2005

Design Time Reliability Analysis of Distributed Fault Tolerance Algorithms

Elizabeth Latronico
Carnegie Mellon University

Philip Koopman
Carnegie Mellon University, koopman@cmu.edu

Design Time Reliability Analysis of Distributed Fault Tolerance Algorithms

Elizabeth Latronico, Philip Koopman
Carnegie Mellon University
Pittsburgh, PA, USA
beth@cmu.edu, koopman@cmu.edu

Abstract

Designing a distributed fault tolerance algorithm requires careful analysis of both fault models and diagnosis strategies. A system will fail if there are too many active faults, especially active Byzantine faults. But, a system will also fail if overly aggressive convictions leave inadequate redundancy. For high reliability, an algorithm's hybrid fault model and diagnosis strategy must be tuned to the types and rates of faults expected in the real world. We examine this balancing problem for two common types of distributed algorithms: clock synchronization and group membership. We show the importance of choosing a hybrid fault model appropriate for the physical faults expected by considering two clock synchronization algorithms. Three group membership service diagnosis strategies are used to demonstrate the benefit of discriminating between permanent and transient faults. In most cases, the probability of failure is dominated by one fault type. By identifying the dominant cause of failure, one can tailor an algorithm appropriately at design time, yielding significant reliability gain.

1 Introduction

Distributed fault tolerance algorithms are used for many systems that require high levels of reliability, where a centralized component might present a single point of failure. For example, aviation fly-by-wire and automotive drive-by-wire networks need to reliably deliver data despite the presence of faults. These algorithms tolerate faults through a combination of redundancy, diagnosis and fault removal.

An algorithm's maximum fault assumption states the number of active faults that can be tolerated. For a formally proven algorithm, the system may fail if this assumption is violated. Fault diagnosis procedures aim to keep the number of active faults within the bounds of the maximum fault assumption by removing suspected faulty nodes. However, if fault-free nodes are incorrectly diagnosed as faulty and removed, the risk of inadequate redundancy increases.

Given this tension, how will an algorithm perform under a real-world fault profile? We introduce a methodology to measure the reliability of an algorithm's maximum fault assumption, focusing on two types of design decisions. First, it is important to pick a hybrid fault model that corresponds well with the physical fault sources. A 'good' hybrid fault model defines easy-to-handle categories for many of the physical faults, thereby reducing the risk of failure due to an active fault. Second, a fault diagnosis strategy should treat transient and permanent faults differently. Many transient faults expire quickly and are not caused by a node. For example, channel noise might corrupt a frame, but removing the sending node will not prevent future problems.

We apply our methodology to two case studies. The clock synchronization case study reviews two hybrid fault models. The group membership case study investigates three fault diagnosis strategies. To perform reliability analysis, we first define a reusable physical fault model based on real-world fault arrival rates and types. Next, we show how to construct the reliability models. The models can be customized to include other types of faults; we give an extensibility example. Hundreds of configurations are studied with the NASA Langley Semi-markov Unreliability Range Evaluator (SURE) tool set [6], [7]. By examining many configurations, we can make recommendations without needing precise failure rate data that is usually unavailable at design time. We find that the Strictly Omissive hybrid fault model improves reliability, as does a diagnosis strategy that discriminates between permanent and transient faults.

Section 2 reviews protocols and related work. Section 3 discusses the physical fault model, the hybrid fault model, and the mapping between the two. Section 4 presents results, and Section 5 summarizes conclusions.

2 Protocol Overview and Related Work

We study the clock synchronization service of the FlexRay protocol and variants of the group membership strategy of the Time Triggered Protocol, Class C (TTP/C) [13], [37]. FlexRay is intended for safety-critical automotive applications such as brake-by-wire, where electronic connections will replace the mechanical linkages between the brake pedal and the braking actuators [13]. The FlexRay protocol provides distributed clock synchronization among member nodes. TTP/C is a leading multipurpose safety-critical protocol slated for use in avionics applications and

other domains [37]. TTP/C provides a distributed membership service in addition to clock synchronization. Both protocols use a broadcast Time Division Multiple Access (TDMA) sending scheme, where nodes transmit frames in a predetermined static schedule on dual redundant channels.

Related work has noted the need to measure the reliability of specifications. Powell defines 'assumption coverage' as the probability that a failed component's behavior will be covered by one of the assumed failure modes [29]. Powell demonstrates that adding nodes may decrease the reliability, because adding nodes also increases the fault rate [29]. Bauer, Kopetz and Puschner address the assumption coverage of TTP/C, noting that "every fault-tolerant system relies on the existence of a minimum number of correct components [4]." Even an optimal system may fail in the event of too many coincident faults [4]. Per Powell's definition, we assume that all faults are covered (detected through value or timing checks), but coincident faults may exceed the maximum fault assumption. Our previous work examined the assumption reliability of the NASA Scalable Processor Independent Design for Electromagnetic Resilience (SPIDER) protocols in the face of coincident faults [22].

The design time reliability analysis we perform complements existing work in the area of fault injection. Since exhaustive physical testing is infeasible for ultra-reliable systems [8], other validation approaches are needed. One use of fault injection is to verify that the implementation fulfils its requirements (i.e., faults within the maximum fault assumption do not cause unacceptable errors). Ademaj, Sivencrona, Bauer, and Torin investigate propagated faults in the TTP/C-C1 version of the TTP/C communication controller [1]. Through software and heavy-ion fault injection, that work reported the percentages of different types of observed errors (slightly off specification, reintegration, asymmetric, and babbling idiot) [1], [33].

Fault injection has also been used to test dependability under conditions not covered by the maximum fault assumption. Herout, Racek, and Hlavička tested a C-based reference model of the TTP/C protocol coupled with a set of generic and automotive applications [15]. Part of that work investigated robustness to burst faults that did not conform to TTP/C's maximum fault assumption (the single fault hypothesis) [15]. Our work estimates the probability of multiple simultaneous faults exceeding the maximum fault assumption. We base our fault arrival rates on real-world fault occurrence data, instead of random parameters as is typically done in requirements conformance testing.

Additionally, we show that our methodology and reliability models are extensible, by showing how to incorporate one of the fault types from the DBench Dependability Benchmarking project [10]. While our fault model contains a useful set of fault types, certainly not every fault is included. One of the interesting behaviors the DBench project discovered was component failure due to accumulated errors (for example, due to corruption of hidden registers) [10]. We discuss two ways to represent this behavior: direct extension and phased missions.

