
AFRL-RZ-WP-TP-2008-2186

# JET ENGINE CONTROL USING ETHERNET WITH A BRAIN (POSTPRINT)

**Brendan Hall, Michael Paulitsch, Dewey Benson, and Alireza Behbahani**

**Structures and Controls Branch**
**Turbine Engine Division**

**JULY 2008**

**Approved for public release; distribution unlimited.**

***See additional restrictions described on inside pages***

**STINFO COPY**

**AIR FORCE RESEARCH LABORATORY**
**PROPULSION DIRECTORATE**
**WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7251**
**AIR FORCE MATERIEL COMMAND**
**UNITED STATES AIR FORCE**

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS**.

| 1. REPORT DATE *(DD-MM-YY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| July 2008 | Conference Paper Postprint | 02 January 2008 – 21 July 2008 |

**4. TITLE AND SUBTITLE**
JET ENGINE CONTROL USING ETHERNET WITH A BRAIN (POSTPRINT)

**5a. CONTRACT NUMBER**
In-house

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
62203F

**6. AUTHOR(S)**
Brendan Hall and Michael Paulitsch (Honeywell Advanced Technology, MN)
Dewey Benson (Honeywell Advanced Technology, AZ)
Alireza Behbahani (AFRL/RZTS)

**5d. PROJECT NUMBER**
3066

**5e. TASK NUMBER**
03

**5f. WORK UNIT NUMBER**
306603TM

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Honeywell Advanced Technology
Golden Valley, MN 55422
---
Honeywell Advanced Technology
Tempe, AZ 85284

Structures and Controls Branch (AFRL/RZTS)
Turbine Engine Division
Air Force Research Laboratory, Propulsion Directorate
Wright-Patterson Air Force Base, OH 45433-7251
Air Force Materiel Command, United States Air Force

**8. PERFORMING ORGANIZATION REPORT NUMBER**
AFRL-RZ-WP-TP-2008-2186

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Research Laboratory
Propulsion Directorate
Wright-Patterson Air Force Base, OH 45433-7251
Air Force Materiel Command
United States Air Force

**10. SPONSORING/MONITORING AGENCY ACRONYM(S)**
AFRL/RZTS

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)**
AFRL-RZ-WP-TP-2008-2186

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited.

**13. SUPPLEMENTARY NOTES**
Conference paper presented at the 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 21 –23 July 2008, in Hartford, CT. The PowerPoint slideshow presented at the conference is an attachment within the digital Adobe Acrobat .pdf report file.

PAO Case Number: WPAFB 08-3731; Clearance Date: 17 Jun 2008. The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work.

**14. ABSTRACT**
Distributed control architectures are becoming increasingly prevalent as smart actuation and sensing technology becomes more cost-effective and realizable. However, achieving a distributed architecture that supports the increasing computational demands of engine control and prognostics strategies whilst surviving in the harsh on-engine environment remains to be a significant challenge. In this paper we present a Hybrid Ethernet based architecture that combines gigabit per second capacity for computational agreement and replication, with low complexity fault-tolerant Ethernet based braided ring segments for robust on-engine message distribution. To establish a rationale for the architecture the communications requirements for distributed control are briefly summarized and a discussion of current and emerging communications technologies is also presented. The BRAIN (Braided Ring Availability Integrity Network) dependability augmentation strategies are then presented to illustrate how the dependability issues of current communications may be mitigated.

**15. SUBJECT TERMS**
BRAIN, Braided Ring Availability Integrity Network, Gas turbine, FADEC, disturbed based control applications, Communication, Distributed Control System Architecture, MAC Architecture, DCS, TTP, FlexRay, Intellibus, IEEE-1394/SAE 5643, ARINC-664, TTEthernet

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT: | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON (Monitor) |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Alireza R. Behbahani |
| Unclassified | Unclassified | Unclassified | SAR | 24 | 19b. TELEPHONE NUMBER *(Include Area Code)* N/A |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std. Z39-18

# Jet Engine Control Using Ethernet with A BRAIN

Brendan Hall[1] and Michael Paulitsch[2]
*Honeywell Advanced Technology, Golden Valley, Minnesota, 55422*

Dewey Benson[3]
*Honeywell Advanced Technology, Tempe, Arizona, 85284*

*and*

Alireza Behbahani[4]
*Air Force Research Laboratory, Wright-Patterson, AFB, Ohio, 45433*

**Distributed control architectures are becoming increasingly prevalent as smart actuation and sensing technology becomes more cost-effective and realizable. However, achieving a distributed architecture that supports the increasing computational demands of engine control and prognostics strategies whilst surviving in the harsh on-engine environment remains to be a significant challenge. In this paper we present a Hybrid Ethernet based architecture that combines gigabit per second capacity for computational agreement and replication, with low complexity fault-tolerant Ethernet based braided ring segments for robust on-engine message distribution. To establish a rationale for the architecture the communications requirements for distributed control are briefly summarized and a discussion of current and emerging communications technologies is also presented. The BRAIN (Braided Ring Availability Integrity Network) dependability augmentation strategies are then presented to illustrate how the dependability issues of current communications may be mitigated. The example architecture presenting an Ethernet based hybrid solution is then discussed, in relation to superior dependability, performance and life-time cost optimization it can offer in contrast to current solutions.**

## A Introduction

Current state of the art engine controls have converged on the notion of the Full Authority Digital Engine Control (FADEC), which consists of a centralized controller with two independent channels to provide redundancy and improved availability. As 'Full Authority' implies, the operation of the engine is completely dependent on the proper operation of the controller. In current systems, the FADEC is often located on the relatively cool engine fan case to allow use of conventional electronics or is fuel cooled if located more centrally on the engine, the later approach being more costly due to complexity of the controller enclosure.

One of the big challenges of the all-in-one approach involves development and life cycle costs. The typical FADEC is optimized for a particular engine, which limits application-to-application re-use. Each new application is often a 'clean sheet' design. It also means that any obsolescence issues often have to be handled by a major redesign of the controller. New features can only be added during a major redesign effort. The many unique designs mean no commonality, costly spares provisioning, no recurring cost leverage, and limited opportunity for technology insertion.

---

1 Staff Engineer, Honeywell Advanced Technology Platform Systems.
2 Senior Engineer, Honeywell Advanced Technology Platform Systems
3 Fellow, Honeywell Advanced Technology.
4 Senior Aerospace Engineer, Turbine Engine Division, Propulsion Directorate, AFRL/RZTS, Senior Member.

It has long been hypothesized that considerable savings in development cost, weight, and life cycle cost could be realized, if a distributed architecture were implemented. To date, these perceived savings have yet to overcome the considerable hurdles to implementation of such an architecture. Beside the extreme temperature environment on some parts of a jet engine, no communications technology has yet been adopted by which multiple suppliers can contribute engine components (e.g. sensors, actuators, etc) that are interoperable over a common bus interface.

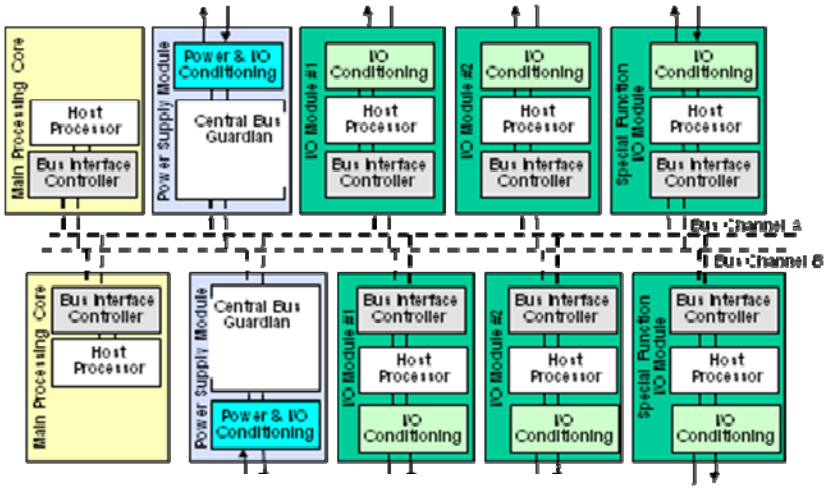


*Figure 1: Generation 1 MAC Architecture*

Advances in electronics and fault-tolerant communication technology have helped to achieve greater functionality at reasonable cost as exemplified by the Modular Aerospace Controls (MAC) architecture developed by Honeywell in 2000 (see Figure 1). A Modular architecture based on the COTS TTP protocol, MAC enabled modular re-use within the FADEC system boundary, allowing engine customization to be achieved via the selection of generic modules. Initially targeted at three engines, the re-use of the MAC architecture has been very promising in this regard. The architecture modularity and systematic redundancy management, leveraging the composability and determinism of the time-triggered TTP protocol has also been demonstrated to significantly reduce non-recurring engineering expense and design complexity. However more radical measures will be necessary to achieve significant reductions in the future. The implementation of intelligent propulsion concepts requires advancements in the area of robust distributed control synthesis techniques with embedded systems, automated diagnostics, and development of advanced enabling technologies such as smart sensors and actuators. Concepts integrating distributed sensing, actuation and control logic will impact performance and the environment for micro-level control of parameters for a fault-tolerant turbine engine of the future.

