Preprint: The International Conference on Dependable Systems and Networks, DSN-2004.

Quantifying the Reliability of Proven SPIDER Group Membership Service Guarantees

Elizabeth Latronico
ECE Department
Carnegie Mellon University
Pittsburgh, PA, USA
beth@cmu.edu

Paul Miner
NASA Langley Research Center
Hampton, VA, USA
paul.s.miner@nasa.gov

Philip Koopman
ECE Department
Carnegie Mellon University
Pittsburgh, PA, USA
koopman@cmu.edu

Abstract

For safety-critical systems, it is essential to quantify the reliability of the assumptions that underlie proven guarantees. We investigate the reliability of the assumptions of the SPIDER group membership service with respect to transient and permanent faults. Modeling 12,600 possible system configurations, the probability that SPIDER's Maximum Fault Assumption will not hold for an hour mission varies from less likely than 10^{-11} to more likely than 10^{-3}. In most cases examined, a transient fault tolerance strategy was superior to the permanent fault tolerance strategy previously in use for the range of transient fault arrival rates expected in aerospace systems. Reliability of the Maximum Fault Assumption (upon which the proofs are based) differs greatly when subjected to asymmetric, symmetric, and benign faults. This case study demonstrates the benefits of quantifying the reliability of assumptions for proven properties.

1. Introduction

Formal proofs provide an attractive means of developing safety-critical network protocols. Protocols destined for aerospace or automotive use may need to exhibit fewer than 10^{-9} failures per hour [12]. Exhaustive testing of these protocols is a daunting challenge, potentially requiring on the order of 10^{9} hours of testing or more. This precludes exhaustive testing as a sole means of verification. Also, exhaustive testing of the implementation provides no feedback at the design stage, when changes are easiest to make. Formally proven protocols are guaranteed to provide their services at all times - if all of the assumptions hold. Unfortunately, the assumptions will not hold for all possible fault cases. Eventually, weakening the assumptions makes the proof untenable. Arguing that the assumptions are 'reasonable' is inadequate for safety-critical systems, due to stringent reliability requirements. The assumptions must be shown to be reliable to conclude that the formally proven service is reliable.

It is important to investigate assumption reliability over a range of design space, and explore design-stage policy trade-offs. An instantiated system will occupy one point in a large space of possible systems. Current techniques, such as Failure Mode Effects Analysis and Fault Tree Analysis, predict the reliability of a particular instantiation, and may rely on high-precision failure rates to achieve high-precision reliability estimates. However, high-precision failure rate data may not be available, especially for novel systems. We present a methodology based on Markov modeling techniques that evaluates assumption reliability for ranges of parameters that can differ by an order of magnitude or more. Multiple policies can be compared to determine which is more reliable in a given range of the design space.

Our case study reveals that systems with an identical formal proof basis can have vastly different assumption reliability depending on the parameters of the instantiated system. We analyze the reliability of the assumptions of the group membership service for the SPIDER protocols developed by NASA Langley Research Center. First, we study three alternative policies for removing faulty nodes from membership. We focus on assumption reliability in the presence of transient faults, in addition to a permanent fault model. The formal proof does not make a distinction between transient and permanent faults, as it does not need to - both types of faults can be proveably handled with the same mechanisms. However, we show that the presence of transient faults significantly affects the probability that the assumptions will hold for the duration of the mission. We also investigate assumption reliability with respect to different types of faults (asymmetric, symmetric, and benign). The assumptions are less reliable overall for asymmetric faults versus symmetric or benign faults, as asymmetric faults require additional redundancy to handle.

The results clearly show the value of testing the assumptions with expected fault conditions. The model of the service includes static parameters, plus six parameters that vary over bounded ranges. Our experiments cover 12,600 combinations of parameters, showing it is feasible to cover a wide range of design space. This paper complements work on assumption coverage illustrating that 'reasonable' assumptions are not always 'reliable' assumptions, which we discuss following our results. Section 2 reviews the SPIDER protocols and guarantees, and Section 3 describes studied policies. Section 4 explains the modeling process, with an example. Section 5 covers our fault model. Section 6 presents results, and Section 7 discusses related work.

2. SPIDER

As described by Geser and Miner in [10], "The Scalable Processor-Independent Design for Electromagnetic Resilience (SPIDER) is a family of general-purpose fault-tolerant architectures being designed at NASA Langley Research Center to support laboratory investigations into various recovery strategies from transient failures caused by electromagnetic effects." At the heart of SPIDER is the Reliable Optical Bus (ROBUS), which processing elements use to reliably transmit data in a fully-connected, broadcast manner. Formal proofs define the fault tolerance abilities of the ROBUS. The proofs are valid for all transmission media. The ROBUS has two types of components: Bus Interface Units (BIUs) and Redundancy Management Units (RMUs). The BIUs are fully connected to all RMUs, and vice-versa. Each BIU has a one-to-one connection to a corresponding Processing Element (PE). A Processing Element cannot exhibit asymmetric ('Byzantine') faulty behavior, since there is only one direct consumer of its data. Asymmetric BIU or RMU faults are handled internally by the ROBUS, freeing the application designer from this concern, as long as the proof assumptions hold. Figure 1 from Geser and Miner [9] illustrates the SPIDER architecture.

2.1. SPIDER Maximum Fault Assumption

SPIDER is designed to tolerate multiple faulty nodes, yet still provide firm guarantees. SPIDER achieves this multiple fault tolerance in part through its Diagnosis protocol that detects and classifies faulty nodes. The Diagnosis protocol classifies nodes as one of the following [10]:

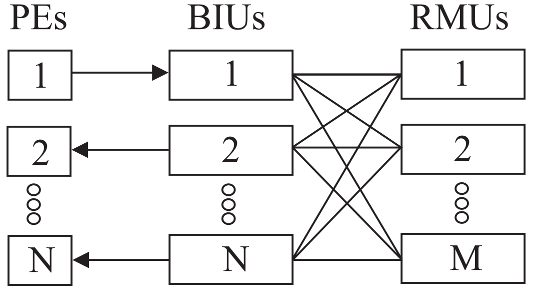

Figure 1. ROBUS Topology, Geser and Miner [9]

- **Good**: A good node behaves according to specification.
- **Benign faulty**: A benign faulty node only sends messages that are detectably faulty (for example, a message with a bad Cyclic Redundancy Code value). This includes nodes that have failed silent.
- **Symmetric faulty**: A symmetric faulty node may send arbitrary messages, but each receiver receives the same message.
- **Asymmetric faulty**: An asymmetric ('Byzantine') faulty node may send arbitrary messages, including different well-formed messages to different receivers.