We study four sources of physical faults: permanent hardware faults, single event effects, bit error rate, and electromagnetic interference. These types and rates, in Table 1, are representative of the aviation domain. For permanent hardware faults, we use a fault rate of 10^{-5}/hr for a node (large fault containment region) and 10^{-6} for a star coupler or bus (small region) [38]. We test a link fault range of 10^{-8}/hr to 10^{-6}/hr, which is slightly conservative compared to [38] but slightly optimistic compared to [16]. The single event effects class includes faults due to particle collisions. Single Event Latchup (SEL) is the dominant permanent effect [32], with observed SEL rates around 10^{-8} to 10^{-6} latchups/device-hr [26]. Single Event Upset (SEU) is the most prevalant transient effect [11], with measured SEU rates from $1*10^{-8}$ to $4*10^{-10}$ upsets/bit-hr [26]. The bit error rate class includes faults from jitter and amplitude disturbances on the network. Three optical standards give worst-case BERs ranging from 10^{-12} to 10^{-10} [9], [23], [35]; we study a less pessimistic range of 10^{-13} to 10^{-11}. The fourth class, electromagnetic interference, includes correlated burst errors [30], [17], [18]. We focus on lightning strikes, estimated at one strike per 2500 flight hours [12].

Other related topics include protocol comparisons and reliability estimation methods. In his comparison of TTP/C, the NASA SPIDER protocols, the Honeywell SAFEbus network, and FlexRay, Rushby argues that "Any fault-tolerant system must be designed and evaluated against a specific *fault hypothesis* that describes the number, type, and arrival rate of the faults it is intended to tolerate [31]." Kopetz discusses the fault tolerance abilities of TTP/C vs. Flexray in [19], and the PALBUS project reviews a number of data buses including an early version of TTP/C [34]. For reliability estimation, the Probabilistic Model Checker supports probabilistic assurance of properties, including properties modeled through continuous time Markov chains [20].

3 Fault Models and Mappings

To evaluate the reliability of a proposed algorithm, we map the physical fault model to the maximum fault assumption hybrid fault model. The maximum fault assumption (MFA) states the maximum number of active faults such that guarantees can be proven to hold. Physical faults map to one or more of the hybrid fault types. We give the hybrid fault models and mappings for the two types of algorithms studied: clock synchronization and group membership.

Table 2 lists system parameters needed for model transition rates. FlexRay and TTP/C both support 1 MBit/sec bandwidth, with plans to support 10 MBit/sec and possibly 25 MBit/sec [13], [37]. The round duration is determined by the shortest message period required by the system, since each node typically sends exactly once per round [37], [27]. A message period of 10 ms is representative of many embedded networks. A frame duration of 0.1 ms would allow 100 frames of 100 bits each to be sent per second. The fault arrival rate due to SEU faults depends on the number of bits that could be affected. We assume 64 kilobytes, or $64*(2^{10})*8$ bits. This is comparable to the size of protocol controllers. For example, the TTP-C2NF revision 1.2 chip

Table 1. Physical Faults and Rates Studied

Physical Fault Type	Rates Studied
Perm. Node [38]	10^{-5}/hr
Perm. Bus/Star [38]	10^{-6}/hr
Perm. Link [38], [16]	10^{-8}, 10^{-7}, 10^{-6} /hr
SEL [26]	10^{-8}, 10^{-7}, 10^{-6} /device-hr
SEU [26], [11]	10^{-10}, 10^{-9}, 10^{-8} /bit-hr
BER [9], [23], [35]	10^{-13}, 10^{-12}, 10^{-11} err/bit
Lightning [12]	4*10^{-4} /hr

Table 2. System Parameters and Values

Parameter	Value
Bandwidth	1*10^6 bits/sec
Round Duration	10 ms
Frame Duration	0.1 ms
Frames/hour	3.6*10^7 (3600000 ms / 0.1ms)
Memory/Node	64 kilobytes
Channels	2
Nodes	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

has 40 kBytes of SRAM and 32 kBytes of ROM [2]. We perform sensitivity analysis for 256 kBytes.

3.1 Hybrid Fault Models

A hybrid fault model classifies faulty nodes according to fault severity with respect to a group of observers. The Byzantine fault model from Lamport, Shostak, and Pease placed no restrictions on the behavior of a faulty node, thereby covering all possible faulty behaviors and requiring $3n$ + 1 nodes to tolerate n faulty nodes [21]. However, many less severe faults are easier to tolerate, as noted by Meyer and Pradhan [24]. Since fault definitions vary, we use the definitions from the NASA Langley Scalable Processor Independent Design for Electromagnetic Resilience (SPIDER) safety-critical protocol suite [25]. The SPIDER definitions are based on the Thambidurai and Park fault model [36]. Also, we include strictly omissive faults, a useful category proposed by Azadmanesh and Kieckhafer [3].
Good (G) [25] Each good node behaves according to specification; that is, it always sends valid messages.
Benign (B) [25] Each benign faulty node either sends detectably incorrect messages to every receiver, or sends valid messages to every receiver.
Symmetric (S) [25] A symmetric faulty node may send arbitrary messages, but each receiver receives the same message.
Asymmetric (A) [25] An asymmetric (Byzantine) faulty node may send arbitrary messages that may differ for the various receivers.
Strictly Omissive Asymmetric (A) [3] "A strictly omissive faulty node can send a single *correct* value to some processes and no value to all other processes." A fault can "garble a message in transit, but not in an undetectable manner."

To measure the reliability of a configuration, a Markov model is created with states given in terms of the hybrid fault model. A fault may be (P) Permanent or (T) Transient. Abbreviations for the state of a single node or channel are (G) Good, (B) Benign, (S) Symmetric, and (A) Asymmetric/Strictly Omissive Asymmetric. The hybrid fault model is applied to components in three ways: a node may become faulty (subscript N), a channel may become faulty (subscript C), or a node may appear faulty if both channels are simultaneously faulty (subscript NC). All perceived node faults due to channel faults are transient (since if both channels are permanently faulty, the system has failed). As two examples, PS_N would be a Permanent Symmetric faulty Node, and TA_{NC} would be a Node affected by Channel faults that appears to be Transient Asymmetric faulty. A node can be convicted (CONV) and permanently removed from the group. While not explicitly represented in the hybrid fault model, convicted nodes are tracked in the reliability models since the total number of nodes is conserved.

Transitions between states are specified with an exponential transition rate (which assumes uncorrelated fault arrivals). An exponential transition rate is specified in the form $e^{-\lambda t}$ where λ is the transition rate per unit time, and t is time (here, in hours). A single transition may change the state of one or more nodes or channels. We represent correlated faults (such as lightning) with transitions that alter the state of multiple nodes or channels.

We use the NASA Langley Semi-markov Unreliability Range Evaluator (SURE) [6] reliability modeling tool set for this analysis. The SURE tool calculates a reliability bound, where the SURE bounding theorems have algebraic solutions. SURE was designed to evaluate fault tolerant systems and handles models with multiple fault arrivals and recoveries well due to the algebraic nature of the solution engine. Iterative solution methods may take a long time to converge, since the probability of violating the maximum fault assumption can be very low (10^{-8} or less is not unusual). While a detailed discussion of modeling tools is outside the scope of this paper, Butler and Johnson describe the mathematics and give numerous modeling examples [6], [7]. Also, the methodology is not limited to this tool suite.

3.2 Clock Synchronization Model and Mapping

The reliability of a protocol depends in part on how the protocol's hybrid fault model classifies faults. We demonstrate this by comparing the Welch and Lynch clock synchronization algorithm to the improved strictly omissive asymmetric algorithm by Azadmanesh and Kieckhafer [3].

