a redundant system that needed predictable time domain data was a complex integration task. The roadblocks to a successful implementation did not end there. A system designer might try to initiate data transfer at a point where the data was needed but had no idea where in the iteration cycle the redundant system was nor had any idea if the branch being communicated with had the primary responsibility for driving the system’s outputs. Of note concerning the

issues with data latency and data coherency was the operation of 1553 as a network. As a command-response system, a centralized bus controller initiated all data communications in a manner that is best described as “send the best data you have now to the following location”. The bus controller had no clue where the sending system was in its cycle nor did it know where the receiving system was in its cycle. The end result was not only increased data latency but latency that was not consistent over time adding beat frequencies to the uncertainty. In many cases, for analysis purposes, the best a designer could count on determining was the average latency over a given period of time or a statistical range of latency values. The mathematical approaches for analysis of linear control design falls apart in these circumstances. Without being too pessimistic, please note that systems have been made to successfully operate this way for a very long time. For instance, navigation systems wanting to incorporate GPS found that time stamping everything and then extrapolating the data to the time frame required of the integration steps works fine. There are some very impressive and expensive Kalman filters to show for it. In this kind of application, a sensor is essentially being slaved to a model that is using other sensor inputs to produce a navigation solution. This is not the same as providing closed loop control of an unstable or at least neutrally stable plant. There are always unwanted system limitations that result when attempting to use a similar approach as the navigation solution such as limited bandwidth, reduced phase and gain margins or reduced robustness to outside stimulus. The cost of analyzing and testing such closed loop systems becomes increasing prohibitive as well, because of the multitude of time variations that can exist. Imagine what happens when someone has the brilliant idea to standardize all safety critical communications over 1553. This happened for a proposed system design for the A-12, an aircraft program which was cancelled in early 1990. One big issue: getting data to the right place at the right time was problematic using 1553 for time critical communications. The approach taken to address this issue with an exclusive 1553 network architecture was to leave certain critical signal paths in the analog domain or on a separate independent serial data path, proliferating the number of networks needed. Examples include platform state sensors, controller inputs, interchannel communication and actuation outputs. What became a measurable constant was the amount of time needed for a change in input to effect a change at the output. The net effect of this approach to system design was to limit architecture designs from becoming fully federated. This effect was also seen in areas such as electrical power and hydraulic systems.

### **Benefits of 1394b and AS5643**

Trade studies were conducted early in the F-35 program that attempted to balance a number of considerations to determine a best fit solution for serial data networks within a vehicle management system with the idea of creating a fully federated system. In this type of system, the controlling electronics is placed as close to the application as possible. This desire is especially appealing for “more electric” technologies, which tend to require high bandwidths and computational iteration rates. The net effect is better control efficiency and more affordable upgrade paths for electrohydraulic actuation, high voltage dc power generation and control, environmental systems control and engine or propulsion control.

Another desired effect of federation is minimizing wire weight and reducing overall wire volume. Therefore, the topology supported by 1394b and formalized for a redundant safety critical system in AS5643 typically consists of two types of nodes; a Control Computer (aka VMC) node, located at the choice of the system designer and Remote Nodes (aka LRU), which can refer to any desired subsystem that implements the architecture. Control Computer and Remote Nodes may be connected to optimize the subsystem's location and implementation requirements for weight, volume and redundancy. 1394b allows the system architect to optimize for both weight and redundancy by supporting a mixture of daisy chaining, tree, loop and star topology configurations without the need for centralized switching. 1394's flexible cable topology support allows the system implementer to optimize cable routing to meet weight/volume and robustness requirements and, unlike star or switched technologies homerun cabling is not required.



*Figure 1: Flexible 1394b bus topology*

The next definitive attribute of 1394b is that it may be implemented using industry standard 8b10b SERDES with LVDS-type differential signaling. Both are industry standard for serial I/O and therefore power/performance optimized SERDES implementations are available in both standard silicon and many FPGA families. Currently almost all aerospace and defense applications use standard COTS 1394 silicon with all the cost and ecosystem benefits that implies; development and production test tools/equipment already exists, multiple software stacks operating under multiple OSs are widely available and, if desired, FPGA PHY and Link IP cores are also available. Additionally, the digital logic in most S400 PHY and Link designs operate at less than 50MHz, which reduce switching currents and help make 1394 Beta FPGA friendly. Beyond the COTS synergy benefits, AS5643 makes use of 8b10b SERDES to allow

implementation of both passive and active transformer coupling to be used with standard 1394b. These transformers provide the electrical isolation necessary to meet stringent common mode noise rejection requirements and requirements such as RTCA/DO-160 lightning susceptibility. The active transformers boost the guaranteed minimum differential signal amplitude from 600mV(P-P) to 1100mV(P-P) allowing cable lengths of greater than 15 meters at 500Mb/s to be achieved with a desired degree of signal to noise margin.