The SPIDER protocols make a number of guarantees, contingent upon a Maximum Fault Assumption (MFA). The Maximum Fault Assumption specifies the maximum number and type of faults that SPIDER can tolerate. As long as the system satisfies the Maximum Fault Assumption, the SPIDER guarantees are proven to hold. If additional faults are present in the system, the guarantees may not hold. If nodes are not permitted to reintegrate, the Maximum Fault Assumption is as stated in the three parts below, MFA.1, MFA.2, and MFA.3 [10]. (The Maximum Fault Assumption changes slightly if reintegration is permitted. Geser and Miner give further details in [10]). Node amounts are positive [10]:

- MFA.1. Number of Good BIUs > (Number of Symmetric BIUs + Number of Asymmetric BIUs)
- MFA.2. Number of Good RMUs > (Number of Symmetric RMUs + Number of Asymmetric RMUs)
- MFA.3. (Number of Asymmetric BIUs = 0) or (Number of Asymmetric RMUs = 0)

Reliability analysis determines the probability that these conditions will not be true, subject to a given fault model. SPIDER's Maximum Fault Assumption is tight, in the sense that if any of the three parts are violated, there exists an allowed behavior of the faulty node(s) that will violate the SPIDER guarantees. Within the constraints of our fault model, the MFA conditions are necessary. However, the allowed faulty behavior can be quite broad. It is intractable to measure error manifestations for all possible fault sources, in part because it is not possible to determine the set of all possible faults.

2.2. SPIDER Guarantees and Policy choices

First, we examine the guarantees that SPIDER provides, and look at the group membership service for fault tolerance. We focus on the Interactive Consistency (IC) protocol and Diagnosis protocol that the ROBUS implements. The Interactive Consistency protocol provides two guarantees, validity and agreement [10].

- **Validity**: Every good node receives the value sent by a good node.
- **Agreement**: All good nodes agree in the value sent.

SPIDER uses group membership to enable good nodes to ignore some faulty nodes. Chockler, Keidar, and Vitenberg state that "The task of a membership service is to maintain a list of currently active and connected processes in a group." [5]. Active nodes form a set of eligible voters. A group membership service identifies faulty nodes and removes them from this set. Removed, or **convicted**, nodes become 'benign' faulty nodes, as eligible voters will ignore removed nodes. For SPIDER, the goal is to enhance the fault tolerance of the system by turning asymmetric and symmetric faulty nodes into benign faulty nodes. SPIDER's Diagnosis protocol provides two guarantees [10]:

- **Conviction Agreement**: All good nodes agree on convictions.
- **Correctness**: No good node is ever convicted.

Conviction agreement is common to many protocols; however, protocol designers can choose to guarantee correctness or completeness (or neither). Geser and Miner state, "In the presence of arbitrary asymmetric failures, it is impossible to guarantee both correctness and completeness" [10], where completeness means that "all faulty nodes are eventually convicted." By preserving correctness instead of completeness, SPIDER can accumulate evidence against a node as an alternative to immediate conviction. In comparison, guaranteeing completeness would require conviction of transiently faulty nodes, which can significantly decrease assumption reliability (as we will show). A disadvantage to correctness is that it is not possible to convict faulty nodes in some cases.

3. Conviction Policies

The SPIDER Diagnosis protocol may remove suspected faulty nodes from membership. We define the **conviction policy** as the method for determining which nodes to remove from the set of eligible voters. The conviction policy plays an important role in the reliability of the system, because it affects the likelihood that the Maximum Fault Assumption will be violated. The conviction policy must balance the risk of inadequate redundancy versus the risk of too many faulty nodes in the set of eligible voters.

The existence of transient faults poses an interesting dilemma. As used here, the term **transient fault** refers to a fault which persists for a finite duration, ceases to exist after that duration has expired and does not alter the state of the affected component beyond that duration. A **permanent fault** is a fault with infinite duration or lasting effects on state. In our fault model, faults can occur at a node or on the broadcast network. Convicting transiently faulty nodes may decrease the reliability of the system, as the number of available redundant components will decrease. If the transient fault duration is short, then it is probably better to do nothing, letting the transient fault expire.

SPIDER's Maximum Fault Assumption is stated in terms of nodes, so frames that are corrupted due to faults on the network will be perceived as node faults. In the Fault-Error-Failure classification scheme proposed by Deswarte, Kanoun, and Laprie [7], this can be thought of as the 'error' stage. We assume a one-to-one relationship between the incident faults and the errors. Other relationships are possible, depending on the transient fault duration and the evidence required for conviction. Section 5.2 discusses the relationship between transient fault duration and the execution period of the Diagnosis protocol. We do not model faults outside of the SPIDER MFA. Assumption coverage, as defined by Powell [13], addresses these types of faults; please see Section 7 for more details.

We examine three conviction policies: All Permanent, All Transient, and Perfect. These policies represent extreme points in the space of possible conviction policies, and require only a single error for conviction. It may not be possible to perfectly implement the 'All Permanent' and 'Perfect' strategies for two reasons. First, it may not be possible to diagnose the source of a fault, in which case the correctness property prohibits removing the faulty node from membership. Second, it is impossible to distinguish transiently faulty nodes from permanently faulty nodes in some cases. A system can specify a duration ΔT, where faults that persist longer than ΔT are considered permanent. However, one could define a transient fault that persists longer than ΔT for any ΔT chosen, except for a ΔT of infinity, which would be a permanent fault. The 'All Transient' strategy takes no action, posing no obstacles for implementation.