The FlexRay clock synchronization algorithm is based on the formally proven algorithm from Welch and Lynch [39], an approximate agreement algorithm using sets of local clock values. The Welch and Lynch algorithm guarantees synchronized clocks as long as $n > 3a + b$ for n total nodes, a asymmetric faults and b benign faults. In the worst case, an asymmetric faulty node could send a too-high value to one node and a too-low value to another. For a benign fault, the frame could arrive too early or too late at all receivers. To separate failure causes, our modeled maximum fault assumption (MFA) in Table 3 checks asymmetric and

Table 3. Clock Synchronization Maximum Fault Assumptions

Clock Sync$_{WelchLynch}$ MFA.1: $n > 3a$ for n nodes and a asymmetric nodes
Clock Sync$_{WelchLynch}$ MFA.2: $n > b$ for n nodes and b benign nodes
Clock Sync$_{WelchLynch}$ MFA.3: $n > 3a + b$ for n nodes, a asymmetric nodes and b benign nodes
Clock Sync$_{Omissive}$ MFA.1: $n > \alpha$ for n nodes and α strictly omissive asymmetric nodes
Clock Sync$_{Omissive}$ MFA.2: $n > b$ for n nodes and b benign nodes
Clock Sync$_{Omissive}$ MFA.3: $n > b + \alpha$ for n nodes, b benign nodes and α strictly omissive asymmetric nodes

Table 4. Clock Synchronization Transitions

Source	Dest.	Items	[Guard], Main Rate Contributor	Rate Range Tested (λ), Per Hour
G_N	PB_N	1	Perm. HW	G_N * 10^{-5}
G_N	PB_N	1	SEL	G_N * (10^{-8}, 10^{-7}, 10^{-6})
G_N	TA_N	1	SEU * Asym. Susceptible Bits	G_N * 10K * 8 * (10^{-10}, 10^{-9}, 10^{-8})
G_N	TB_N	1	SEU * Bits	G_N * 64K * 8 * (10^{-10}, 10^{-9}, 10^{-8})
G_N	TB_N	$\lfloor N/2 \rfloor$	Lightning	4*10^{-4}
G_C	PA_C	1	Perm. Link (one link)	G_C * (10^{-8}, 10^{-7}, 10^{-6})
G_C	PB_C	1	Perm. Link (bus/star)	G_C * 10^{-6}
G_C	TA_C	1	BER * Bandwidth	G_C * 1*10^6 * 3600 * (10^{-13}, 10^{-12}, 10^{-11})
G_C	TB_C	1	BER * Bandwidth	G_C * 1*10^6 * 3600 * (10^{-13}, 10^{-12}, 10^{-11})
G_N	A_{NC}	1	[$\neg(\exists\ G_C) \wedge \exists\ A_C$], 1/Frame Dur.	3.6*10^7
G_N	B_{NC}	1	[$\neg(\exists\ G_C) \wedge \neg(\exists\ A_C)$], 1/Frame Dur.	3.6*10^7
TA_N	G_N	1	1/Round Dur.	TA_N * 3.6*10^5
TB_N	G_N	1	1/Round Dur.	TB_N * 3.6*10^5
TA_C	G_C	1	1/Frame Dur.	TA_C * 3.6*10^7
TB_C	G_C	1	1/Frame Dur.	TB_C * 3.6*10^7
A_{NC}	G_N	1	[$\exists\ G_C$], 1/Round Dur.	3.6*10^5
B_{NC}	G_N	1	[$\exists\ G_C$], 1/Round Dur.	3.6*10^5

benign faults separately. Since the modeling tools check conditions in order, states that fail to satisfy MFA.1 (for example) will not be checked further for MFA.2.

For TDMA clock synchronization, symmetric faults mentioned previously are equivalent to benign faults since there are no undetectably invalid frames. In a TDMA system, a frame's arrival time is calculated with respect to a time slot defined by the receiver's local clock. If the frame is too early or too late, it will be considered invalid. Unlike an explicitly transmitted timestamp, undetected timing faults are not possible since a frame arriving within the slot window is valid by definition, and a frame arriving outside the slot window is detectably invalid by definition. For a different fault model, undetected faults might be possible.

Recently, an improved bound was developed for a family of approximate agreement algorithms. Azadmanesh and Kieckhafer obtained better fault tolerance for strictly omissive asymmetric faults by enabling voting on different sized local sets [3]. The improved bound is $n > 3a + b + \alpha$, for n nodes, a asymmetric faults, b benign faults and α strictly omissive asymmetric faults. For clock synchronization, all asymmetric faults caused by non-malicious physical phenomena will be strictly omissive asymmetric, since a frame is either valid or detectably invalid. Therefore, the maximum fault assumption reduces to $n > b + \alpha$. Three maximum fault conditions were checked in order to determine the dominant cause of failure, listed in Table 3.

For clock synchronization, the system state S is given by the tuple {G_N, PB_N, TA_N, TB_N, A_{NC}, B_{NC}, CONV, G_C, PA_C, PB_C, TA_C, TB_C}, where Σ (G_N, PB_N, TA_N, TB_N, A_{NC}, B_{NC}, CONV) equals the total number of nodes N and Σ (G_C, PA_C, PB_C, TA_C, TB_C) equals the total number of channels C. Unfortunately, a graphical representation would be prohibitive. The smallest clock synchronization models (four nodes) had 333 states and 2750 transitions. The largest clock synchronization models (fourteen nodes) had 24,783 states and 227,560 transitions. Our previous work contains a graphical model, although for a different protocol [22].

Table 4 lists the clock synchronization model transitions. A transition moves one or more nodes or channels from a good state to a faulty state, or vice-versa, as specified in the Source and Dest. columns. The number of nodes or channels involved in the transition is listed in the Items column. The guard and the reason for the transition are given next. A guard is a condition that must be true for the transition to be taken. For example, for a node to transition to a faulty state due to faulty channels, all channels must be faulty at that point in time (otherwise, at least one valid frame would be transmitted). The rate range tested is given in the last column. For most transitions, each component (node or channel) has an equal and independent probability of being affected, so the rate is multiplied by the number of nodes or channels in the source state. The Table 4 transitions were determined as follows (from top to bottom).

Permanent Hardware Faults. A fail-silent node will not send any frames. This behavior is detectable by all receivers due to the TDMA schedule. Therefore, this fault is permanent benign.

Table 5. Membership Maximum Fault Assumption

Membership MFA.1: If $(\exists a)$, then $a + s + b = 1$ for a asymmetric, s symmetric, and b benign nodes
Membership MFA.2: $s \leq g$ for s symmetric and g good nodes
Membership MFA.3: $g \geq 3$ for g good nodes

SEL. Single Event Latchup may cause a node to transmit an improperly formatted frame or transmit a frame at the wrong time. For clock synchronization, a frame must be both on time and correctly formatted, so we model SEL as permanent benign.