These challenges involve the optimization of performance and improving reliability in the face of constraints on communication bandwidth, congestion, and contention for communication resources, delay, jitter, noise, fading, and the management of signal transmission power. Taking a broad view of distributed networked intelligent control systems, we find that in addition to the challenges of meeting real-time demands in controlling data flow through various feedback paths in the network, there are complexities associated with mobility and the constantly changing relative positions of intelligent nodes which are connected to sensors and actuators and other embedded systems in the network. In designing a control system for turbine engines there are many application-specific challenges that will be encountered in trying to implement a control system in addition to those mentioned above. Some of these challenges are design specific and some are hierarchical distributed control system related. These challenges includes system timing issues where different portions of the overall system may be operating at different sampling times and in synchronous or asynchronous modes. There may be many control inputs and sensed outputs and interactions between spatially distributed parts of the plant.

However, in other industries Distributed Control Systems (DCS) have been shown to be a reliable method since several decades ago. In addition in the 1980's through 1990's, DCS has been suggested and demonstrated by several technologies in aerospace applications. The US Air Force has been funding several programs related to distributed control to increase the technology levels for turbo engines. However, many technology maturations have been taking place in other fields. Several trade studies have suggested that DCS is becoming more attractive in aerospace applications for lower cost and obsolescence issues. Application of DCS for turbine engines requires additional technology maturation to make DCS more practical, reliable, and less expensive. DCS requires drastic change in the design process as well as supply chain changes in Tier 1 and Tier 2. Justifications for use of DCS to take advantage of the benefits offered by DCS needs to be proven and a cost/benefit analysis must be performed. prior to

implementation. DCS does not imply a specific architecture; instead a clustering of different system elements can be arranged in any configuration that best *maximizes customer value*. That said, distributed control architectures, drawing hardware nodes from pre-existing programs, permit re-combining into a newly capable system to minimize cost. Coupled with control strategy tools and techniques, the re-distribution of hardware and reuse of software components is virtually transparent. In addition, the application of distributed control to multiple applications, in both hardware and software, benefits the prototyping phase with reduced schedule, cost, and risk. Therefore such a strategy offers promise to meet the challenges of the industry as shown in Table 1.

Table 1: DCS Controls Architecture and Challenges

| Requirement | Effect |
|---|---|
| Improved performance in aircraft | Tighter coupling of controls, distributed engine controls |
| Reduced fuel consumption | Lighter controls and monitors, smaller more restricted enclosures |
| Removal of cooling apparatus | Hotter operation of electronics without airflow or heat sinking |

However, in this paper, we are not trying to justify the use of DSC for future applications. DCS needs to develop ranges of architectural changes appropriate for specific applications. Hardware and software adapted for DCS needs to be designed and be implemented. These technologies require many challenges from sensors, communication, protocols, cost, performance, actuation systems, processing capability, and higher temperature components capability. The need for high temperature electronics for control system in the turbine engine is not new. There are a number of companies including Honeywell engaged in manufacturing of these components. Additionally, a number of recent requirement changes have pushed the upper operating temperature range limit higher [Goe98] (see Table 1).

The focus of this paper is the communications architecture required to support the DCS. From the lessons learned on MAC, and other demonstrators, the attributes and properties of the communications system are a key enabler for cost effective system realization. Modular distributed real-time engine control architectures will require open communication infrastructure that will be able to satisfy all the engine control architectural needs with added functions of increasing sophistication and multiple levels of control. Communications architectures for real-time high-speed, large-volume data transfer are required to provide modular control systems connectivity with low latency and high reliability for safety critical components. We therefore start this paper with a review of the requirements for the communications system within the distributed control system. We then review some current and emerging communications technologies with respect to their strengths and weaknesses in light of these requirements. Next we present a dependability augmentation strategy based on a Braided Ring topology, called BRAIN. Finally, we illustrate how a hybrid system architecture may be implemented integrating the BRAIN for on-engine distributed control and high-performance time-triggered switched Ethernets for computational hosting.

## I. Communications Requirements for Distributed Engine Control Architectures

The goals of the distributed control vision can only be realized if a standard communications infrastructure can be achieved. In this section, the requirements for such an infrastructure are briefly examined. Several papers outlining the requirements of the network for distributed control applications have been published [JJ+03][Rus01][PH08]. The following summarizes and highlights these requirements:

**Real-Time Communication and Determinism.** Timely, deterministic and bounded communications guarantee that data delivery is essentially constant and communication-induced jitter is minimal, which are assumptions made in real-time control systems [WNT95]. Deterministic communication also simplifies the system integration process.

**Support for Architectural Composability**. Architectural composability concerns about the evolution of a system architecture over time without, or with minimal, impact on existing established architecture. For example, adding or removing a node should not result in a complete re-assessment of communication performance and have minimal impact on existing communication patterns. Rushby [Rus01] talks about partitioning properties of networks. Partitioning is basically another word for architecture composability that also explicitly extends the

definition to consider fault situations. In addition to properties mentioned above, Kopetz and Obermaisser [KO02] argue that independent development at node-level should be supported by specification of a precise interface.

**Synchronization** is a key requirement for control systems to achieve not only real-time communication and determinism which supports architectural composibility in the time domain, but also to coordinate access to shared and limited resources hence minimizing resource requirements. If required, synchronization can also be leveraged for time-stamping of events and/or to coordinate output in distributed systems.

**Depedendability** spans a multitude of different aspects.

*Protocol-level fault tolerance and Robustness: F*or any safety relevant system, the core algorithms of the protocol action need to be sufficiently tolerant to faults. The protocol should not be left vulnerable to any justifiable fault scenarios including tolerance of transient faults if required by the environment. For protocols that leverage synchronization, the foundation and assumptions of the synchronization algorithms are paramount. The synchronization protocol should not only include fault tolerant clock synchronization during synchronous operation, but should also consider the less often invoked but equally important startup and integration scenarios The protocol behavior of synchronization should not only include tolerance to all faults within the fault hypothesis, but also include means to address robustness aspects in order to resolve any potential system state after faults outside of the fault hypothesis, within bounded periods in a legal system state. Such robustness is also called self-stabilization of a protocol and in the case of synchronous communication networks referred to as clique resolution or aggregation [PH08,SPK06].

*Message Authentication and timeliness:* For distributed systems, the ability of the communication protocol to authentic a signals source is very important, since voting planes and other fault mitigation strategies built commonly on top of the communication infrastructure may be defeated by masquerade failures. The core protocol itself should also be tolerant to masquerade induced protocol failures if suitable masquerade protection strategies such as guardians are not provided. Similarly, a communication protocol should ensure that a signal sourced by a node, is delivered within a timely manner. For communications systems that incorporate storage during message transport, e.g. Switches, strategies for the erroneous delay and re-cycling of messages need to be provided.

*Independence and coverage of Fault containment capabilities:* For protocols that implement protection and fault containment services e.g. [BKS03][HD+05][HPD07], the fault coverage and strength of the protection is key. If not sufficiently independent, it may lead to dependability vulnerability and escapes. In addition, since "covering functions" are usually transparent to normal mode operation, mechanisms to periodically check (i.e. scrub) the protections are required in order to detect latent faults. Exercising of the protection circuit can be a non-trivial task and should be considered during the design of the communication network.

*Physical layer Robustness and Isolation:* The physical layer and strength of the protocol topology is also an important attribute, especially in harsh environments, where physical damage, e.g. fire, etc., may occur. Such events can lead to "spatial proximity faults" that may cause damage to the communications medium at at single physical location. A simple local short could result in loss of system-wide communication. In addition, the physical layer needs to support special requirements of isolation against environment or system faults, such as ground faults or lightning (e.g. by being able to be transformer coupled).

*Interlocking and Protocol Mode Control:* For protocols that incorporate loading and diagnostic modes of operation, interlocking between the different modes needs to be assured. Usually such diagnostic and loading modes of operation are not designed to be fault-tolerant, and may indeed require nodes to be taken off-line. Therefore, inadvertent entry into such modes may impact system availability.

Protocols should scale with the current and future expected needs of the network deployment.

**Low Complexity**: The design correctness or integrity needs to be assured for safety-critical environments, which is also often referred to as the certification process. The complexity of a communication network can have a significant impact on the guarantees of correctness and/or certification costs. Therefore, it should be the goal to minimize complexity.

**Efficient Redundancy Management:** Independence of layers on top of network; all system-level protocols should consider faults of the software on top of the network. Since software faults can be common mode faults, any vulnerability – often present by simple dependency of initiation of the algorithms on software – should be avoided if possible [PH07]. If not, all software hosted on top of the network needs to be assured to the level required by the system.