- **All Permanent** (Treat All Faults as Permanent)

In this strategy, all faulty nodes are convicted, regardless of whether the fault is permanent or transient. This strategy equates to a 'treat everything as permanently faulty' strategy. It represents the outcome if permanent fault tolerance only is applied to a system with transient and permanent faults. Note that SPIDER is not able to convict all faulty nodes if correctness is guaranteed, since sometimes the fault source cannot be determined. If completeness is preserved instead of correctness, this strategy is possible, but good nodes may be convicted. In this study, we do not examine strategies where good nodes may be convicted.

- **All Transient** (No Action)

In this strategy, faulty nodes are never convicted. This strategy is an inaction strategy. Faulty nodes are not removed, as transient faults (per our definition) will disappear after a finite duration with no lasting consequences. It represents the outcome if transient fault tolerance only is applied to a system with transient and permanent faults.

- **Perfect**

The third strategy is to convict all permanently faulty nodes, and leave transiently faulty nodes in the set of eligible voters to let the transient faults expire. In this experiment, we consider only transient faults with duration shorter than or equal to the diagnosis period. This is true for populations of transient faults we are considering, such as noise on the communication network which gets mapped to nodes by the group membership service.

4. Reliability Modeling

We present how to construct Markov models to measure the probability that the Maximum Fault Assumption fails to hold (the relevant 'failure' according to the Fault-Error-Failure terminology [7]). If the Maximum Fault Assumption is violated, the guarantees may not hold and the system may fail. Figure 2 illustrates the Markov model for one possible system configuration. This particular model is quite compact – most of the models have hundreds of states and thousands of transitions.

4.1. Software Tools: ASSIST, SURE, STEM

Three Markov analysis tools developed at NASA Langley Research Center were used to model SPIDER group membership service fault tolerance and to estimate the probability that each part of the Maximum Fault Assumption would not hold. The ASSIST program translates parameterized text specifications (in the ASSIST language) into Markov models. Then, either STEM (Scaled Taylor Exponential Matrix) or SURE (Semi-markov Unreliability Range Evaluator) solves the Markov model. STEM provides an exact solution and is limited to pure Markov models. SURE provides upper and lower bounds on reliability, usually within five percent of each other [3], and can handle other classes of transitions besides exponential transitions. Butler and Johnson explain the underlying mathematics of these tools and give numerous fault-tolerance examples in [4]. Our models involved only exponential transitions, so either tool could be used. All of the measurements presented here were done with STEM. Note that another Markov solver could potentially be used - the approach is not limited to these three software tools. ASSIST, STEM, and SURE can be obtained from NASA Langley at: http://shemesh.larc.nasa.gov/fm/ftp/sure/sure.html.

4.2. State Space

A variety of configurations can be modeled with reasonable effort. The designer first specifies the possible component state space items. Link failures get mapped to nodes, so the state space need only cover nodes. The two types of component nodes, Bus Interface Units (BIUs) and Redundancy Management Units (RMUs), are not interchangeable. A node may be good, permanently faulty, or transiently faulty. The SPIDER Maximum Fault Assumption lists three types of faults: benign, symmetric and asymmetric. We modeled these types separately to keep the state space manageable. Convicting a faulty node causes that node to become benign permanently faulty, so all models included benign permanently faulty BIU and RMU state space items. The maximum number of state space items in our models is eight: (2 component types) * (3 fault manifestations) + (2 benign faulty state space items).

The example in Figure 2 shows transient fault arrivals and recovery only, no permanent faults, and no convictions, so there are (2 component types) * (2 fault manifestations, good/faulty) = four items in the state space that the designer must specify. The ovals represent possible states. Since redundant components are present, the system can tolerate some combinations of faulty components. The numbers of working and faulty nodes are listed inside each oval, as the state space, given as (Good RMUs, Faulty RMUs, Good BIUs, Faulty BIUs). The example system in Figure 2 has three RMUs and four BIUs. In the start state at the top, all nodes are working, so that state space is (3, 0, 4, 0). Combinations of faulty nodes are listed in other ovals, for example, the state space (3, 0, 3, 1) represents one faulty BIU, and the state space (2, 1, 4, 0) represents one faulty RMU.

4.3. Transitions

Next, the designer must specify transitions. In this model, nodes are conserved, so a transition will take a node

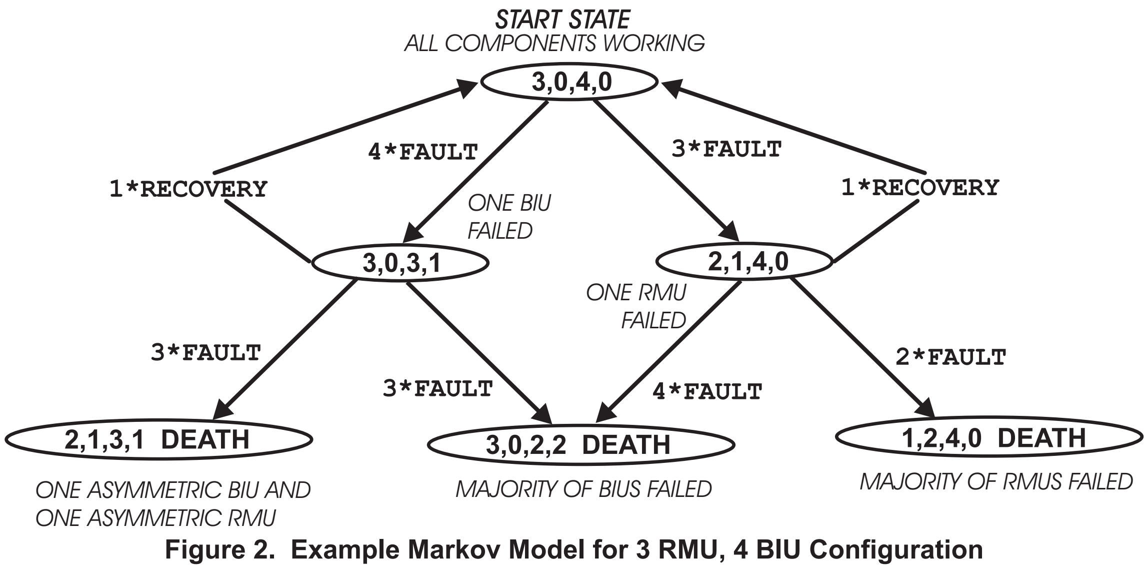

Figure 2. Example Markov Model for 3 RMU, 4 BIU Configuration

out of one state and put it into another. There are three types of transitions. Fault arrival transitions transform a good component into a faulty component. Errors caused by transient faults have a finite duration, so when this duration expires a transiently faulty node reverts to a good node. Finally, an asymmetric or symmetric node can be convicted, becoming a benign permanently faulty node. In this model, there is no reintegration of permanently faulty nodes, so these cannot be transformed into good nodes.