SEU. Single Event Upset is modeled as a transient bit upset (either detected by other nodes or local error codes). If this occurs at the sending node, the effect would be benign. If this occurs in the clock synchronization logic of the receiver, this might be asymmetric, since a transient SEU might alter a single frame only. (If all frames were altered, the receiver would be benign faulty since it would not stay synchronized). The SEU rate is multiplied by the number of susceptible bits (here, 64 kilobytes is modeled). An SEU would have to hit a certain portion of the integrated circuit to cause the asymmetric fault described, modeled as 10 kilobytes with sensitivity analysis in Section 4.

Lightning. Lightning is modeled as affecting half of the nodes simultaneously. These nodes are temporarily benign, recovering after the strike. In general, electromagnetic interference could have many other effects.

Permanent Link Faults. Link faults can have two effects. If a single link between a node and the bus/star coupler fails, the channel appears to be asymmetric faulty, since some nodes will receive the frame and others will not. If the entire bus/star coupler fails silent, then the channel delivers no frames and appears to be benign faulty. We studied a range for the first case, and modeled the second case as a permanent hardware failure at a rate of 10^{-6} failures/hour.

BER. Noise on the communication channel can also have two effects when detected. If the noise is localized near a subset of receivers, the channel will appear to be asymmetric faulty, delivering different frames to different receivers. If the noise affects all receivers, the channel will appear to be benign faulty since no receivers get a valid frame. The BER is multiplied by the bandwidth and converted to hours to get the rate per hour.

Perceived Faulty Nodes due to Faulty Channels. If there are no good channels, and at least one asymmetric channel, then the sender will be perceived as asymmetric faulty since some receivers may get a valid frame and others may receive none (for example, if jitter causes the frame to be received too late at a subset of the receivers). If there are no good channels, and no asymmetric faulty channels, no valid frame will be sent to any receiver and the sender will appear benign faulty. Each time a frame is sent, one good node will be affected, for a rate of 1 / Frame Duration. These transitions are not multiplied by the number of nodes in the source state since there is only one sender at a time (this also applies to transient fault expiration).

Transient Fault Expiration. All transient faults in the model eventually expire. For nodes, the effective fault duration is one message round, since a sender transmits once per round. For channels, a channel is considered good if it can send a frame. Transient channel faults (namely, bit errors) are assumed to have a duration of one frame, which is an appropriate model for bit errors. The transient expiration rates are stated as 1 / (duration in hours).

3.3 Group Membership Model and Mapping

The reliability of a group membership service depends on the diagnosis strategy chosen. A group membership service guarantees that all correct nodes in the group reach consensus on the members of the group within a certain period of time after a fault. The diagnosis strategy dictates which nodes to convict and remove from the group (if any).

The maximum fault assumption we use in Table 5 extends the TTP/C single fault hypothesis slightly with respect to benign and symmetric faults when no asymmetric faults are present. The TTP/C group membership maximum fault assumption is that exactly one fault may occur within two rounds, worst-case [27]. Since the modeling techniques we use do not explicitly support a notion of rounds, MFA.1 states there may not be an asymmetric faulty node and another faulty node at the same time. If only symmetric and benign faults are present, the system should operate as long as half of the group members are good nodes (MFA.2), due to the TTP/C Clique Avoidance procedure [37], p. 68. The minimum fault tolerant configuration is four nodes with at least three good nodes (MFA.3) [37], p. 27. These extensions for symmetric and benign faults have not been formally proven; however, we believe that a single MFA condition of no two simultaneous faults would be pessimistic.

We examine three diagnosis strategies. The standard strategy convicts all faulty nodes and removes them from the group (Convict All). The second strategy is the opposite: no faulty nodes are ever convicted (Convict None). The third strategy attempts to convict permanent faulty nodes and to leave transient faulty nodes in the group, with some misclassification (Convict Some). Note that the new strategies are not formally proven – the goal is to investigate robust application level diagnosis. For example, these could be run at the FlexRay application layer.

Alternatively, one could restate the diagnosis strategies as rapid reintegration rules. For the Convict None strategy, nodes could join the group right after two rounds when consensus is reached. For the Convict Some strategy, the group could use a threshold where nodes are allowed into the group immediately after consensus, up until f faults within some time t when the node is permanently removed from the group. Since reintegration is substantially decoupled from membership, this could minimize any proof changes.

Table 6 lists the states and transitions for the hypothesized membership service, and Table 7 lists the conviction probabilities studied. Here we review changes from

Table 6. Membership Transitions

Source	Dest.	Items	[Guard], Main Rate Contributor	Rate Range Tested (λ), Per Hour
G_N	PS_N	1	SEL	G_N*(10^{-8}, 10^{-7}, 10^{-6})
G_N	PB_N	1	Perm. HW	G_N*10^{-5}
G_N	TA_N	1	SEU * Asym. Susceptible Bits	G_N*10K*8*(10^{-10}, 10^{-9}, 10^{-8})
G_N	TS_N	1	SEU * Bits	G_N*64K*8*(10^{-10}, 10^{-9}, 10^{-8})
G_N	TB_N	$\lfloor N/2 \rfloor$	Lightning	4*10^{-4}
G_C	PA_C	1	Perm. Link (one link)	G_C*(10^{-8}, 10^{-7}, 10^{-6})
G_C	PB_C	1	Perm. Link (bus/star)	G_C*10^{-6}
G_C	TA_C	1	BER * Bandwidth	G_C*1*10^6*3600*(10^{-13}, 10^{-12}, 10^{-11})
G_C	TS_C	1	BER * Bandwidth	G_C*1*10^6*3600*(10^{-13}, 10^{-12}, 10^{-11})
G_N	A_{NC}	1	[$\neg(\exists G_C) \wedge \exists A_C$], 1/Frame Dur.	3.6*10^7
G_N	S_{NC}	1	[$\neg(\exists G_C) \wedge \neg(\exists A_C) \wedge \neg(\exists B_C) \wedge \exists S_C$], 1/Frame Dur.	3.6*10^7
G_N	B_{NC}	1	[$\neg(\exists G_C) \wedge \neg(\exists A_C) \wedge \exists B_C$], 1/Frame Dur.	3.6*10^7
G_N	CONV	1	[$\exists A_N \vee \exists A_{NC}$],(1/(2*Round Dur.))*Pr.Conv.Good	1.8*10^5*(1/G_N)
PS_N	CONV	1	(1/(2*Round Dur.)) * Prob. Conv. Perm.	1.8*10^5*(1.0, 0.99, 0.95, 0.90)
PB_N	CONV	1	(1/(2*Round Dur.)) * Prob. Conv. Perm.	1.8*10^5*(1.0, 0.99, 0.95, 0.90)
TA_N	CONV	1	(1/(2*Round Dur.))*Pr.Conv.Trans.*Pr.Conv.Asym.	1.8*10^5*(0.0, 0.01, 0.05, 0.10)*0.95
TS_N	CONV	1	(1/(2*Round Dur.)) * Prob. Conv. Trans.	1.8*10^5*(0.0, 0.01, 0.05, 0.10)
TB_N	CONV	1	(1/(2*Round Dur.)) * Prob. Conv. Trans.	1.8*10^5*(0.0, 0.01, 0.05, 0.10)
TA_N	G_N	1	(1/(2*R. Dur.))*(1-Pr.Conv.Trans.*Pr.Conv.Asym.)	TA_N*1.8*10^5*(1-((0.0, 0.01, 0.05, 0.10)*0.95))
TS_N	G_N	1	(1/(2*Round Dur.)) * (1-Prob. Conv. Trans.)	TS_N*1.8*10^5*(1-(0.0, 0.01, 0.05, 0.10))
TB_N	G_N	1	(1/(2*Round Dur.)) * (1-Prob. Conv. Trans.)	TB_N*1.8*10^5*(1-(0.0, 0.01, 0.05, 0.10))
TA_C	G_C	1	1/Frame Dur.	TA_C*3.6*10^7
TS_C	G_C	1	1/Frame Dur.	TS_C*3.6*10^7
A_{NC}	CONV	1	(1/(2*Round Dur.))*Pr.Conv.Trans.*Pr.Conv.Asym	1.8*10^5*(0.0, 0.01, 0.05, 0.10)*0.95
S_{NC}	CONV	1	(1/(2*Round Dur.)) * Prob. Conv. Trans.	1.8*10^5*(0.0, 0.01, 0.05, 0.10)
B_{NC}	CONV	1	(1/(2*Round Dur.)) * Prob. Conv. Trans.	1.8*10^5*(0.0, 0.01, 0.05, 0.10)
A_{NC}	G_N	1	[$\exists$ G_C],(1/(2*Round Dur.))* (1-Prob. Conv. Trans. * Prob. Conv. Asym)	1.8*10^5*(1-((0.0, 0.01, 0.05, 0.10)*0.95))
S_{NC}	G_N	1	[$\exists$ G_C],(1/(2*Round Dur.))*(1-Prob. Conv. Trans.)	1.8*10^5*(1-(0.0, 0.01, 0.05, 0.10))
B_{NC}	G_N	1	[$\exists$ G_C],(1/(2*Round Dur.))*(1-Prob. Conv. Trans.)	1.8*10^5*(1-(0.0, 0.01, 0.05, 0.10))