**Scalability:** Scaling of the number of nodes in a DCS, within expected ranges, is an important property. This means the adding of nodes should not require redesign of the whole communication architecture (e.g. physical layer drivers in busses only have limited drive capability preventing extension unless properly designed; point-to-point architectures are more favorable) and maintain performance requirements (bandwidth). Non-functional properties

such as reliability and an independent of number of nodes are also favorable.

## II. Discussion of Current and Emerging Communication Technologies

Over the past decade, a number of communication technologies have been considered viable solutions for the distributed engine control problem. This section briefly describes some of the favored technologies in light of the desired attributes discussed above. This section is not an exhaustive examination or survey of all suitable technologies; our purpose is to briefly illustrate the current "state of the art" and to identify the challenges in deploying current communications technologies.

### A Controller Area Network

Controller Area Network (CAN) was originally developed for the automotive industry by Robert Bosh. GmbH. Since its introduction in the early 1990's, CAN has become the defacto networking standard within the automotive industry. Hence, CAN is now commonly integrated into embedded microcontroller platforms and therefore is very attractive from a cost perspective because the costs associated with CAN protocol itself are largely removed. CAN has also been demonstrated within a laboratory environment to supply suitable bandwidth for a distributed engine control architecture [TB+99]. However, although inexpensive and pervasive, CAN has a number of dependability challenges in relation to safety-relevant application deployment.

When implementing a carrier sense, multiple arbitration (CSMA) scheme, CAN nodes monitor and arbitrate access to the communications medium according to the priority of the message that they want to send. Messages with higher priority are non-destructively arbitrated using a bit-wise arbitration of a globally-assigned unique message identifier that is transmitted at the head of each CAN frame. Contending nodes that lose arbitration immediately reconfigure to receive the winning message. It has been argued that by using a suitable allocation of message priorities the worst-case temporal behavior of message propagation can be established using analysis. [TB+99]. However, as the network load increases, the determinism of the CAN network breaks down. Similarly, the predictability and composability also degrades as nodes and messages are added and removed from the network.

With respect to fault-tolerance, the CAN Protocol encompasses a number of node-local, self-monitoring strategies that require erroneous nodes to "remove themselves" from the network in response to detected erroneous conditions and strike counters. However, CAN provides no means of independent fault-containment, thus a faulty node that does not "obey protocol" can disrupt and inhibit communications availability. In addition, since CAN is implemented as a bus topology, it is also vulnerable to spatial proximity faults; physical damage to the communications medium at any single point can lead to loss of communications availability. The data-link layer is also not DC-balanced, hence electrically isolating components is not straightforward.

With respect to communications integrity, CAN offers little protection as it has no mechanisms to implement system partitioning guarantees. On a CAN network, any node is allowed to send any message; therefore, the masquerade vulnerability (the ability to erroneously source a message from an incorrect location) is systemic, and such failures have been observed during software and hardware fault-injection campaigns [SM05]. Studies have also uncovered vulnerabilities with respect to interference between CAN's CRC and bit-stuffing implementation [TB+99] and issues of atomic broadcast consistency [RV+98].

A number of dependability augmentation strategies have been proposed for CAN, including simple guardian schemes [BB03], physical layer isolation and reconfiguration schemes [SO+06], and centralized guardian approaches [BP+06], [HP+07]. Similarly, high-level protocols such as TT-CAN [LH01] and FT-CAN [AFF98][APF02] have also been proposed to increase CAN fault-tolerance and determinism. However, these strategies require augmentation above the core protocol, and as such, the costs, overheads, and complexities of implementing the extensions must be considered in addition to the costs of the base protocol.

### B Time-triggered Protocol: TTP®

TTP [KB03] is a technology that was designed for safety-critical transportation systems (automotive, aerospace, railway) [TTA98] and originally intended to be the low cost communications platform for full-authority hard real-time, x-by-wire control applications [XbW98]. Developed in the mid 1990's, TTP is a fully deterministic protocol implementing a strictly time-triggered communications model. In TTP, each node is allocated access to the network in accordance with a static *a priori* configured TDMA schedule. Unlike CAN, the determinism of a TTP network is not affected by network loading. The strong, deterministic nature of the communications model offers strong system composabilty that allows messages to be added or removed from the network without impact—providing that they are suitably provisioned within the initial TDMA schedule. Therefore, TTP offers significant advantages over CAN in relation to certification and incremental change management.

TTP is a strong candidate for a distributed control communications backbone. It has been successfully applied to representative FADEC data flows within the Honeywell Modular Aerospace Control (MAC) in use across multiple engine platforms. . In addition, following its deployment within commercial aerospace applications such as A380 and the Boeing 787, the TTP communications controller itself has been argued to satisfy the stringent requirements of FAA DO-254 and DO-178; as such, it is very attractive for aerospace focused applications. With the provision for Manchester encoding at the data link layer, TTP also enables easy galvanic transformer isolation between distributed components.

Although strong in many areas, the lessons learned during the first 10 years of working with TTP in MAC indicate that there are areas for possible improvement. The scheduling of TTP networks in heavily loaded systems is non-trivial, complicated by the constraints of the strict TDMA protocol, in the form of same slot order and same slot size per node/per TDMA round. Hence good scheduling capability and associated tooling is essential. Secondly, the fail-silence fault hypothesis, that underpins the protocol, needs to be carefully examined in relation to the needs and implementation of the target application domain. For the MAC platform an independent low-complexity central bus guardian was developed to maximize fault containment. Finally, in common with any protocol with strong autonomous behavior, the system level influence and interaction of core protocol algorithmic behavior needs to be carefully examined with respect to the requirements and implementation constraints of the target application. The continued research [SK06] of time-triggered protocols presents some interesting advances and capabilities with respect to robust time-triggered communication.

## C FlexRay™

FlexRay [Fle05] has been designed as an alternative to TTP for the next generation of automotive x-by-wire applications. Targeting more flexibility, FlexRay offers two types of communication: strict time-triggered messaging, and mini-slot arbitrated dynamic messaging that enables run-time, priority-based message arbitration. In many ways the suitability of FlexRay to distributed control is similar to that of TTP. With respect to scheduling constraints, FlexRay offers different trade-offs, providing the flexibility for nodes to send more than once in a round, with the additional constraint that all TDMA transmissions need to be the same size. Note that, since FlexRay provides no guarantees with respect to the dynamic segment, i.e. the mini-slotting protocol, it is uncertain whether this capability could be used by safety-relevant functions, and it is therefore not considered a viable mitigation to the scheduling issues.

In relation to fault-tolerance of the FlexRay protocol there is no fault-tolerance hypothesis stated or published. The protocol exhibits vulnerabilities during start-up and node integration [HPD07]. Finally, although FlexRay is finding gradual acceptance in the automotive industry and being integrated onto next-generation microcontroller platforms, the lack of D0-254 certification may inhibit broad aerospace acceptance. In addition, the physical encoding scheme of the network is not DC balanced; hence, electrically isolating nodes from the communications medium may be a challenge. Furthermore the protocol and physical layer is optimized for automotive usage and therefore specified to maximum 24 meters length.

## D Intellibus

IntelliBus is a transducer bus developed by Boeing and Aeroflex from the aerospace domain targeted for automotive and similar applications [Ell01][Bau04]. IntelliBus's protocol leverages a master/slave-based bus mechanism derived from MIL-STD-1553B. In Intellibus, a Network Interface Controller (NIC) functions as a master that coordinates sendnig, receiving, and management of multiple network device interfaces (NDIs).

IntelliBus deploys multiple variants of physical layers, which are designed having electromagnetic compatibility (EMC) friendliness in mind. The physical targets up to 30 Mbit/s. Intellibus also contains additional features such as a discovering, recognizing and assigning logical addresses, which is called the membership service.

IntelliBus is not yet widely used and has many advantages for use in distributed control networks, like master/slave-driven variable schedule deployment, which allows bandwidth-efficient deployment in sensor networks, and no requirement for a host processor in NDIs, which allows very simple transducer implementations for NDIs.

A main drawback of IntelliBus for safety-critical deployment is that it does not address dependability support at the network level. Even though multiple redundant interfaces and busses can be deployed, the protocol itself is linked on the logical (protocol) level leaving redundancy useless for device fault that could impact protocol operation. Also

even simple failures like babbling nodes are not tolerated as no guardian support is provided.