The transitions occur at a rate defined by the fault arrival rate and by the number of components currently occupying a state. For example, if a fault occurs at rate F that transforms a good BIU into a faulty BIU, and there are currently 10 good BIUs, the Markov model transition rate will be 10*F. Our models use exponential rates only, because we are modeling uncorrelated, independent faults. In Figure 2, transition rates are listed on the arrows between states. FAULT is the fault arrival rate and RECOVERY is the transient fault expiration rate (derived from the duration). The example uses constant rates; however, most of our models have multiple fault arrival rates according to fault type. The FAULT or RECOVERY rate is multiplied by the number of eligible components in the source state, for example, the 3*FAULT transition from the start state for the chance of an RMU becoming faulty and the 4*FAULT transition from the start state for the chance of a BIU becoming faulty.

4.4. Death States

Finally, the designer must specify the conditions for death states, where the guarantees may not hold. Our three death state conditions map to the three parts of the Maximum Fault Assumption. The bottom of Figure 2 shows the three types of death states, labeled with the word ‘DEATH’. MFA.3 assumes that there will not be an asymmetric faulty BIU and an asymmetric faulty RMU at the same time (the leftmost death state – 2, 1, 3, 1). MFA.1 assumes that a majority of BIUs are good (the middle death state – 3, 0, 2, 2) and MFA.2 assumes that a majority of RMUs are good (the rightmost death state – 1, 2, 4, 0). More death state space combinations are possible – for our experiments, the combinations were aggregated by the first MFA violation.

5. Experiment

5.1. Fault Model

The fault model includes permanent and transient faults, and three fault types: asymmetric, symmetric, and benign. The fault model is not intended to represent all possible fault sources. Correlated faults are not included. Reintegration (and reintegration faults) are not considered. Comprehensive proofs of reintegration are not currently available; however, reintegration would be an interesting area of future work. The fault types are tested separately to facilitate comparison and to reduce model complexity. Also, since there is no benefit to convicting benign faulty nodes, it is useful to examine types separately to avoid overemphasizing any benefits of a transient fault tolerance policy. However, the reliability of a combination of types may not equal the weighted percentage of individual type data. Despite these limitations, the study provides insight into the expected reliability of the Maximum Fault Assumption with different conviction policies.

Permanent faults are modeled with a fixed exponential rate per component. A Bit Error Rate (BER) model is used as the transient fault source, where, in our model, each single corrupted bit constitutes a single transient fault. We assume the bit errors occur randomly and independently, so the BER is an exponential rate. Our fault model extends somewhat to correlated faults. In SPIDER as in most other Time Division Multiple Access network protocols, a single corrupted bit within a frame will cause the entire frame to be faulty. Appended error detection codes can be constructed that detect most multiple bit errors in the same frame. Therefore, our fault model accounts for multiple bit errors within one frame, given an error detection code with adequate detection power. The BER fault model does not represent correlated multiple bit errors that span frames.

5.2. Variable Parameters

The list of variable experiment parameters is given in Table 1. The Perfect policy was only tested for a BER range of 10^{-12} through 10^{-9}. For computational efficiency reasons, a separate Markov model was generated for each combination of parameters (called a design point), for a total of 12,600 models. (10 BIU values * 5 RMU values * 12 BERs * 3 Durations * 3 Fault Types * 2 Policies) + (10 BIU values * 5 RMU values * 4 BERs * 3 Durations * 3 Fault Types * 1 Policy). Modeling other ranges of parameters is possible as well; we selected these ranges as most representative of the aerospace/aviation domain.

Table 1. Variable Experiment Parameters

Parameter	Examined Values
BIUs	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
RMUs	1, 2, 3, 4, 5
Bit Error Rate	10^{-20}, 10^{-19}, 10^{-18}, 10^{-17}, 10^{-16}, 10^{-15}, 10^{-14}, 10^{-13}, 10^{-12}, 10^{-11}, 10^{-10}, 10^{-9}
Diagnosis Period/ Transient Duration	1 round, 10 rounds, 100 rounds
SPIDER Fault Types	Asymmetric, Symmetric, Benign
Conviction Policies	All Permanent, All Transient, Perfect

- Bus Interface Unit (BIU)

The number of Bus Interface Units ranged from one through ten. Ten is a conservative number for the expected number of BIUs in a system, since the BIUs map one-to-one with processing elements, and fielded systems are expected to have in the tens of nodes. A maximum of ten BIUs was chosen due to the $O(2^n)$ processing time of the Markov solver, and since adding BIUs did not drastically improve assumption reliability past five or six BIUs. This was probably due, in part, to the decision to apportion the maximum bandwidth among the BIUs instead of having each BIU send a fixed amount of data.

- Redundancy Management Unit (RMU)

The number of RMUs ranged from two through five. The number of RMUs is not expected to exceed five in a fielded system. Three RMUs is the minimum number required to mask a single fault by majority vote (with two RMUs the fault will be detected, but not masked), so fielded systems would likely have three or four RMUs.

- Bit Error Rate (BER)

We analyzed a range of 10^{-12} errors/bits through 10^{-9} errors/bits as the expected aerospace operational range. Data on acceptable and actual BERs can be found from standards and equipment manufacturers, for example, Jain reports that the Fiber Distributed Data Interface (FDDI) standard used for fiber optic local area networks mandates a BER of less than $2.5 * 10^{-10}$ errors/bits [11]. For additional study, we examined lower bit error rates, down to 10^{-20} errors/bits, with data for 10^{-17} errors/bits and higher presented here.