Table 7. Membership Conviction Probabilities

Prob. of Convicting Permanent	1.0, 0.99, 0.95, 0.90
Prob. of Convicting Transient	0, 0.01, 0.05, 0.10
Prob. of Convicting Asymmetric	0.95
Prob. of Convicting Good [if $\exists$ ($A_N \vee A_{NC}$)]	1/G_N

the clock synchronization model. We assume an asymmetric identification probability of 0.95, and assume that the probability of a good node being convicted if an asymmetric fault occurs is 1/G_N. 1/G_N is the minimum for good node conviction in TTP/C, since the symmetric category covers cases where all receivers correctly identify the fault source. More probabilities are investigated in the sensitivity analysis, but the expected probability of good node conviction is uncertain. Some protocols forbid good node conviction [25]. Our hypothesized membership service (Convict Some) may misclassify faults. Not all permanent faulty nodes will be convicted, and some transient faulty nodes will be mistakenly convicted, as shown in Table 7.

For membership, the system state S is given by the tuple {G_N, PS_N, PB_N, TA_N, TS_N, TB_N, A_{NC}, S_{NC}, B_{NC}, CONV, G_C, PA_C, PB_C, TA_C, TS_C}, where Σ (G_N, PS_N, PB_N, TA_N, TS_N, TB_N, A_{NC}, S_{NC}, B_{NC}, CONV) equals the total number of nodes N and Σ (G_C, PA_C, PB_C, TA_C, TS_C) equals the total number of channels C. For the Convict Some strategy, the smallest models (four nodes) had 128 states and 1121 transitions. The largest models (fourteen nodes) had 91,866 states and 1,104,902 transitions. Model size can vary by strategy (for example, in the Convict None strategy, Prob. Conv. Trans., Prob. Conv. Perm., and Prob. Conv. Good are zero so the related transitions are removed).

SEL, SEU, BER. For group membership, benign faults are now symmetric since they may cause data value errors. Asymmetric faults from these sources remain the same.

Permanent Fault Conviction. Permanent faulty nodes, if detected, can be convicted and removed from the group. The rate is multiplied by the probability of convicting a permanent faulty node. For asymmetric faults, the conviction rate is also multiplied by the probability of correctly identifying an asymmetric faulty node. The diagnosis algorithm takes two rounds (worst-case) to execute, so the rate per hour is (1 / (2*Round Duration)).

Transient Expiration; Transient Conviction. Transient faulty nodes may be misdiagnosed as permanent faulty and convicted, at some probability. The rate for this set of

transitions is (1 / (2*Round Duration)).

Good Node Conviction. A new transition is introduced from G_N to CONV, at a rate of (1 / (2*Round Duration)) with a probability of $1/G_N$. Good node conviction only occurs if an asymmetric fault is present, so the guard is [$\exists$ A_N $\vee$ $\exists$ A_{NC}]. This rate is not multiplied by the number of good nodes since an asymmetric fault does not necessarily affect more nodes if the group is larger.

Channels. In group membership, a channel may be symmetric faulty. In TTP/C, if one channel is noisy and the other silent, the receiver counts this as a null frame (not an invalid frame) [37]. Thus a symmetric faulty frame will be received only if both channels are symmetric faulty (since an asymmetric channel dominates all faulty channels).

3.4 Extensibility

Although we focus on a particular set of physical faults, the reliability models and modeling techniques are not limited to this set. We demonstrate extensibility by representing a latent fault, as described in the DBench Dependability Benchmarking project [10]. Latent faults are characterized by a potentially long delay between the fault arrival and the observed component failure. For example, latent faults can occur due to accumulated errors in registers that are not directly observable by the user [10].

One way to represent this type of fault would be to explicitly model accumulated errors. Imagine two types of error counter states, a good register state G_R and a faulty register state F_R. At initialization, there is a specified maximum number RMAX of G_R registers and there are zero F_R registers. A latent fault is represented by a transition from G_R to F_R at some rate λ_{latent}. If latent faults have occurred, these faults may cause good nodes to become faulty. Assume that after some minimum number of latent faults MINFAULT that a good node may become permanently benign faulty. This can be modeled as a transition from G_N to PB_N at some rate $\lambda_{activate}$ (or, at a rate that is a function of the number of latent errors c*F_R*$\lambda_{activate}$ for some constant c). The transition will be guarded with [$F_R >=$ MINFAULT] for some value of 0 <= MINFAULT <= RMAX. If the failed component is modeled as containing the latent faults, this transition will also reset the registers, setting G_R to RMAX and F_R to zero. Other variations on this scheme could be modeled; for example, perhaps the good node will experience a different type of fault, or errors will continue to accumulate until the end of the mission.

A second way to represent this type of fault would be to use a phased mission, as described by Butler and Johnson [7]. This technique applies to missions with non-constant rates, and to missions where failures have different consequences during different operating stages (for example, during aircraft take-off vs. in flight) [7]. A model is created for each phase, where models may have different transitions, transition rates and mission times, but the state space must be the same. At the end of a phase, the probabilities of being in each state are output and used to initialize the next phase model [7]. This sequence is repeated until the last phase.