As this article started out talking about, data latency and data coherency, probably the most important and versatile attribute of 1394b and the implementation defined by AS5643 protocol defines a time synchronization methodology and Anonymous Subscriber Messaging (ASM) protocol. The basic attribute of 1394b is Asynchronous Streaming capability coupled with the concept of defining a "Top of Frame" (TOF) event as implemented in AS5643. Asynchronous Stream packets are a very attractive feature, especially as a device-to-host notification mechanism, because the device need not know the current physical ID of the Control Computer. This idea has huge implications to implementing system redundancy and providing multiple transmission paths after a network reconfiguration. Time synchronization is accomplished by the Control Computer sending out Start of Frame (STOF) packets at the network profile specific frame rate (typically 80Hz or 100Hz). All Remote Nodes receive the STOF packets and base the assigned offset times to the received STOF packets. The ASM messages are transmitted at the network profile assigned offset times relative to the STOF. This allows the receiving node to deterministically anticipate and therefore guarantee resources are available to process the received messages. The idea is to provide all remote nodes with a view of the overall iteration cycle of the control computer, align itself to that frame in time and provide future windows (identify when data is needed) for when data should be made available with the least amount of data latency and the maximum amount of coherency. Setting up an Anonymous Subscriber Messaging protocol simply means that any remote node needing that data can simply subscribe to the message (a priori) and the data only need to be transmitted once in a given frame.

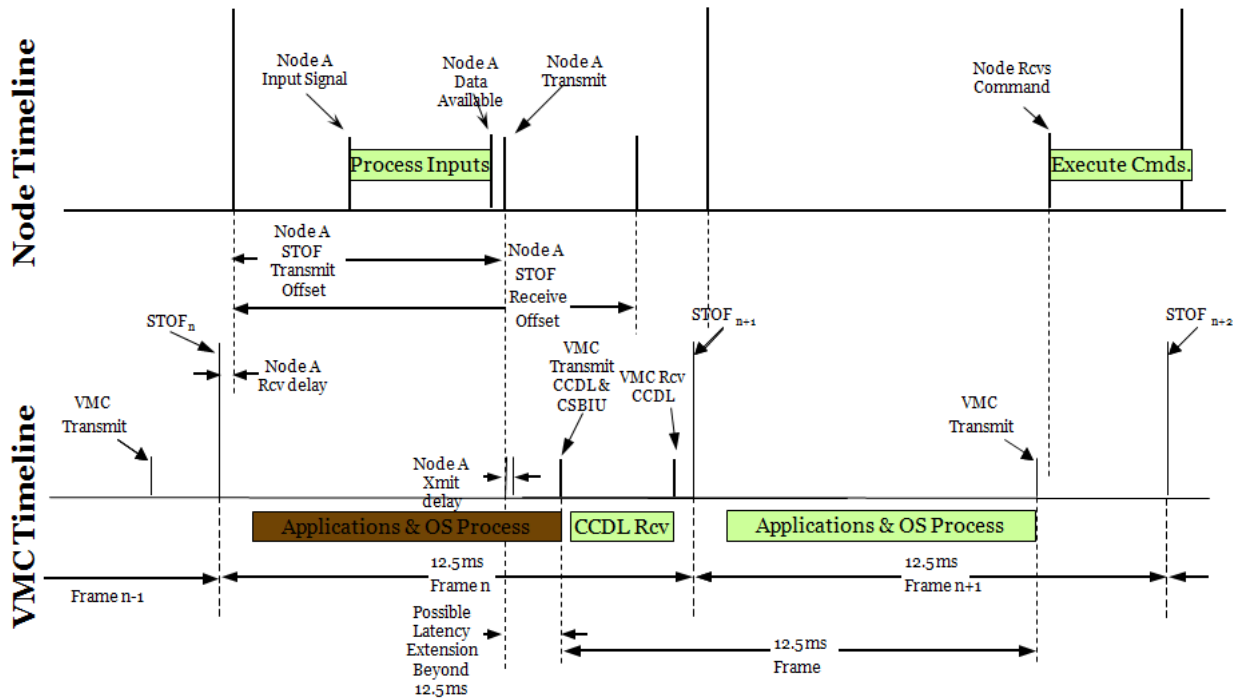


Figure 2: AS5643 system architected deterministic timing diagram

### What improvements are needed?

1394b does have some shortcomings, which arm the system designer with real and useful information instead of the “unknown unknowns” still lurking in the application of other popular serial communication protocols. The improvements cited below are aimed at the ability to test, maintain and verify installed correct network operation, not at changing the off the shelf IEEE-1394 standard capabilities that would make the resulting system non-standard. In fact, the standard capabilities of IEEE-1394 implemented in accordance with AS5643 have provided unprecedented speed, flexibility, reliability and robust redundant operation.

If there was ever a shortcoming of the defined trade space used to initially select a serial network communications protocol for the F-35, it is the omission of support equipment for and diagnostic capabilities of the product. Likely as not, however, if this attribute had been included in the original trade studies, no candidate would have been selected. In fact, if the same trade were repeated today, most of the considered candidates would still fail for this very reason.