- Diagnosis Rate / Transient Error Duration

In SPIDER, the transient error duration is conservatively determined by the diagnosis period. Any fault-related evidence accumulated by a node is cleared upon each execution of the Diagnosis protocol. Therefore, an error from a transient fault according to the BER model will last, at most, for a time equal to the diagnosis period. In a TDMA round, if each node sends one frame per round, the round length is determined by the shortest frame period in the system. We used a 5 ms round. In SPIDER, the Diagnosis protocol executes periodically at the end of an integer number of rounds, so we selected a range of 1 to 100 rounds for the diagnosis period. We did not include overhead from the Diagnosis protocol, instead using the maximum bandwidth for all design points. We did not investigate transient faults with longer durations than the diagnosis period, as a BER model would not adequately represent those faults.

5.3. Static Parameters

A number of other parameters needed to be specified, which were kept static for these experiments.

- Permanent Fault Arrival Rates

The permanent fault arrival rate was 10^{-5} faults/hour for the BIUs (as the processing element fault rate will dominate) and 10^{-6} faults/hour for the RMUs.

- Data Rate

The maximum data throughput of SPIDER RMUs is currently 1 MBit/sec. We modeled 1 MBit/sec per each RMU, and apportioned the bandwidth among the BIUs, in order to keep the total data rate at 1 MBit/sec. An alternative would be to specify a fixed data rate per BIU, but the total data rate would vary.

- Bit Representation

What constitutes one ‘bit’? In a bus topology, a sender sends one distinct message per bus. In a star topology, a sender sends one distinct message per star, and each star then forwards N distinct messages to each of its N receivers. Should these replicated bits count as multiple bits or as a single bit for BER computations? Since the amount of energy and distance traveled could be the same, we modeled this as a single bit regardless of the topology.

- Mission Time

A one-hour mission time was used, to easily obtain the reliability per hour and to limit the size of the models.

6. Results

This section presents results to three topics of investigation, with these main observations.

- Viable Design Space: For each conviction policy, how reliable is the Maximum Fault Assumption for different system parameter values?

The Maximum Fault Assumption reliability differed drastically for studied parameter configurations, from fewer than 10^{-11} violations per hour to more than 10^{-3} violations per hour. Inadequate assumption reliability is one reason a system may fail, impacting system reliability. Some of these configurations seem inadequate for a safety-critical system, while other configurations perform well. Surprisingly, the ‘All Transient’ policy outperformed the standard ‘All Permanent’ policy for an important range of the design space.

- Policy Trade-off: At what BERs does the ‘All Transient’ policy surpass the ‘All Permanent’ policy?

The ‘All Transient’ policy outperformed the standard ‘All Permanent’ policy for the expected operational BER range of 10^{-12} to 10^{-9} errors/bits. The trade-off points occurred at approximately a BER of 10^{-14} for asymmetric faults and 10^{-15} for symmetric faults. Above these points, the ‘All Transient’ strategy was advantageous. Below these points, the ‘All Permanent’ strategy was advantageous.

- Sensitivity to Fault Type: How does assumption reliability differ with respect to the type of fault incurred (benign, symmetric, or asymmetric?)

The Maximum Fault Assumption reliability also differed greatly depending on the type of fault. MFA reliability was poorest when subject to asymmetric faults, with no design points achieving the best reliability of fewer than 10^{-11} assumption violations per hour for any of the policies.

6.1. Design Space

One goal was to determine the design space in which the Maximum Fault Assumption is expected to have adequate reliability. For this study, we examined the combinations of parameters from Table 1 with a Bit Error Rate of between 10^{-12} and 10^{-9}. Next, we limited this population to design points that might achieve a reliability level of 10^{-9} failures/hour or better. Single BIU or single RMU configurations were omitted here, as the MFA would be violated as soon the single BIU or RMU became faulty. The permanent fault rate of the BIU (10^{-5}) or RMU (10^{-6}) bounds the reliability. Asymmetric and symmetric configurations with two BIUs or two RMUs were also excluded here, as a single faulty BIU or RMU would violate the Maximum Fault Assumption. However, for benign faults, two BIU or RMU systems will function despite a single fault, so these configurations were included here. There were 1,008 design points per conviction policy, for a total of 3,024 points.

Table 2 summarizes overall design space results. For each design point, we calculated the probability that the Maximum Fault Assumption would not hold. Table 2 gives the percentage of design points that fall into six reliability bands, per conviction policy. For example, out of the 1008 design points considered for the 'All Permanent' strategy, 579 points (57.4%) had a 10^{-3} chance per hour or greater of the Maximum Fault Assumption being violated, showing that service reliability is expected to be poor. We observe:

- Wide reliability spread - Design points occupy each of the reliability bands.

Even with the same formal proof foundation, assumption reliability can vary greatly due to choice of design parameters. Therefore, it is essential that the reliability of the assumptions be tested. Otherwise, a designer risks deploying an inadequately reliable system. Recall that obviously unreliable design points were excluded from Table 2, just as a designer would exclude them from consideration.

- At this level of granularity, the 'All Transient' strategy seems superior to the 'All Permanent' strategy.

This is an interesting conclusion, since the 'All Transient' strategy never convicts any nodes - it is basically a no-op strategy. For the parameter range studied (BER of 10^{-12} through 10^{-9} and permanent fault arrival rates of 10^{-5} and 10^{-6}), the results indicate that transient faults are the dominant type of fault. The reliability cost due to lost redundancy is greater than the reliability cost of unconvicted permanently faulty nodes. For these parameters, taking no action seems preferable to an inappropriate conviction.

- The 'Perfect' strategy outperforms the others, but has fairly close distribution to the 'All Transient' strategy.

The perfect strategy is superior by construction, but requires that the service perfectly discriminate between permanent and transient faults. The data show that perfection is not necessary to attain adequate reliability.

6.2. Policy Trade-off

Protocol fault tolerance strategies, including SPIDER's, usually emphasize a permanent fault model. These assumption reliability estimates show that the effects of transient faults are non-negligible, and may dwarf the impact of permanent faults.