## E IEEE-1394/ SAE 5643

Another technology often discussed in the context of distributed control is IEEE-1394b [IEE02] and, more specifically, SAE AS 5643 [SAE06b] protocol extension. Finding acceptance as the backbone network on an advanced jet fighter, IEEE1394 is often favored due to its high bandwidth capabilities. However, being conceived to satisfy the requirements of consumer electronics with provisions for plug-and-play etc., IEEE 1394 is an interesting challenge with respect to fault-tolerance. On the positive side, the point-to-point connectivity of the IEEE-1394 tree topology provides a foundation on which spatial proximity and physical medium fault-tolerance can be built. Indeed, the core link layer protocol of IEEE-1394 can detect and reconfigure physical layer connectivity in response to a detected connectivity failure or requested connectivity reset. On the negative side, the ability of the core protocol to reset and reconfigure presents a significant challenge to availability. Unfortunately, IEEE 1394B presents no mechanisms to guard against erroneous bus reset or connectivity upsets. In addition, since the loss of power to a PHY on one node may result in a system-wide reset, to meet any level of dependability a specialized power architecture that contains such faults is required. Similarly, since IEEE-1394 protocol flow is determined by software, the protocol is also vulnerable to software-induced errors, including inadvertent bus reset faults. IEEE-1394 is also vulnerable to integrity failure since the protocol incorporates no authentication mechanisms for message transport, even though the basic topology is implemented using active repeater stage action. With such an active repetition action it is difficult to argue that a CRC alone will be sufficient to capture all possible node induced errors, since CRC fault coverage assumptions may not hold [PM05].

The SAE 5643 extensions partially mitigate this integrity failure vulnerability by introducing heart-beat and message sequencing fields into higher level protocol action. This protocol also addresses some of the timing indeterminism of 1394 by implementing a master driven TDMA protocol on top of the underlying IEEE-1394 asynchronous data transport mechanisms. Since the protocol depends on the underlying IEEE 1394 foundation, it is difficult to argue any level of protocol fault tolerance even with these extensions. For systems that have leveraged IEEE-1394, extensive architectural mitigation with multiple independent networks are required as suggested by the reference architecture of SAE 5643. Such architectural mitigations may drive overhead and redundancy in cost-constrained environments.

## F Aerospace and Industrial Ethernets

### *F.1.1 ARINC 664*

Another broadband technology that has gained significant traction with aerospace and industrial control is Ethernet. In aerospace applications, ARINC 664 [ARI05] or AFDX™ (a full-duplex, profiled, switched Ethernet) has established itself as the de facto standard on large air transport applications. It is deployed on both the Airbus A380 and Boeing 787.

ARINC-664 augments standard switched Ethernet by incorporating additional enforcement mechanisms with message rate limiting and source authentication in the switch, together with additional message traffic shaping and redundancy management services in end-systems. The data flow of ARINC 664 is asynchronous, and the worst-case system behavior can only be predicted using the aggregation of all network data flows. Similar to CAN, the propagation of a message through the network depends on network loading but with much stronger jitter guarantees. System composibility is therefore compromised and incremental changes and certification need careful analysis and provisioning. Currently used and targeted as an avionics only solution, the initial ARINC 664 implementations are relatively costly when contrasted with COTS Ethernet technology. The lack of synchronization support as available in industrial real-time Ethernet complicates its deployment in high-speed real-time distributed control applications. Therefore the feasibility of ARINC 664 achieving cross-industry acceptance is uncertain at this time.

### *F.1.2 Ethernet Powerlink*

Ethernet Powerlink [Pow06] is a deterministic real-time protocol mostly deployed in the industrial control application. Powerlink deploys a periodic access cycle over standard Ethernet. A cycle is divided in three major

variable phases. In the start phase the current master node (managing node) sends out a synchronization message to all nodes. The start phase is followed by an isochronous phase, where the managing node addresses each node by a special request frame, which is answered individually by the addressed node. As each node is continuously listening, all nodes can listen to responses from other nodes. The third phase is the asynchronous phase and initiated by the managing node. During this phase standard IP-based protocols and addressing can be used.

In general Powerlink is a master-driven protocol with failure checks and an initial master selection and ability to take over masters. Yet, the approach is based on gentleman principles as such dependent on the master implementation in case of failures; therefore to meet high-availability requirements independent network segments may be required. With respect to integrity, the Ethernet Powerlink Safety (EPLsafety) protocol an extension to Powerlink which supports transport of safety-related data and short cycle times (100 microseconds) is recomemend.. It s argued to satisfy the requirements of SIL (safety integrity level) 3 as defined in IEC61508 [IEC05]. EPLsafety is based on checksums and network monitors for failure detection. As such, it achieves error detection coverage that may be limited as argued in [PM05].

### *F.1.3 EtherCAT™*

EtherCAT [IEC04] is a master/slave based protocol on top of a full-duplex Ethernet physical layer. The physical topology can be nearly arbitrary, but the logical architecture forms a ring using both Ethernet directions for its deployment. The novel aspect of EtherCAT is that it uses a message insertion mechanism into Ethernet frame. Each node on the logical ring will insert message data into Ethernet frame data sections on-the-fly according to specified table entries and referenced by header information. The initial frame is sent by the master which signals to slaves via the EtherCAT header embedded in the Ethernet frame data. From this perspective it is similar to MOST [MOS05]. Due to this message insertion mechanism data can be distributed extremely fast (in the order of the Ethernet frame transmission time) and the bandwidth utilization is very efficient. A drawback of EtherCAT is that the insertion behavior requires dedicated non standard Ethernet hardware. However EtherCAT master nodes may be implemented using standard COTS hardware.

Dependability in EtherCAT is based on consistency checks. Due to its master/slave nature it is dependent on the implementation of the master. EtherCAT has a potential vulnerability with respect to data corruption as each forwarding node can access and potentially destroy or alter all data in an Ethernet frame. Integrity of data is determined by the checksum error detection coverage.

### *F.1.4 TTEthernet™*

Time-triggered (TT) Ethernet is a recent addition to the real-time Ethernet protocols. A joint development undertaken by Honeywell and TTTech, TTEthernet is targeted at broad variety of cross industry applications e.g. aerospace, automotive and industrial control applications. TTEthernet provides backward compatibility with existing avionics standards ARINC 664 (at layer 2), while augmenting the services to support true real-time, time-triggered message exchange. It therefore offers similar real-time performance to the industrial Ethernets discussed above. However, targeted with the needs of fault-tolerant Avionics in mind the TTEthernet protocol offers additional fault tolerant synchronization start-up and error recovery algorithms. In addition to address scalability, TTEthernet also provides the ability to support multiple independent synchronization domains and thus enables the hosting of independently synchronized distributed sub-systems. Since TTEthernet uses a standard frame format (which can be compatible with ARINC-664) for all messages, communication between independent synchronous domains and asynchronous network clients is implicitly provided via normal TTEthernet switching action.

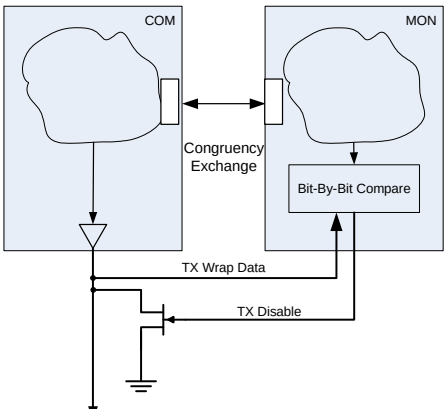


*Figure 2. TTEthernet Command/Monitor Configuration*

High-integrity variants of TTEthernet also mitigate the complex failure modes of the switching action by incorporating self-checking command/monitor (COM-MON) component configurations as indicated on Figure 2. In such configurations one IC monitors the output of another and any disagreement in expected output results in a disabling of the Tx Path, before the message completes. Hence to all clients of the component pair the data is either good or detectably faulty. To prevent erroneous input into the pair from impacting pair agreement the COM/MON

configuration also introduces input congruency exchange where input validity if each input frame is exchanged and agreed between COM and MON components. TTEthernet also facilitates self-checking configurations for end-system components. The near full fault coverage of the self-checking component configurations presents a validated fail-silent fault model. Such a model can be used to simplify application redundancy management schemes, since all data sourced from such modules is either good to detectably faulty.

TTEthernet has also been conceived to function in mesh and line-based topologies, and therefore offers similar wiring optimizations: PowerLink and Ethertcat. However, with a focus on core dependability, TTEthernet offers additional fault containment and synchronization fault-tolerance when contrasted to industrially focused alternatives. TT-GbE, a Gb-only variant of TTEthernet has already been selected by NASA for use on the ORION manned space program. TTEthernet has been selected for an industrial automation application.

### F.2 Conclusions With Respect To Current Communications Technologies

The previous sections have outlined some of the issues with current communications technologies. Although these technologies serve their target markets well, and some offer promise in the context of on engine distributed control, no single technology can meet the needs of a full distributed control vision. The constraints of on-engine operation and the desire to implement a truly integrated architecture preclude the use of architecture mitigation of multiple independent redundant network segments. Similarly, the overhead of centralized guardian schemes and associated constraints of centralized wiring topologies may also detract from efficient on-engine distribution. In addition the performance of on-engine electronics may limit the bandwidth capabilities of engine mounted communications hardware and therefore protocol efficiency and protocol scheduling restrictions may remain a significant concern. Finally as the computational needs of improved control and life optimizing algorithms increase, the scalability of the communications and computational architecture is also an increasingly important consideration.