4 Results

For each parameter combination, a Markov model was created and solved using the NASA Langley ASSIST and SURE tools [6]. This allowed a large number of configurations to be investigated. For clock synchronization, there were (4 through 14 nodes) * (3 SELs) * (3 SEUs) * (3 Perm. Link fault rates) * (3 BERs) = 891 models for each study. For group membership, there were also 891 models for the Convict All and Convict None strategies, and there were 891 * (3 Prob. of Perm. Conviction) * (3 Prob. of Trans. Conviction) = 8019 models for the Convict Some strategy.

The models were more sensitive to the transient faults (SEU and BER) than the permanent faults (SEL and Permanent Link faults). The results suggest that despite their short duration, transient faults can still significantly impact reliability because of their high arrival rate. The models were insensitive to changes in the Single Event Latchup rate. SEL occurrence was modeled as a permanent benign fault for clock synchronization, and a permanent symmetric fault for membership. This indicates that other physical faults are the dominant cause of assumption failures.

4.1 Clock Synchronization Results

Table 8 summarizes the assumption failure rate of the two clock synchronization hybrid fault models studied. Figure 1 shows a histogram of the percent of the 891 configurations falling within each failure rate bin. To examine the cause of assumption failure, Table 9 summarizes the dominant assumption that is violated.

Figure 1 shows that the Strictly Omissive Asymmetric hybrid fault model outperforms the Welch and Lynch hybrid fault model, overall. Over three hundred of the Strictly Omissive Asymmetric configurations achieve an assumption failure rate of 10^{-11} or better, as listed in Table 8. In contrast, none of the Welch and Lynch hybrid fault model configurations achieve a failure rate equal to or lower than 10^{-11}. The lowest assumption failure rates are $1.1*10^{-12}$ for the Strictly Omissive Asymmetric model and $1.3*10^{-11}$ for the Welch and Lynch model, both for the 13 node configurations with the lowest physical fault rates. The highest assumption failure rates are $9.8*10^{-9}$ for the Strictly Omis-

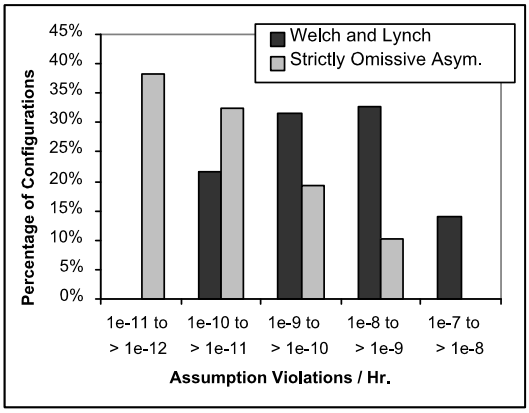

Figure 1. Clock Synchronization Comparison

Table 8. Clock Sync Assumption Violations/Hr.

	Welch and Lynch	Strictly Omissive Asym.
More than 10^{-7}	0	0
10^{-7} to $> 10^{-8}$	126	0
10^{-8} to $> 10^{-9}$	291	90
10^{-9} to $> 10^{-10}$	282	171
10^{-10} to $> 10^{-11}$	192	288
10^{-11} to $> 10^{-12}$	0	342
10^{-12} or fewer	0	0

Table 9. Clock Sync Dominant Failure

	Welch & Lynch	Strictly Omis. Asym.
Active Faults (MFA.1)	837	540
Too Few Nodes (MFA.2, MFA.3)	54	351

sive Asymmetric model and $8.6*10^{-8}$ for the Welch and Lynch model. Moreover, the Strictly Omissive Asymmetric model has the lower failure rate for all configurations when configurations with the same fault rates are compared.

In both hybrid fault models, the assumption failure rate was most sensitive to the Bit Error Rate. The Welch and Lynch hybrid fault model was about equally sensitive to SEU and Permanent Link faults. The Strictly Omissive Asymmetric model was more sensitive to Permanent Link faults than SEU faults. As Table 9 shows, the Welch and Lynch model is likely to fail from active faults. In all configurations except 4 node configurations, MFA.3 (see Table 3) was more likely to be violated. The Strictly Omissive Asymmetric model balances the risk of active faults vs. the risk of inadequate redundancy. MFA.1 was more likely to be violated in some configurations, MFA.2 in others.

Adopting a Strictly Omissive Asymmetric model should be feasible. The clock synchronization algorithm needs to exclude null or detectably invalid values from the voting process. Since correction values are typically stored in a table [13], just valid values in the table could be voted.

4.2 Membership Results

Table 10 summarizes the assumption failure rate for all of the configurations for the three diagnosis strategies studied. Figure 2 plots the percentage of configurations that fall into each assumption failure rate bin. Table 11 lists the number of configurations for each strategy according to the dominant cause of failure.

Overall, the Convict Some strategy had the lowest assumption failure rate, and the standard Convict All strategy had the highest assumption failure rate, as shown in Figure 2 and Table 10. There were Convict Some configurations that achieved a three orders of magnitude decrease in assumption failure rate compared to the other two strategies (10^{-10} to $> 10^{-11}$ in Table 10). The 4, 5, and 6 node configurations all had high failure rates (10^{-5} and up) as compared to the 7 node and above configurations. The assumption failure rates for the Convict All and Convict Some strategies show more of a spread than the Convict None strategy, in Figure 2. This could be due to the conviction of good nodes (no

Table 10. Membership Assumption Viol./Hr.

	Conv. All	Conv. None	Conv. Some
More than 10^{-3}	27	81	243
10^{-3} to $> 10^{-4}$	108	0	729
10^{-4} to $> 10^{-5}$	63	0	729
10^{-5} to $> 10^{-6}$	126	261	972
10^{-6} to $> 10^{-7}$	312	408	972
10^{-7} to $> 10^{-8}$	255	141	2328
10^{-8} to $> 10^{-9}$	0	0	1236
10^{-9} to $> 10^{-10}$	0	0	756
10^{-10} to $> 10^{-11}$	0	0	54
Fewer than 10^{-11}	0	0	0

Table 11. Membership Dominant Failure

	Conv. All	Conv. None	Conv. Some
Active Faults (MFA.1)	0	744	3159
Too Few Nodes (MFA.2, MFA.3)	891	144	4860

nodes are convicted in the Convict None strategy), or the loss of redundancy due to convicted transient faulty nodes.

In all cases studied, the Convict All strategy failed by running out of redundancy, as shown in Table 11. This may be due to lightning strikes, since as modeled two lightning strikes will cause the entire group to become convicted in the Convict All strategy. The probability would be $(4*10^{-4})^2$, or $1.6*10^{-7}$. Other burst effects for lightning could be investigated, but this shows that the Convict All strategy might be a poor performer for burst faults in general. In contrast, the Convict None strategy failed primarily due to too many active faults. The Convict Some strategy balanced the two risks best. The models were most sensitive to the transient faults (BER and SEU), with varying sensitivity to permanent link faults. All three strategies were insensitive to the SEL rate.