For what range of design space is the 'All Transient' policy expected to be better? We expect the trade-off point to depend on the ratio of the transient fault arrival rate vs. the permanent fault arrival rates. Equation 1 approximates the trade-off point. The transient fault arrival rate equals the Bit Error Rate times the data rate (1 MBit/sec), in hours. Nodes participating in group membership typically transmit a frame in each allotted time slot, so we assumed that the full bandwidth is used. The permanent fault arrival rates are $1*10^{-5}$ faults/hour for the BIUs and $1*10^{-6}$ faults/hour for the RMUs. Equation 1 approximates the permanent fault arrival rate as $5*10^{-5}$ faults/hour. Equation 1 assumes each bit corruption causes a unique faulty frame. This is slightly pessimistic, since multiple bit corruptions in the same frame would create only one faulty frame.

$$1) \qquad BER * \frac{1*10^{6}\,bits}{\text{sec}} * \frac{3600\text{sec}}{hour} = \frac{5*10^{-5}\,faults}{hour}$$

Solving this equation for the BER gives a BER of about $1 * 10^{-14}$. One would expect the 'All Transient' policy to perform better at higher BERs and the 'All Permanent' policy to perform better at lower BERs. To test this, we examined 9,450 configurations ranging from a BER of 10^{-17} through 10^{-9}, with each fault type tested independently. For this experiment, we included the 1 and 2 BIU or RMU configurations, as the aim was to determine which policy was better, and not to achieve a fixed reliability goal. For benign faults, the 'All Transient' policy always outperforms the 'All Permanent' policy, since conviction has no benefit.

Figures 3 and 4 summarize data comparing the reliability of the 'All Transient' and 'All Permanent' policies with respect to BER. We used a 'twice as reliable' metric to highlight differences. If one policy had X assumption violations/hour estimated for a particular design point, a competing policy would need an estimate of X/2 assumption

Table 2. Design Space Results

Assumption violations /hour	All Permanent	All Transient	Perfect
More than or equal to 10^{-3}	57.4%	9.0%	9.0%
< 10^{-3} to 10^{-5}	16.7%	19.6%	17.0%
< 10^{-5} to 10^{-7}	12.5%	27.3%	20.5%
< 10^{-7} to 10^{-9}	8.0%	14.2%	21.3%
< 10^{-9} to 10^{-11}	3.3%	4.5%	6.1%
< 10^{-11}	2.1%	25.4%	26.1%

violations/hour to be considered twice as reliable. Since the measurements differed in precision, we did not use statistical significance as a metric. There are 1,050 design points for each BER. For each BER, Figures 3 and 4 show the percent of design points where the 'All Permanent' strategy was twice as reliable, where the 'All Transient' strategy was twice as reliable, and where neither strategy was twice as reliable as the other. Observations include:

- The 'All Transient' policy shows superior assumption reliability for the BER range expected of aviation and aerospace systems (about 10^{-12} through 10^{-9}).The trade-off region occurs approximately where expected, at a BER range of 10^{-14} to 10^{-12} in Figure 3 for asymmetric faults, and a BER range of 10^{-15} to 10^{-12} in Figure 4 for symmetric faults. For larger BERs, the 'All Transient' strategy generally outperforms the 'All Permanent' strategy (for configurations where a difference is observed).
- The trade-off range is slightly different when subject to asymmetric vs. symmetric faults. For asymmetric faults, not convicting a node presents larger risk, as MFA.3 states that any BIU may not be asymmetrically faulty at the same time as any RMU, and vice-versa. The 'All Permanent' conviction strategy outperforms the 'All Transient' strategy at higher BERs compared to the symmetric fault case. For symmetric faults, MFA.1 and MFA.2 require a majority of good BIUs and a majority of good RMUs. An unconvicted permanently faulty node will be less detrimental, as long as there are redundant nodes (i.e., more than 2 BIUs and RMUs).
- For about half of the design points, neither policy outperforms the other, shown by the white bars in Figures 3 and 4. This is true for all BERs. Given a fixed conviction policy, a designer might change other system parameters to maximize assumption reliability.

6.3. Fault Types

Tables 3, 4, and 5 summarize the assumption reliability of design points classified by fault type. These results are for BERs of 10^{-9} through 10^{-12}, omitting configurations bounded by the permanent fault rates of components (the 1 and 2 BIU or RMU configurations for asymmetric and symmetric faults, and the 1 BIU or RMU configurations for benign faults). Because there are more benign configurations, percentages are also given. Actual systems would experience a mix of the three fault types instead of a hundred percent of a single type. If a design point tolerates all three types of faults with acceptable assumption reliability, then it will tolerate any combination of those types of faults. However, this could be extremely conservative, as benign faults typically represent a large percentage of total faults. Future work will investigate mixed fault models.

- Asymmetric (Table 3)

Models subject to 100% asymmetric faults show the lowest overall assumption reliability levels. For the 'All Permanent' policy, no configurations achieved a level of 10^{-9} assumption violations/hour or better. The 'All Transient' policy had no configurations at a level of 10^{-7} assumption violations/hour or better. It is interesting that the 'All Permanent' policy can achieve higher assumption reliability than the 'All Transient' policy. This suggests that the penalty for leaving an asymmetric faulty node in the system can be large. Even for the most pernicious fault type, it might be possible to meet safety-critical assumption reliability requirements. For the Perfect policy, some configurations fall in the 10^{-9} to 10^{-11} assumption violations/hour range. However, the Perfect policy serves as a theoretical bound as this policy requires perfect discrimination between transient and permanent faults. No policy had configurations better than 10^{-11} assumption violations/hour.