## G Dependability Augmentation Using a BRAIN

In this section, we introduce some dependability augmentation strategies that can be blended with any of the above technologies to close the identified gaps. The strategies presented are termed BRAIN, for Braided Ring Availability Integrity Network. The BRAIN is not a communications protocol per se; it is better viewed as a guardian and dependability augmentation strategy that can be fused with any protocol. See [HPD07] for an illustration of this strategy that uses FlexRay. Much of the rationale that underpins the BRAIN was adopted from the lessons learned during the application of TTP to the MAC architecture. Hence, the BRAIN strategy was conceived to address the significant challenges of on engine distributed control.

As the name suggests, the BRAIN is built upon a braided-ring topology that augments the standard ring topology with increased connectivity. In addition to neighboring connections, a node is also connected to its neighbor's neighbor via a link called the braid or skip link (see Figure 3).

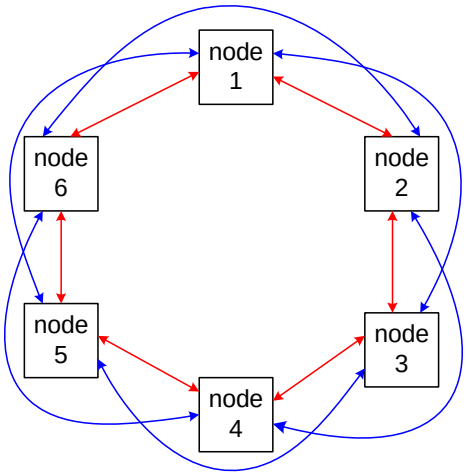


*Figure 1. Channel Availability Mapping*

The BRAIN is a flooding network, as opposed to a store-and-forward network; hence, inter-node propagation delay is minimal, comprising only a few bits delay for each hop. For protocols such as TTP and FlexRay, the BRAIN can be viewed as a logical bus from a protocol perspective. Forming a bi-directional ring, the BRAIN offers two channels, directions of availability, and multiple mechanisms to augment data integrity. For a detailed review of the mechanisms refer to [HD+05][PH08][PH07]. However, a brief summary of these mechanisms is given in the next section.

## A Guardian Action

Guardian capability is incorporated into the BRAIN architecture via a Brother's Keeper Guardian physiology, where nodes guard their geographic neighbors. In synchronous operation, the nodes adjacent to the currently-scheduled transmitter implement guardian enforcement actions, thus the guardian can be pictured as moving around the ring as the TDMA communication sequence progresses. The policies enforced by the guardian circuitry can vary dramatically depending upon protocol requirements and assumptions. Since the BRAIN topology enables the implementation of the guardian on board the same silicon as the communications controller, it is possible for the guardian to leverage the protocol state information maintained by the controller. Therefore, it is feasible for the guardian behavior to include intelligent, complex, fault-containment strategies, for example the enforcement of protocol semantic state correctness. For protocols such as IEE1394, such guardian action could also comprise the enforcement of STOF frame [SAE06b] source or message identification polices. Note that the geographic relation of the guardian ensures that guardian action is fully independent, even if it is embodied into the communications controller hardware.

Note that, with such a guardian strategy, the early bus topology limitation of slot order and slot size for protocols such as TTP and FlexRay may be conceptually removed. Using the Brother's Keeper guardian strategy, the central guardian overheads can be avoided and the cost savings of an integrated controller guardian component can be fully realized, without a loss of guardian integrity.

## B High Integrity Data Propagation

The guardian strategy described above is sufficient to ensure that the nodes scheduled to transmit do not introduce erroneous messages into the system. However, in serialized topologies such as rings and trees, the Brother's Keeper guardian strategy does not protect against faults injected downstream of the guardian nodes. Also given that each node in such topologies comprises active repeater action, the topology is vulnerable to repeat stage induced errors discussed previously in relation to IEEE-1394. To mitigate this issue, the BRAIN incorporates additional high-integrity data propagation mechanisms which are described in detail in [HD+05] In summary, as data propagates around the ring, each node is monitored for correct data propagation by the next node downstream

through bit-for-bit comparison between the data received on the direct and the skip link. Data corruption is signaled to nodes downstream with special integrity fields in the data flow or indicated via truncation, e.g. truncation before the CRC is propagated. The precise action depends on the configuration of the ring (full-duplex or half-duplex links) , the host protocol properties and framing, and fault tolerance level that is to be achieved. Because data flows in two directions, each node receives correct data despite any arbitrary or even malicious single point failure. To tolerate multiple faults, each receiving node compares data received from two directions and accepts data if it is bit-for-bit identical—even if it is not signaled with inline high data propagation integrity (integrity reconstitution). This comparison makes the system tolerant to multiple benign faults with high integrity. Extensions with achieving benign faults relying on detectable faults are possible.

### C Topology Strength

The braided-ring topology of the BRAIN is also arguably optimal with respect to fault-tolerant systems. The reliability of braided-ring topologies has been investigated in literature and a research on this related to BRAIN is described in [HD+05]. Summarizing, the additional links provide significant additional reliability especially for long mission times.

In [PH07], the reliability of braided-ring compared to dual-star topologies is evaluated in the context of extended dispatch scenarios very similar to time-limited dispatch in engine control systems as described in ARP5107 [ARI06]. The results are that braided rings generally outperform dual stars significantly in typical architecture and deployment scenarios with respect to reliability.

With point-to-point links, the BRAIN architecture implicitly addresses the spatial and physical layer damage issues discussed above and can tolerate complete loss of communications at any single geographic location on the ring. Similarly, a node may drop out from the ring and the system will remain operational with integrity guarantees intact. The point-to-point connectivity also mitigates physical layer composability of a shared medium bus topology. In a BRAIN topology, it is also possible to change the physical medium between ring segments. Thus, long segments or segments subject to harsh EMI, HIRF, and cross-talk requirements may be made optical without forcing the costs of the optical links to all systems nodes.

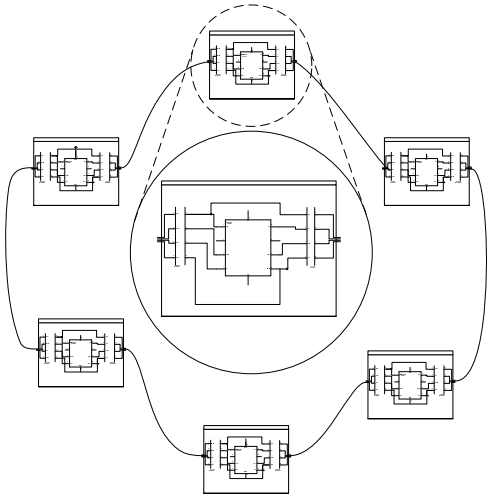

*Figure 4. Physical Topology of BRAIN can be conventional ring (even with skip links routed on board)*

Additionally, the skip links in the BRAIN are mainly for integrity. Hence, the physical routing of skip links could be in the same shield as the direct links and potentially even via the neighboring boards, resulting in simple physical ring-like architectures from a cabling perspective as indicated in Figure 4.

.

## III. An Example Distributed Architecture Using TTEthernet and BRAIN Fusion

The principal mechanisms of the BRAIN are largely protocol-agnostic and equally applicable to protocols such as FlexRay, TTP/C, and IEEE-1394. However, in light of the goals of the MAGIC [BAR07and Universal FADEC [BEH06] technology visions, we believe that an Ethernet-based implementation may be the best instantiation to meet short and long application needs for the following reasons:

1. Openness and pervasiveness of Ethernet technology.
   Ethernet is everywhere, with even low-end processing elements such as PICs incorporating Ethernet hardware. Since industrial control is also migrating from CAN to Ethernet-based infrastructure, the availability of suitable Ethernet hardware is assured.
2. The inherent scalability of Ethernet.
   Ethernet has demonstrated ability to scale and is a strong candidate for large-scale systems deployment, such as ground support and airframe systems. If the on-engine segments are also Ethernet, it is possible to remove the traditional overhead associated with gateway and conversions between sub-systems.
3. The ability to decouple between high-bandwidth and low-bandwidth communication segments via store and forward switching action.
4. This last point is particularly important, as it enables the separation of high-bandwidth computational exchange data-flow from on-engine control data-flow. Ethernet's ability to decouple sender and receiver using switching

technology enables high performance producers to be decoupled from lower performance consumers. When this capability is integrated with a coordinated time-triggered data flow across high and low performance segments, it enables optimization of lower bandwidth on-engine data without limiting the performance of time-triggered high-bandwidth hosted computations. To illustrate the fusion of BRAIN and Ethernet, consider the hypothetical architecture is depicted in Figure 5.