Further investigation shows that adding nodes might not improve reliability if the dominant cause of failure is too many active faults. In the Convict None strategy, configurations with 9 or more nodes failed due to too many active faults. The configurations with the lowest failure rates had 9 or 10 nodes – configurations with more nodes had higher failure rates. In the Convict Some strategy, configurations with 13 or more nodes failed due to too many active faults. The configurations with the lowest failure rates had 13 or 14 nodes (the greatest number tested).

We hypothesize that adding nodes will eventually decrease reliability for algorithms whose maximum fault assumption includes a fixed term. Adding nodes increases the chance of a pair of faults, so for membership if MFA.1 is the dominant assumption violated, adding nodes is expected to decrease reliability. Many Byzantine fault tolerant algorithms are expected to include a fixed term in their MFAs, because for a round-based algorithm there must be at least $f + 1$ rounds to tolerate f Byzantine faults [14], and the total number of rounds is usually fixed.

For the Convict Some strategy, we studied various probabilities for permanent faulty node conviction (0.99, 0.95,

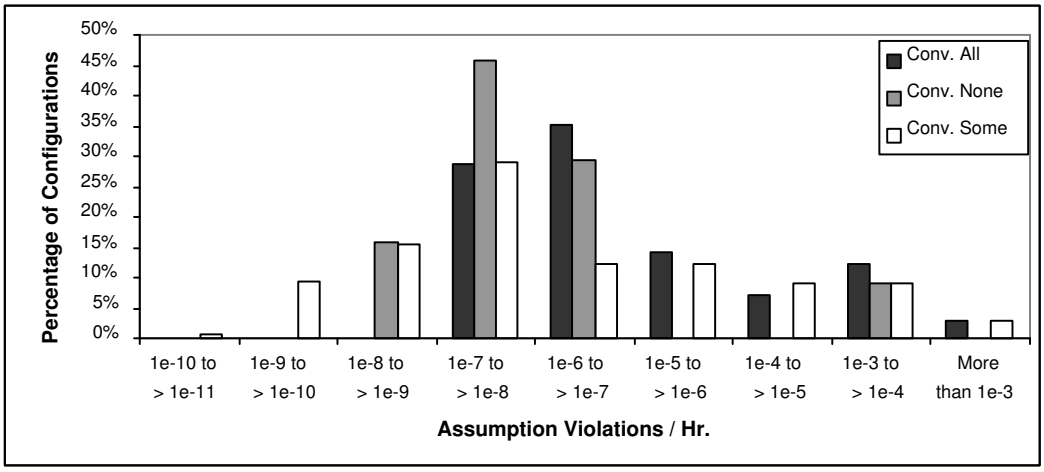

Figure 2. Membership: Assumption Failure Rate Comparison

and 0.90) and for incorrect transient fault conviction (0.01, 0.05, and 0.10). There was some sensitivity to the probability of convicting transient faulty nodes when one of the transient fault rates (Single Event Upset) was high. There was little sensitivity to the probability of permanent fault misclassification, even when permanent fault rates were high. Since the transient fault rates were higher than the permanent fault rates, it makes sense that the models would be most sensitive the type of fault that occurs most often.

4.3 Sensitivity Analysis

This section explores sensitivity to some of the system parameters assumed in Table 2 and Table 7. We selected two studies, the Welch and Lynch clock synchronization and the Convict All membership diagnosis strategy. We fixed the number of nodes at 8. For both models, we studied two different chip sizes (64K and 256K) and four different percentages of bits affected by asymmetric SEUs (0, 15%, 50%, and 100%). There were (3 SELs * 3 SEUs * 3 Perm. Link fault rates * 3 BERs * 2 chip sizes * 4 asym. bits) = 648 configurations for clock synchronization.

For the membership model, we additionally investigated three probabilities of convicting asymmetric faulty nodes (1.0, 0.95, and 0.90) and three probabilities of good node conviction in the event of an asymmetric fault ($1/G_N$, 0.25, and 0.50). At maximum, half the good nodes in the group could be convicted. The SEL rate was kept constant at 10^{-6}, the highest rate studied to see if sensitivity to this parameter increased (it does not). There were (3 SEUs * 3 Perm. Link fault rates * 3 BERs * 2 chip sizes * 4 asym. bits * 3 Prob. Conv. Asym * 3 Prob. Conv. Good) = 1944 configurations for membership.

For the 8 node configurations, the Welch and Lynch clock synchronization model was insensitive to changes in the total amount of memory and to changes in the amount of memory affected by asymmetric SEU faults. However, this model was sensitive overall to the SEU rate as noted in Section 4.1. Upon further inspection, the 7 node or fewer configurations show sensitivity to the SEU rate while the 8 node or more configurations do not. This indicates that other fault types (BER in particular) dominate for the 8 or more node configurations.

The Convict All diagnosis strategy was sensitive to all of the system parameters studied. The model was more sensitive to the total amount of memory than to the amount of memory susceptible to asymmetric SEU. Increasing the amount of memory increases the rates of all SEU faults (benign and asymmetric), so since the Convict All model also convicts benign transient faulty nodes, this may lead to inadequate redundancy. Even for only 90 percent asymmetric conviction, some configurations achieved a failure rate between 10^{-6} and 10^{-5}, which was the lowest failure rate achieved by perfect conviction. This indicates that a practical fault diagnosis algorithm performs fairly well with respect to an ideal algorithm for the Convict All strategy.

5 Conclusions

Distributed fault tolerance algorithms must balance the risk of failure due to too many active faults versus the risk of failure due to inadequate redundancy caused by improper fault diagnosis. An algorithm's maximum fault assumption states the maximum number and type of faults that can be tolerated without possible system failure. The designer's choice of a hybrid fault model and diagnosis strategy affects the probability of violating this maximum fault assumption.

We present a methodology to assess the reliability of the maximum fault assumption at design time, and to determine the dominant cause of failure with respect to this assumption. We illustrate our methodology through two case studies, clock synchronization and group membership. We base our physical fault model on real-world fault types and arrival rates, providing a reusable summary of physical fault rate data, and we give an example of how to extend the models to incorporate a new fault type.

For clock synchronization, a Strictly Omissive Asymmetric hybrid fault model has a significantly lower assumption failure rate than the Welch and Lynch hybrid fault model. A Strictly Omissive Asymmetric hybrid fault model

should be fairly easy to adopt. For membership, a diagnosis strategy that discriminates between permanent and transient faults has a much lower assumption failure rate overall. Also, for a maximum fault assumption including a constant term, adding nodes actually decreases reliability when the dominant cause of failure is too many active faulty nodes. This information could be used to design a rapid reintegration strategy, without changing the underlying proofs.

Acknowledgments

This work is supported in part by the National Aeronautics and Space Administration, Langley Research Center, under agreement NCC-1-02043 awarded to the National Institute of Aerospace, the General Motors Collaborative Research Laboratory and Carnegie Mellon University, the United States Department of Defense (NDSEG/ONR), and the American Association for University Women and the Zonta International fellowship programs.