Legend:	
□	Neither strategy twice as reliable
▨	'All Permanent' twice as reliable
■	'All Transient' twice as reliable

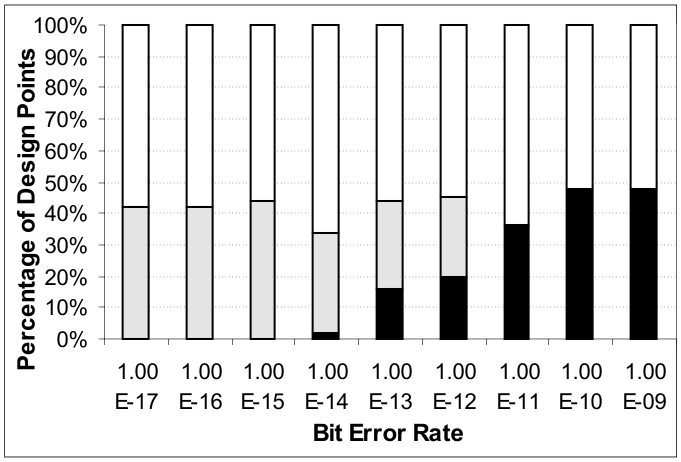

Figure 3. Asymmetric Faults Is One Strategy Twice as Reliable?

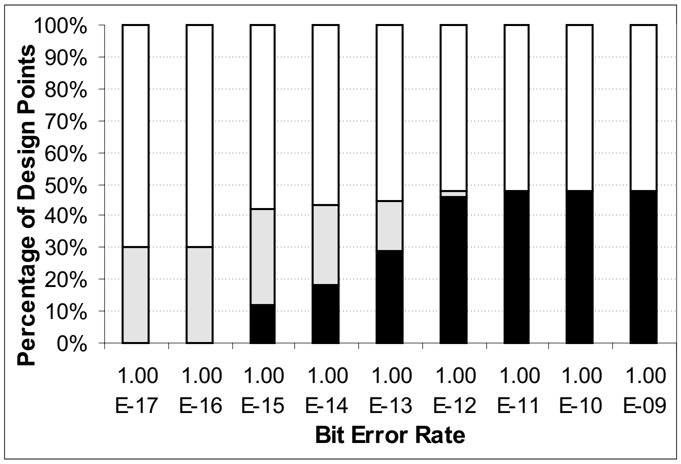

Figure 4. Symmetric Faults Is One Strategy Twice as Reliable?

- Symmetric (Table 4)

  Overall reliability is better here than for the 100% asymmetric fault experiments. The 'All Permanent' policy has a low-end loaded distribution, with over 50% of the design points at a level of 10^{-3} assumption violations per hour or greater. The 'All Transient' policy has the majority of its design points in the 10^{-3} through 10^{-9} range (20.8%, 27.8%, and 22.9%). The 'Perfect' policy has a slightly greater percentage of design points in the highest reliability level than the 'All Transient' policy. All three policies have design points in the highest and the lowest reliability levels, suggesting that apt parameter choice is important.
- Benign (Table 5)

  Benign faults are the least harmful of the three fault types. However, even for benign faults, the 'All Permanent' policy has over 50% of the design points at a level of 10^{-3} assumption violations per hour or greater. The 'All Transient' and 'All Permanent' policies have the same distribution, since convicting benign faulty nodes has no benefit. These two policies have the majority of design points at 10^{-11} assumption violations per hour or better.

7. Related Work

Other researchers underscore the need to critically examine how the assumptions of a formally proven system will withstand expected fault conditions. Powell introduces the concept of assumption coverage, showing the need to balance the risk of uncovered faults versus the risk of increased failure rate due to adding redundant components [13]. Powell defines assumption coverage as "The failure mode assumption coverage (px) is defined as the probability that the assertion X defining the assumed behavior of a component proves to be true in practice conditioned on the fact that the component has failed: px = Pr{X = true|component failed}." [13]. In other words, assumption coverage is a measure of how well actual system faults map to the fault types defined in the assumptions of the fault tolerance mechanisms. Faults that do not map are not 'covered', and the system may fail. Powell illustrates the paradox that weakening assumptions may lower reliability [13]. Weaker assumptions typically require more redundancy, which increases fault rates proportional to the number of components. Cukier and Powell discuss how to use testing data to estimate the assumption coverage in [6].

Bauer, Kopetz, and Puschner note that a measure of how the system withstands multiple faults is needed, even if one implements the weakest assumptions available [2]. In their study of the assumption coverage of the Time Triggered Architecture, they observe that "In general, every fault-tolerant system relies on the existence of a minimum number of correct components. Thus, even an optimal system architecture, which has 100% assumption coverage with respect to the tolerated failure modes, can never have 100% assumption coverage with respect to coincident faults." [2]. Our notion of assumption reliability is a natural complement to assumption coverage, as assumption reliability is the ability of the system to withstand faults with perfect assumption coverage (as defined by [13]). We also show that the choice of system parameters may greatly affect assumption reliability, which is correlated with system reliability.

As motivation for measuring assumption reliability, recent work reports on observed asymmetric/Byzantine faults. Driscoll, Hall, Sivencrona, and Zumsteg report on phenomena that cause Byzantine faults, noting that the probability of Byzantine faults in distributed systems inherently cannot be zero (for example, clocks can never be per-

Table 3. Asymmetric Faults, Assumption Reliability

Assumption violations/ hour	All Permanent	All Transient	Perfect
More than or equal to 10^{-3}	168 (58.3%)	48 (16.7%)	48 (16.7%)
$< 10^{-3}$ to 10^{-5}	54 (18.8%)	99 (34.4%)	72 (25.0%)
$< 10^{-5}$ to 10^{-7}	45 (15.6%)	141 (49.0%)	82 (28.5%)
$< 10^{-7}$ to 10^{-9}	21 (7.3%)	0 (0%)	72 (25.0%)
$< 10^{-9}$ to 10^{-11}	0 (0%)	0 (0%)	14 (4.9%)
$< 10^{-11}$	0 (0%)	0 (0%)	0 (0%)

Table 4. Symmetric Faults, Assumption Reliability

Assumption violations /hour	All Permanent	All Transient	Perfect
More than or equal to 10^{-3}	168 (58.3%)	34 (11.8%)	34 (11.8%)
$< 10^{-3}$ to 10^{-5}	54 (18.8%)	60 (20.8%)	60 (20.8%)
$< 10^{-5}$ to 10^{-7}	45 (15.6%)	80 (27.8%)	71 (24.7%)
$< 10^{-7}$ to 10^{-9}	9 (3.1%)	66 (22.9%)	66 (22.9%)
$< 10^{-9}$ to 10^{-11}	12 (4.2%)	20 (6.9%)	22 (7.6%)
$< 10^{-11}$	0 (0%)	28 (9.7%)	35 (12.2%)