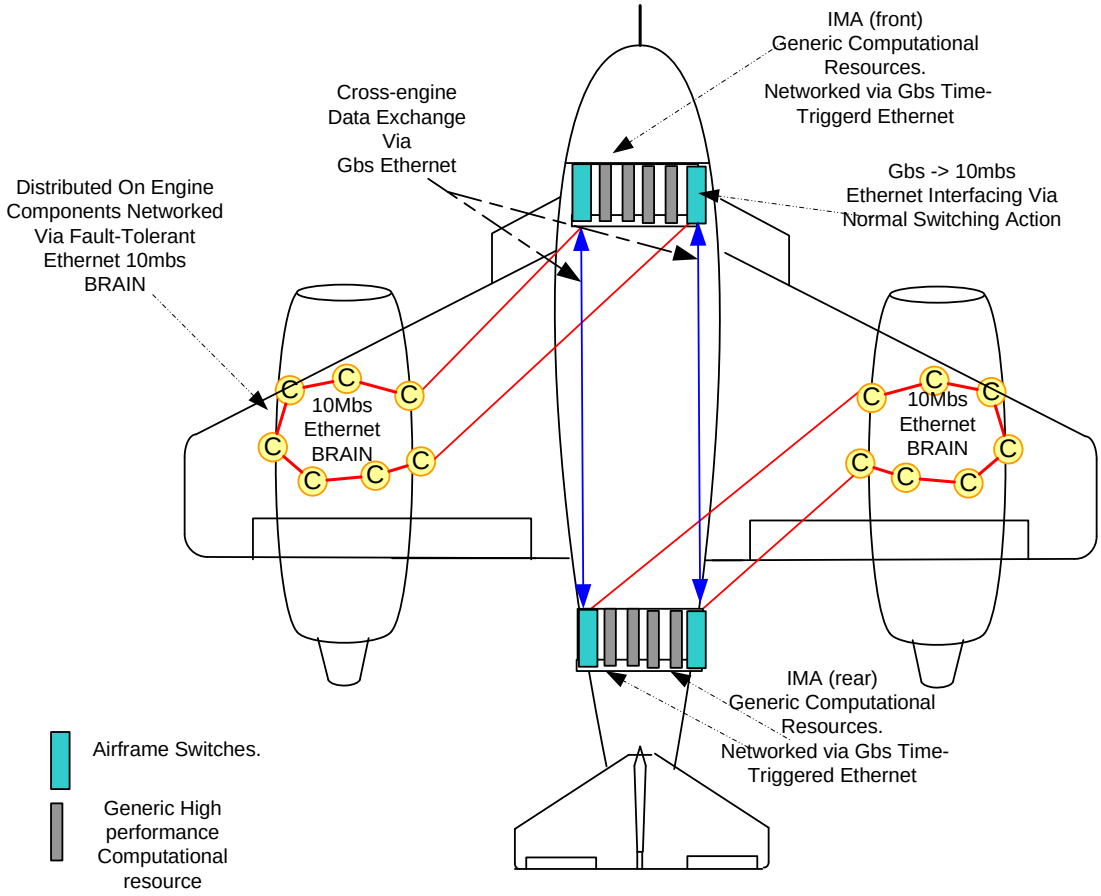


*Figure 4. Hybrid BRAIN TTEthernet Architecture*

In this initial architecture, the high-performance computation of control and prognostics functions have been completely removed from the engine and reside inside a generic integrated modular avionics (IMA) computational resource cabinets. For maximum availability and zonal fault tolerance the IMA is envisioned to be redundant and placed at separate positions in the airframe (i.e. in forward and aft electronics bays). Leveraging this independence in our example each engine is controlled via the resources of separate cabinets, thus any failure of a single cabinet cannot impact both engines. For maximum fault coverage and simpler system level redundancy management, the computational cards in this example architecture are expected to be self-checking and fail-passive in nature. Once again the investment in high-integrity computational platform is not a FADEC only expense, and can be amortized at the airframe level where the associated system returns from simplified redundancy management can present greater system level benefit. To leverage such resources effectively the architectural composability needs to be maintained. To achieve this, the only interface into the dedicated computation cards is Gbs time-triggered Ethernet. In the architecture shown each IMA cabinet is envisioned to be an independent time-triggered Ethernet synchronization domain where computational tasks are aligned with IO resources in accordance with the TTEthernet communications schedule. Each cabinet is therefore conceptually independent and communications between the cabinets is implemented using loosely couple asynchronous (ARINC 664) messaging. This path may also be used for cross–engine data exchange.

The connectivity between the computational cards and the on-engine communications segments is via a high integrity Time-triggered Ethernet switches. To realize real-time high speed control this communication is time-

triggered and therefore high deterministic in nature. The switches are fully schedule aware and enforce, buffer and dispatch messages in accordance with the time-triggered communications schedule. The buffering action of the switches also serves to decouple the high bandwidth computational resources from the lower bandwidth on engine network segments. At the time of writing the feasibility of achieving and/or requiring on-engine Gbs communications for control is currently uncertain; therefore, in this hypothetical architecture the on-engine control segments are implemented using a lower bandwidth for example 10 or 100Mbs full-duplex BRAIN-based Ethernet; which may be more applicable to the constraints and capabilities of high temperature components. Since both the BRAIN and TTEthernet are time-triggered, the timeline of the BRAIN-based communication segments can be driven by the high-integrity computational timeline of the IMA, and thus asynchronous boundaries and oversampling are avoided. Due to the decoupling action of the store and forward switch the high-integrity computational nodes need to "know nothing" of the lower bandwidth BRAIN constraints; everything is abstracted to a time-triggered Ethernet message. In addition since TTEthernet switches also incorporate time-triggered buffering and time-triggered store-and- forward action; the communications schedules of the high performance nodes may be further decoupled from the on-engine communications segments. This is particularly enabling, since it enables changes to the IMA (e.g. processor upgrades, new applications added etc) to be made without impacting the on-engine behavior. Since the entire schedule is time triggered with a common timeline across both high-performance and on-engine segments this flexibility is achieved without any loss in real-time control performance. Simple worst-case deadline analysis can be used to justify cross segment schedule invariants, which can be maintained as system composability temporal anchors. This may greatly ease incremental certification arguments, as the temporal composability and interfaces to the on-engine electronics may be isolated from changes in the shared more volatile IMA schedule.

Additionally, since the high-integrity COM/MON configuration of the TTEthernet switch interfaces naturally as a self-checking interface to the BRAIN segments, with COM and Mon of the switches driving different BRAIN links, the integrity of crossing between the TTEthernet and BRAIN-based segments is assured using the simple mechanisms already present in the BRAIN.

The protocol implemented on the BRAIN-based segment may be a simple Ethernet, i.e. simple message flooding. Alternatively, if bandwidth is very scarce, a variant of the EtherCAT register insertion scheme could be used, i.e. nodes modify a packet in real-time as it is forwarded on the network (see section F.1.3). In either case, the integrity of the data during transport is protected by the BRAIN high-integrity data propagation mechanisms. In this initial archicture BRAIN based on-engine electronics are considered simple slaves to the IMA computation cabinets. Since all data on the network is sourced and relayed in a high-integrity manner, the on-engine control elements can be greatly simplified, adopting a pick-first valid data selection algorithm in place of voting, etc. Thus, they may be realizable with a hardware-only implementation with minimal processing overhead. Such simplicity is attractive given the critical nature and stringent certification design assurance processes, e.g. DO-254, associated with such components

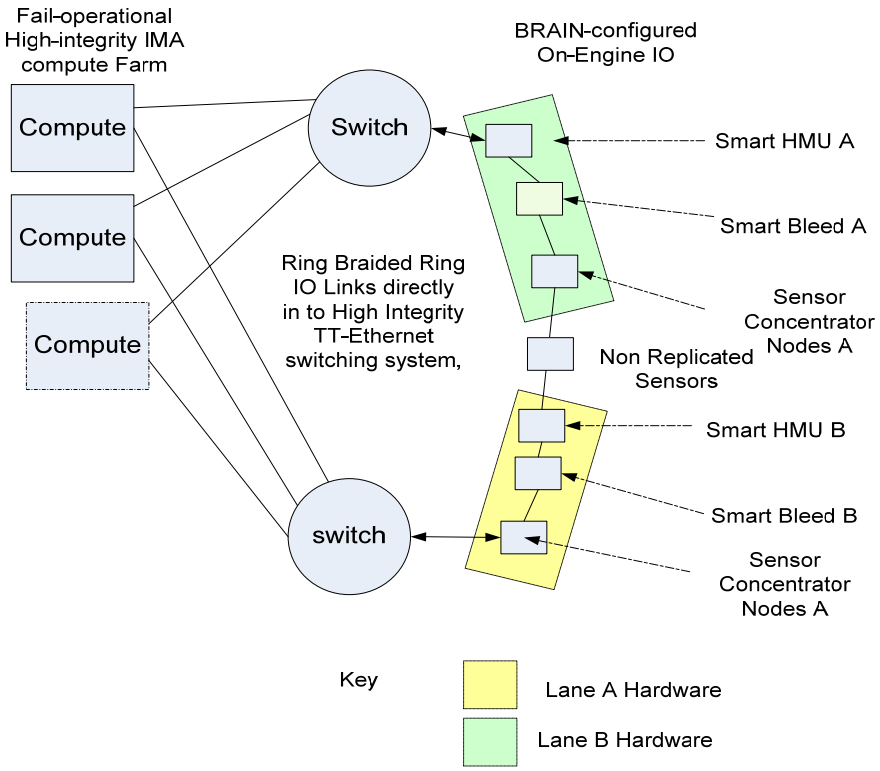


*Figure 6. Architectural HW Lane mapping*

Note, as illustrated in Figure 6, the architectural layout in the example architecture above still maintains the availability of the dual-lane control, since the channel hardware is consolidated with inline elements; hence, a failure in one channel does not impact the failure of the other. However, in accordance with the desires of MACIC and Universal FADEC technology visions, all sense and control data is on a single network.