References

[1] A. Ademaj, H. Sivencrona, G. Bauer, and J. Torin. Evaluation of Fault-Handling of the Time-Triggered Architecture with Bus and Star Topology. *Proc. of the 2003 Intl. Conf. on Dependable Systems and Networks (DSN '03)*, June 2003.

[2] austriamicrosystems AG. AS8202NF TTP-C2NF Communication Controller Data Sheet Rev.1.2, Nov. 2003.

[3] M. Azadmanesh and R. Kieckhafer. Exploiting Omissive Faults in Synchronous Approximate Agreement. *IEEE Trans. on Computers*, Vol. 49, No. 10, Oct. 2000.

[4] G. Bauer, H. Kopetz, and P. Puschner. Assumption Coverage under Different Failure Modes in the Time-Triggered Architecture. *8th IEEE Intl. Conf. on Emerging Technologies and Factory Automation*, Oct. 2001.

[5] G. Bauer and M. Paulitsch. An Investigation of Membership and Clique Avoidance in TTP/C. *19th IEEE Symposium on Reliable Distributed Systems*, October 2000.

[6] R. Butler. The SURE Approach to Reliability Analysis. *IEEE Trans. on Reliability*, Vol. 41, No. 2, June 1992.

[7] R. Butler and S. Johnson. Techniques for Modeling the Reliability of Fault-Tolerant Systems With the Markov State-Space Approach. NASA RP-1348, Sept. 1995.

[8] R. Butler and G. Finelli. The Infeasibility of Experimental Quantification of Life-Critical Software Reliability. *Proc. of the ACM SIGSOFT '91 Conf. on Software for Critical Systems*, Dec. 1991.

[9] E. Chan, Q. Le, and M. Beranek. High Performance, Low-Cost Chip-on-Board (COB) FDDI Transmitter and Receiver for Avionics Applications. *Proc. of the 1998 Electronic Components and Tech. Conf.*, 1998.

[10] DBench Project. Fault Representativeness. Deliverable ETIE2. IST 2000-25425. June 2000.

[11] P. Dodd and L. Massengill. Basic Mechanisms and Modeling of Single-Event Upset in Digital Microelectronics. *IEEE Trans. on Nuclear Science*, Vol. 50, No. 3, June 2003.

[12] Federal Aviation Administration. Instructions for Continued Airworthiness: Advisory Circular 33.4.3 [Draft].

[13] FlexRay Consortium. FlexRay Communications System Protocol Specification, Version 2.0. June 2004.

[14] M. Fischer and N. Lynch. A Lower Bound for the Time to Assure Interactive Consistency. *Information Processing Letters*, 14(4):183-86, June 1982.

[15] P. Herout, S. Racek, and J. Hlavička. Model-Based Dependability Evaluation Method for TTP/C Based Systems. *4th European Dependable Computing Conf. (EDCC 2002)*, LNCS 2485, 2002.

[16] R. Hyle, Jr. Fiber Optics - Failure Modes and Mechanisms. *Proc. of the Reliability and Maintainability Symp.*, 1992.

[17] International Electrotechnical Commission. IEC 61000-4-4. Electrical Fast Transient/Burst Immunity Test. July 2004.

[18] International Organization for Standardization. ISO 7637. Road Vehicles – Electrical Disturbances from Conduction and Coupling. March 2002, June 2004, Nov. 1995.

[19] H. Kopetz. Fault Containment and Error Detection in the Time-Triggered Architecture. *Proc. of the 6th Intl. Symp. on Autonomous Decentralized Systems*, Apr. 2003.

[20] M. Kwiatkowska, G. Norman and D. Parker. Controller Dependability Analysis By Probabilistic Model Checking. *Proc. of the 11th IFAC Symp. on Information Control Problems in Manufacturing (INCOM '04)*, Apr. 2004.

[21] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Trans. on Programming Language Systems*, Vol. 4, No. 3, July 1982.

[22] E. Latronico, P. Miner, and P. Koopman. Quantifying the Reliability of Proven SPIDER Group Membership Service Guarantees. *Proc. of the 2004 Intl. Conf. on Dependable Systems and Networks (DSN '04)*, June 2004.

[23] MAXIM Integrated Products. Accurately Estimating Optical Receiver Sensitivity. App. Note HFAN-3.0.0. Oct. 2001.

[24] F. Meyer and D. Pradhan. Consensus with Dual Failure Modes. *Proc. of the 17th Fault-Tolerant Computing Symp.*, July 1987.

[25] P. Miner, A. Geser, L. Pike, and J. Maddalon. A Unified Fault-Tolerance Protocol. *Formal Techniques in Fault-Tolerance and Real-Time Systems*, 2004.

[26] E. Normand. Single-Event Effects in Avionics. *IEEE Trans. on Nuclear Science*, Vol. 43, No. 2, April 1996.

[27] H. Pfeifer. Formal Verification of the TTP Group Membership Algorithm. *Proc. of FORTE XIII / PSTV XX*, Oct. 2000.

[28] H. Pfeifer, D. Schwier, and F. W. von Henke. Formal Verification for Time-Triggered Clock Synchronization. *7th IFIP Intl. Working Conf. on Dependable Computing for Critical Applications (DCCA-7)*, Jan. 1999.

[29] D. Powell. Failure Mode Assumptions and Assumption Coverage. *Proc. of the 22nd Intl. Symp. on Fault-Tolerant Computing (FTCS '92)*, 1992.

[30] RTCA, Inc. Environmental Conditions and Test Procedures for Airborne Equipment. RTCA/DO-160D, July 29, 1997.

[31] John Rushby. A Comparison of Bus Architectures for Safety-Critical Embedded Systems. NASA CR-2003-212161. March 2003.

[32] F. Sexton. Destructive Single-Event Effects in Semiconductor Devices and ICs. *IEEE Trans. on Nuclear Science*, Vol. 50, No. 3, June 2003, p. 603-21.

[33] H. Sivencrona. On the Design and Validation of Fault Containment Regions in Distributed Communication Systems. Dissertation. Chalmers University of Technology, 2004.

[34] H. Sivencrona, J. Hedberg, and H. Röcklinger. Comparative Analysis of Dependability Properties of Communication Protocols in Distributed Control Systems. PALBUS Task 10.2. Apr. 2001.

[35] R. Stephens. Analyzing Jitter at High Data Rates. *IEEE Optical Communications*, Feb. 2004.

[36] P. Thambidurai and Y.-K. Park. Interactive Consistency With Multiple Failure Modes. *Proc. of the Seventh Reliable Distributed Systems Symp.*, Oct. 1988.

[37] TTTech Computertechnik AG. Time Triggered Protocol TTP/C High-Level Specification Document, Protocol Version 1.1. Specification Edition 1.4.3. Nov. 2003.

[38] U.S. Department of Defense. MIL-HDBK-217F. Reliability Prediction of Electronic Equipment. Dec. 2, 1991.

[39] J. Lundelius Welch and N. Lynch. A New Fault-Tolerant Algorithm for Clock Synchronization. *Information and Computation*, Vol. 77, No. 1, Apr. 1988.