IEEE-1394, having originally been designed for computer and consumer electronics markets, could be improved to better meet AS5643 based application requirements. Most improvements fall into the category of diagnostics and robustness.

### Diagnostics and Robustness

Determining the location of connectivity issues on a buttoned up aircraft is much more difficult and important than determining similar issues in a computer desktop or consumer electronics

environment. In addition, the harshness of the aerospace environment with temperature cycling, vibration, shock, intense EMI, etc... accentuate effects of non-continuous cable shielding, improperly seated connectors and back shells, improperly terminated cables into pins or sockets, inconsistent or no twists within a differential pair that years of static use in a climate controlled environment would never reveal.

Currently 1394b only provides a few of features that can be used to automate the diagnostic process used to locate connectivity issues. The 1394b PHY can tell you if it initiated a bus reset, the connectivity status of each of its ports and if the port received any invalid 8b10b symbols. Of the three, initiated bus reset and port connection can be captured by network diagnostics by simply monitoring self-ID packets that are automatically sent by each PHY layer after a bus reset. To help with diagnostics, the SAE AS-1A3 Mil-1394 Task Group is currently updating AS5643 to include the port detection of an invalid character in the health status of each anonymous subscriber message (ASM) sent by each node.

Additionally, the Air Force SBIR AF151-060, "Common Embedded Vehicle Network Diagnostics Interface Hardware" was issued to help maintenance personnel isolate subsystem wiring, or connector faults and increase robustness by optimizing the bus initialization process. Through this SBIR perhaps signal to noise margin can be better understood using advanced I/O features such as built in eye diagram and loopback tests. Also, the ability to measure and record at the Tx and Rx the dc line voltages on which the data resides at the cable side of the transceiver is one of the easiest and most reliable ways of isolating transceiver failures. It also, provides an alternative to BER tests, which are meaningless in most troubleshooting, circumstances. In addition, a signal to noise ratio margin can be determined.

There are a number of reasons why a 1394b PHY could lose synchronization on one of its ports. While the standard allows for multiple methods of recovery, most implementations simply disconnect and try to reconnect. The disconnected to connection active process takes hundreds of milli-seconds. While the 1394b looped topology allows for the bus to heal itself in approximately 2msec, using the proper method the reconnect time can be reduced to hundreds of  $\mu$ sec thus increasing the robustness of each connection. Increased robustness can also be achieved by simply simplifying the 1394 PHY and Link Layers to only implement the IEEE-1394 features required for AS5643 applications. By removing unused features both the PHY and Link become simpler in functionality and to test.

### **Bus Architecture – Good or Bad?**

1394's bus network topology has many advantages, however one inherent disadvantage is the propagation of bus resets. Under normal operation conditions when power-up and initialization are complete, no bus resets should be generated. Unfortunately we don't live in an ideal world so error conditions do occur and bus resets are generated. Connection or disconnection of a port due to power cycling, connection of new devices (test equipment) and intermittent

electrical connections (bad cable) can all cause bus resets. Marginal system implementations with low signal to noise ratio, EMI coupling on to the signals and/or ground or power supplies can also cause bus resets. The 1394 PHY layer can also generate bus resets when certain timeouts occur and lastly software initiated bus resets which should be limited to only a few cases generate bus resets. No matter the cause, bus resets propagate throughout the topology and disrupt communication on the 1394 network. Additionally bus resets require some “clean-up”. AS5643 limits the amount of clean-up by using channel numbers to address packets. While no device discovery is required, most implementations still require the transmitter to flush asynchronous transmit packets in the queue at the time of the bus reset. This causes software to reinitialize the queue before packets can be sent. As with any network architecture, error handling is required and 1394 is not different. So while the 1394 bus architecture allows bus resets to propagate, with proper error handling the effect can be minimized and the required quality of service can be maintained.

While propagation of bus resets can be problematic at times, the bus architecture allows excellent observability when compared to switched networks. Diagnostic equipment or even the control computer can be connected at one location on the network and still record all communication or just specific events from the complete network. This artifact of a network architecture enables unprecedented diagnostic capability on large and complex networks.

## **Conclusion**

The convergence of IEEE-1394 and AS5643 non-proprietary standards has created a powerful deterministic, robust, and redundant system architecture suitable for aerospace and defense type applications. Unlike its predecessors, such as 1553, 1394/5643 allow a fully federated system architecture that meet today’s vehicle management system requirements.

The federated system architecture allows control electronics to be placed as close as possible to the application which is desirable for 5<sup>th</sup> and 6<sup>th</sup> generation jets, such as the F-35, that tend to require high bandwidths and faster computational iteration rates. The net effect is better control efficiency and more affordable upgrade paths for all vehicle management bus modules. In addition, 1394/5643 based federated system architectures minimize wire weight and reduce overall wire volume by eliminating home run wiring with shorter LRU to LRU connectivity. The net effect is 1394/5643 provides the data latency, data coherency and efficiency along with the redundancy and robustness required for future manned and unmanned aircraft.