In relation to scalability and processing performance perspective, this current architecture appears to be very enabling allowing computational power to be added without impacting the on-engine segments. However, such a strategy may present challenges to certification because the engine operation could be impacted by the reliability of the IMA. Even though IMA platforms are deigned to supply ultra-dependability guarantees (less than 10-9 chance of failure per operating hour), certification of a fully integrated architecture may be challenged by the increase in common-mode dependencies.

To offset this fear and to improve system availability, additional computational nodes and switches may be added into the architecture as illustrated in Figure 7. The additional computational resources may be similar or dissimilar, high-performance computational resources dedicated to engine control computations, alternatively they me be lower power nodes supporting only reversionary fall-back control. With the emergence of DO-254 certifiable soft processing cores entering the industry, the obsolescence issues associated with these dedicated, lower power computational resources may be resolvable as life-time IP ownership, and certification arguments are becoming feasible. Similarly with soft-core processing engines the integration of the backup computation resource and network hardware is also becoming feasible. If interfaces become standardized the common use of components may also enable the production of high temperature instantiations of such integrated components economically feasible. Note that the ability of the BRAIN to compose simple COTS cores into high-integrity self-checking pairs (SCP) [HD+05] is very enabling as it allows standard and unmodified COTS cores to realize the high-integrity computational function with nearly zero software and hardware overheads. Such a configuration may be a suitable platform to host reversionary simplified control functions. As indicated in Figure 7, the placement of the additional backup "revisionary control" elements with self-checking pairs (SCP) is important. In the ideal, they would be placed between the two "channels" of BRAIN based on-engine hardware. From such a location they have full access to both channels of the FADEC independently and all sensor data. This has the advantage that the IMA may cease operating in its entirety i.e. "fall of the plane" and the engine still performs without interruption. Such a strategy may also be useful for engine maintenance, allowing engines to power-up independent of the IMA platform. Note that this hybrid architecture is supported by simple priority-based synchronization and clique aggregation protocols of the BRAIN [PH08] where the start-up and recovery of timelines can be biased and prioritized for key specific nodes.

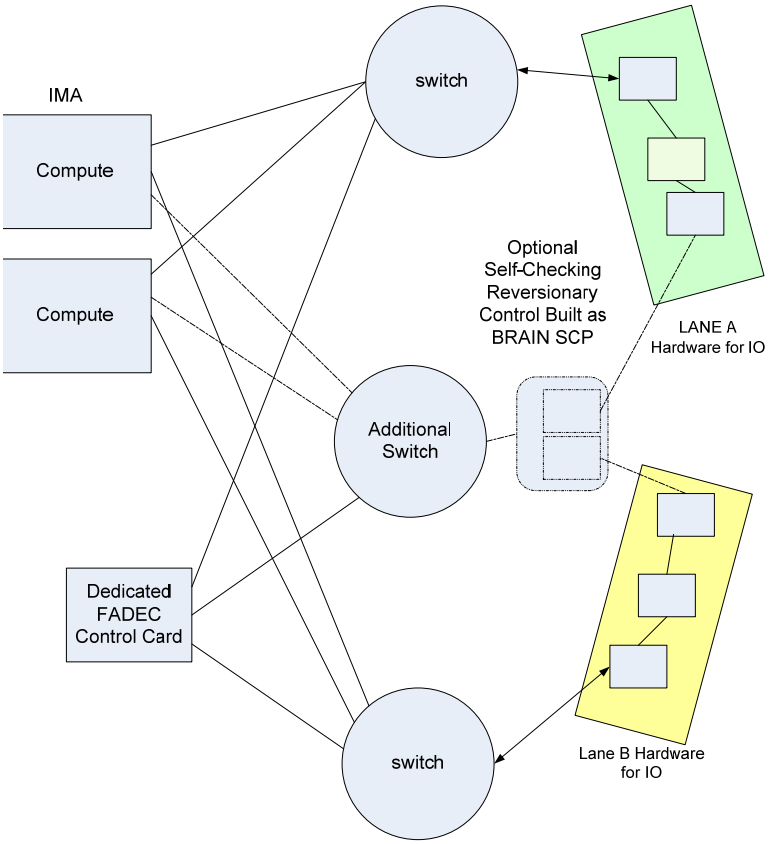


*Figure 7. Architectural Options for Reversionary Backup control*

For larger engines, additional simplex or self-checking computational elements may be added to the network for localized distributed control. It should also be noted that in addition to self-checking configurations the BRAIN also enables three adjacent nodes to be configured into a TMR computational set. Hence, variants on this theme are also possible. Additional ring loop backs and cross channels are also possible.

## IV. Discussion and Conclusion

In this paper we have presented an overview of communications challenges to achieve cost effective reliable on-engine distributed control. We have also reviewed current and emerging communication technologies with respect to their relative strengths and weaknesses in light of these challenges. In the presentation of the BRAIN, we have illustrated how simple communications dependability augmentation strategies can improve availability and integrity, without increasing cost, weight nor wiring complexity.

With the presentation of the hybrid TTEthernet BRAIN architecture we have shown how the selection of a common scalable open technology such as switched Ethernet, that can implicitly support multiple communications bandwidths, it is possible to divide the distributed architecture into high-performance and low-performance segments using a common communications protocol. The flexibility and scalability of this architecture is very enabling towards the distributed FADEC control vision. This scheme enables the high-temperature on-engine control segments to be optimized with respect to the constraints of the extreme on engine environment, however it does not limit the communications bandwidth of the high-performance computational backbone, where experience has shown the requirements of reconfigurable control, and advance health monitoring may continue to increase demand. The integration of the high-performance computational platform within an IMA and the removal of the computation function from the on-engine system electronics are also very attractive from reliability and life-cycle cost optimization. The investment in high integrity compute hardware, and associated obsolescence management can then be amortized and managed at the airframe rather than remain a FADEC only expense. Similarly airframe resources such as cooling etc may leveraged to improve computational hardware reliability. In additional new functionality may be then added to the FADEC system without impacting the design of the on-engine segment.

However to achieve such and architecture we have emphasized that the composability of the architecture is a key element of consideration. In systems that share common computation and communications resources, it is required to be able to change and add additional tasks to the IMA without impacting the on-engine segments. We have shown that by leveraging the time-triggered buffering and store and forward action of the TTEthernet switch such decoupling is indeed possible without impacting real-time control capability. The use of a BRAIN based Ethernet to extend the reach of the high-integrity compute functions to the on-engine electronics is also very attractive, since with guaranteed data integrity redundancy management within the on-engine control electronics can be greatly simplified.

Finally utilizing the BRAIN's unique mechanisms for node pairing to configure high-integrity computational self-checking pairs, together with the BRAIN's advanced synchronization and start-up control primitives, it is possible to supplement the architecture with additional compute capability to implement reversionary back up control. This is enabling as it can mitigate the common mode dependency of the IMA integration and reduce the associated certification risk. In addition it may enable maintenance etc to be performed without the full IMA present. Due to the high integrity guarantees of the self-checking pair configuration the redundancy management and complexity of the on-engine hardware remains and is largely impacted.

There are many other benefits from a common Ethernet based architecture that we have not discussed, such as simplified loading, and test equipment strategies, but it is hoped that the savings of a common communications infrastructure are more obvious in the regard.

The hybrid BRAIN architecture is the critical piece of DCS that interconnects the bus between each smart component. What BRAIN offers is a more effective management of redundancy by reducing complex control logics at the central levels and eliminating the bottlenecks. By standardizing the communication bus with a more fault tolerance, we can encourage COTS, and reuse of hardware and software can improve higher scale ability. This would make DCS more cost-effective and easier air framer integration.

It is hoped the architecture and rational presented in this paper is s considered interesting and relevant to the industry and that it serves to stimulate future architectural evolution, to bring the promised returns of the distributed control vision closer.

## D Acknowledgments

The authors would like to thank Phil Rose, Kelly Morrell, Kevin Driscoll and Chris Wilkinson for the for their encouragement, and feedback during the preparation of this material

## V. References

[AFF98] L. Almeida, J. Fonseca, and P. Fonseca. “Flexible time-triggered communication on a controller area network,” In Proc. of the Work in Progress Session of the 19th IEEE Real-Time Systems Symposium, 1998.