Table 5. Benign Faults, Assumption Reliability

Assumption violations/ hour	All Permanent	All Transient	Perfect
More than or equal to 10^{-3}	243 (56.3%)	9 (2.1%)	9 (2.1%)
$< 10^{-3}$ to 10^{-5}	60 (13.9%)	39 (9.0%)	39 (9.0%)
$< 10^{-5}$ to 10^{-7}	36 (8.3%)	54 (12.5%)	54 (12.5%)
$< 10^{-7}$ to 10^{-9}	51 (11.8%)	77 (17.8%)	77 (17.8%)
$< 10^{-9}$ to 10^{-11}	21 (4.9%)	25 (5.8%)	25 (5.8%)
$< 10^{-11}$	21 (4.9%)	228 (52.8%)	228 (52.8%)

fectly synchronized) [8]. They argue that for safety-critical systems, reducing this probability to an unknown level is inadequate [8]. Ademaj, Sivencrona, Bauer, and Torin give results from fault injection experiments designed to estimate the percentage of certain Byzantine faults (Slightly Off Specification faults and general asymmetric faults) in the Time Triggered Architecture [1]. They produced Byzantine faults in a bus topology (providing some data about the Byzantine fault rate), but did not observe any in a star topology which employs additional fault protection. Assumption reliability could be used to investigate various Byzantine fault percentages, using limited available data instead of requiring a high-precision fault rate figure.

8. Conclusions

Measuring the reliability of a proof's assumptions is a valuable exercise. We modeled the probability that the assumptions of the SPIDER group membership service would be violated for 12,600 possible configurations. In particular, we focused on the impact of transient faults on the system, and examined assumption reliability with respect to three fault types (asymmetric, symmetric, and benign). Analysis at the design stage allows policy trade-offs and predicts how the system will withstand fault occurrences over a wide range of system parameters.

We have presented three examples, using the SPIDER group membership conviction policy as a case study. First, we determined the expected Maximum Fault Assumption violations per hour for 3,024 configurations in the range of expected operational parameters. This varied from fewer than 10^{-11} to more than 10^{-3} assumption violations/hour, where if the Maximum Fault Assumption is violated, the guarantees may not hold and the service may fail. This indicates that even with the same formal proof foundation, actual service reliability may differ greatly. Next, we looked at the trade-off point between two conviction policies for 10,800 configurations, one policy treating all faults as permanent, the other as all transient. The reliability estimates matched the calculated trade-off point. Also, this analysis showed that the all-transient policy is expected to be more reliable than the all-permanent policy for aerospace/aviation Bit Error Rate ranges. Finally, we tested assumption reliability with respect to three fault types - asymmetric, symmetric, and benign - for 3,024 configurations. Assumption reliability subject to asymmetric faults was quite different from the symmetric and benign cases.

This work demonstrates that the assumptions of a formally proven service may be reliable for some system configurations, but not others. In order to claim that a service is reliable, the service's assumptions must also be shown to be reliable. Future work will measure assumption reliability for other safety-critical protocols, including the Time Triggered Protocol, Class C (TTP/C) and the FlexRay protocol.

9. Acknowledgments

This material is based on work supported by the National Aeronautics and Space Administration, Langley Research Center, under the Research Cooperative Agreement No. NCC-1-02043 awarded to the National Institute of Aerospace. This work is supported in part by the General Motors Collaborative Research Laboratory at Carnegie Mellon University and by the United States Department of Defense (NDSEG/ONR).

10. 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 2003)*, June 2003, pp. 123-132.

[2] 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 Automatio*n, Oct. 2001, pp. 333-341.

[3] R. Butler, The SURE Approach to Reliability Analysis, *IEEE Transactions on Reliability*, vol. 41, no 2, June 1992, pp. 210-218.

[4] 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.

[5] G. Chockler, I. Keidar, and R. Vitenberg, Group Communication Specifications: A Comprehensive Survey, *ACM Computing Surveys*, vol. 33, no. 4, Dec. 2001, pp. 427-469.

[6] M. Cukier and D. Powell, Coverage Estimation Methods for Stratified Fault Injection, *IEEE Transactions on Computers*, vol. 48, no. 7, July 1999, pp. 707-723.

[7] Y. Deswarte, K. Kanoun, and J.-C. Laprie, Diversity Against Accidental and Deliberate Faults, *Proc. of Computer Security, Dependability and Assurance*, 1998, pp. 171-181.

[8] K. Driscoll, B. Hall, H. Sivencrona, and P. Zumsteg, Byzantine Fault Tolerance, from Theory to Reality, *Proc. of the 2003 Intl. Conf. on Computer Safety, Reliability, and Security (SAFECOMP 2003),* Sept. 2003, pp. 235-248.

[9] A. Geser and P. Miner, A Formal Correctness Proof of the SPIDER Diagnosis Protocol*, 15th Intl. Conf. on Theorem Proving in Higher Order Logics (TPHOLS)*, Aug. 2002, pp. 71-86.

[10] A. Geser and P. Miner, A New On-Line Diagnosis Protocol for the SPIDER Family of Byzantine Fault Tolerant Architectures, *NASA/TM-2003-212432*, April 2003.

[11] R. Jain, Error Characteristics of the Fiber Distributed Data Interface (FDDI), *IEEE Transactions on Communications*, vol. 38, no. 8, Aug. 1990, pp. 1244-1252.

[12] Analysis and Test of Bus Systems, PALBUS Task 10.2 and 10.3, SP Swedish National Testing and Research Institute, 2001.

[13] D. Powell, Failure Mode Assumptions and Assumption Coverage, *Proc. of the 22nd Annual Intl. Symposium on Fault-Tolerant Computing (FTCS '92)*, July 1992, pp. 386-395.