[APF02] L. Almeida, P. Pedreiras, J.A.G. Fonseca. “The FTTCAN protocol: why and how,” IEEE Trans. on Industrial Electronics, pp. 1189-1201. Vol. 49, Issue 6. Dec 2002.

[AIR05] ARINC. Aircraft Data Network. Part 7 Avionics Full Duplex Switched Ethernet (AFDX) Network. ARINC Specification 664P7. Published June 27, 2005.

[BAR07] Bhal Tulpule, Alireza Behbahani, and Richard Millar, AIAA-JPC 2007, Manuscript # 79420,“Vision for Next Generation Modular Adaptive Generic Integrated Controls (MAGIC) For Military/Commercial Turbine Engines”.

[BEH06] Alireza R. Behbahani**,** AIAA-JPC-2006, Manuscript # 58745, **“**Achieving AFRL Universal FADEC Vision with Open Architecture Addressing Capability and Obsolescence for Military and Commercial Applications ”.

[Bau04] R. Bauer, “IntelliBus Protocol: Flexible and Cost-Effective Automotive Bus,” presented at Real-Time Automotive Seminar, 2004. Available at http://techonline.com/electronics_directory/techpaper/193103708

[BB03] I. Broster and A. Burns. “An analysable bus-guardian for event-triggered communication.”. 24th Real-Time Systems Symp. pp. 410- 419. 2003.

[BKS03] G. Bauer, H. Kopetz, and W. Steiner, 2003. The Central Guardian Approach to Enforce Fault Isolation in the Time-Triggered Architecture. Proc. of the 6th Int. Symp. on Autonomous Decentralized Systems. 2003.

[BP+06] M. Barranco, J. Proenza, G. Rodriguez-Navas, and L. Almeida. “An active star topology for improving fault confinement in CAN networks,“ IEEE Transactions on Industrial Informatics, Vol. 2, Issue 2, pp 78-85. May 2006.

[Ell01] P. Ellerbrock, “IntelliBus Network Protocol Specification”, Intellibus Network Systems, 2001.

[FIT00] Project “Fault Injection for TTA (FIT)”. Proj. Ref. IST-1999-10748. May 2000.

[Fle05] FlexRay Consortium. FlexRay Communications System. Protocol Specification. Version 2.1. Dec. 2005.

[Goe98] X. Jay Goetz, High Temperature Electronics for Sensor Interface and Data Acquisition Sensors Expo, Honeywell SSEC, October 7, 1998.

[HD+05] B. Hall, K. Driscoll, M. Paulitsch, S. Dajani-Brown. “Ringing out Fault Tolerance. A New Ring Network for Superior Low-Cost Dependability”. Proc. of Dependable Systems and Networks. Pp. 298-307. 2005.

[HP+07] B. Hall, M. Paulitsch, K. Driscoll, and H. Sivencrona. “ESCAPE CAN”. SAE 2007 Transactions Journal of Passenger Cars: Electronic and Electrical Systems. Doc. No 2007-01-1487. April 2007.

[HPD07] B. Hall, M. Paulitsch, and K. Driscoll. FlexRay BRAIN Fusion – A FlexRay-Based Braided Ring Availability Integrity Network. SAE World Congress. Paper No 2007-01-1492. 2007.

[IEC04] IEC. “Real Time Ethernet Control Automation Technology (EtherCAT)”, Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/355/NP, Date of circulation: 2004-11-19.

[IEC05] IEC. IEC61508 Functional Safety. Parts 0 to 7. 1998, 2000, and 2005.

[IEE02] IEEE Std. 1394b-2002 IEEE Standard for a High-Performance Serial Bus - Amendment 2. pp. 1-369, 2002.

[JJ+03] R. Johansson, P. Johannessen, K. Forsberg, H. Sivencrona, and J. Torin, 2003. “On Communication Requirements for Control-by-Wire Applications.” Proc. of the 21st Int. System Safety Conference, 2003.

[KB03] H. Kopetz and G. Bauer. „The Time-Triggered Architecture,” Proceedings of the IEEE. Vol. 91(1), 2003.

[KO02] H. Kopetz and R. Obermaisser, 2002 “Temporal composability - Real-time embedded systems”.

Computing & Control Engineering Journal , Vol 13(4), pp.156-162, Aug. 2002.
[LH01] G. Leen and D. Heffernan. “Time-triggered controller area network”, Computing & Control Engineering Journal, Vol. 12(6), pp. 245-256. Dec 2001.
[MOS05] MOST Cooperation. Media Oriented Systems Transport (MOST) Specification. Revision 2.4. available at: http://www.mostcooperation.com, 2005.
[PM+05] M. Paulitsch, J. Morris, B. Hall, K. Driscoll, E. Latronico, P. Koopman. “Coverage and the Use of Cyclic Redundancy Codes in Ultra-Dependable Systems”. Proc. of Dependable Systems and Networks, pp. 346-355. 2005.
[PH07] M. Paulitsch and B. Hall. “Insights into the Sensitivity of the BRAIN (Braided Ring Availability Integrity Network)--On Platform Robustness in Extended Operation.“ Proc. of Int. Conf. on Dependable Systems and Networks. pp. 154-163. 25-28 June 2007.
[PH08] M. Paulitsch and B. Hall. “FlexRay in Aerospace and Safety-Sensitive Systems”. IEEE Aerospace & Electronic Systems Magazine. To be published. 2008.
[Pow06] Ethernet Powerlink Standardization Group. “Ethernet Powerlink V2.0 – Communication Profile Specification”. Draft standard, version 1.0.0. Available at www.ethernet-powerlink.org. 2006.
[Rus01] J. Rushby. “A Comparison of Bus Architectures for Safety-Critical Embedded Systems”. NASA Contractor Report CR-2003–212161, Computer Science Laboratory, SRI Int., Menlo Park, CA, USA, Sep. 2001.
[RV+98] J. Rufino, P. Verissimo, G. Arroz, C. Almeida, and L. Rodrigues. “Fault-Tolerant Broadcasts in CAN” Proc. of the 28th Symp. on Fault-Tolerant Computing. 1998.
[SAE06] SAE. “ARP 5107 (Aerospace Recommended Practice). Guidelines for Time-Limited-Dispatch Analysis for Electronic Engine Control Systems.” Rev. B. Society of Automotive Engineers. Nov 2006.
[SAE06b] SAE. As-1a Avionic Networks Subcommittee. “IEEE-1394b Interface Requirements for Military and Aerospace Vehicle Applications”, Doc. No AS5643. Rev. A. Oct. 2006.
[SK06] W. Steiner and H. Kopetz. “The Startup Problem in Fault-Tolerant Time-Triggered Communication” Proc. of Int. Conf. on Dependable Systems and Networks, pp. 35-44, 2006.
[SM05] H. Salmani, S.G. Miremadi. “Contribution of controller area networks controllers to masquerade failures,” in Proc. of the 11th Pacific Rim Int. Symp. on Dependable Computing. 12-14 Dec. 2005.
[SO+06] H. Sivencrona, T. Olsson, R. Johansson, J. Torin. “RedCAN: simulations of two fault recovery algorithms for CAN,“ Proc. of 10th Pacific Rim Int. Symp. on Dependable Computing. pp. 302-311. IEEE. March 2004.
[SPK06] W. Steiner, M. Paulitsch, and H. Kopetz, “The TTA's Approach to Resilience after Transient Upsets” Real-Time Systems Journal. pp. 213-233. Volume 32 (3),March, 2006.
[TTA98] TTA Project. The EU-funded OMI project 23396 TTA (Time-Triggered Architecture) aimed at the implementation of a time-triggered computer architecture (TTA) for fault-tolerant distributed real-time systems. http://www.vmars.tuwien.ac.at/projects/tta/index.html
[TB+99] H.A. Thompson. H. Benitez-Perez, D. Lee, D.N. Ramos-Hernandez, P.J. Fleming, and C.G. Legge. “A CANbus-based safety-critical distributed aeroengine control systems architecture demonstrator” Microprocessors and Microsystems, Volume 23 (6) pp. 345-355(11),Nov. 29, 1999.
[THW94] K.W. Tindell, H. Hansson, and A.J. Wellings, . “Analysing real-time communications: controller area network (CAN)”. Proc. Real-Time Systems Symposium, 1994. Vol. 7(9), pp. 259-263. Dec. 1994.
[Tra99] E. Tran. “Multi-Bit Error Vulnerabilities in the Controller Area Network“, Master's thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1999.
[WNT95] B. Wittenmark, J. Nilsson, M. Törngren, 1995. “Timing problems in real-time control systems.” Proc. of the American Control Conf. Vol. 3. pp. 200-2004. Seattle, WA, USA. 1995.
[XbW98] Project “Safety Related Fault Tolerant Systems in Vehicles - X-by-wire”. Project Ref. BRPR950032, Brite-EuRam III, 1998.