Abstract—A validation procedure combines field data, arguments-from-design, and fault injection experiments to demonstrate high reliability. This paper covers integrating these elements and deriving results that reduce the fault injection effort. A typical argument about the impossibility of demonstrating ultra reliability by experiment is presented to motivate the emphasis on reducing the experimental effort. An example is used to explain the integration of field data, fault injection, arguments-from-design, and performance monitoring. This validation procedure has stringent requirements, but it is shown that these requirements are common to other procedures. There is an extended analysis on number, type, and time of fault occurrence. These results are applied to a system with an extremely high reliability requirement. They reduce the experimental effort to a moderate level.

TABLE OF CONTENTS
1. Introduction
2. Typical Infeasibility Argument
3. An Illustrative Example
4. Field Data
5. Integrating Performance and Reliability
6. Comparison with Markov Model Approach
7. Probability Derivations
8. Second Example
9. Summary

1. INTRODUCTION

A long standing problem in the field of ultra-reliable digital control systems is the design of a fault injection experiment for system validation. Such an experiment combines arguments-from-design, field-data-on-fault-occurrence, and results-from-fault-injections. If the system successfully completes the experiment, then the system has a certain reliability at a certain confidence level.

The next several sections describe a type of validation experiment based on modifying a natural-life experiment. The procedure is made more efficient by using probability arguments and design arguments. A natural-life experiment requires an enormous number of trials to establish ultra reliability at a high confidence level, but if the system uses quality components then the probability of a component failure during a single trial is small. If there is a design argument that the system operates acceptably when there are no component failures, then there is no need to run the trials with no component failures. Furthermore, a fault injection experiment need last only as long as it takes for the system to remove the faulty component. As explained below, these features may make it possible to establish high reliability with a reasonable effort.

The difficult parts of this procedure are integrating the various elements and deriving the probability results to reduce the number of trials. Section two presents a typical argument that this type of experiment cannot be done. Section three discusses integrating the elements for a concrete example. It concentrates on the relationship between the experiments and the logical arguments. Sections four, and five cover integrating field data and performance into the general procedure. Since this approach has stringent requirements, section six
compares the requirements of this approach with the requirements of the Markov model approach. It is shown that the Markov model approach has the same requirements, although the Markov model approach usually does not state them explicitly. Section seven derives the probability results. These are used to determine the number, type, and time of fault injections. They are also used to reduce the level of experimental effort. Section eight discusses establishing the reliability of a flight control computer for a commercial aircraft. Since the previous example covered integrating the various elements of field data, fault injection, and arguments-from-design, section eight focuses on reducing the experimental effort.

2. TYPICAL INFEASIBILITY ARGUMENT

To motivate the emphasis on reducing the level of effort, consider the following argument.

A proposed requirement for an electronic flight control system is that the chance of system failure be less than one in a billion for a ten-hour flight. To establish this at the $100\left(1 - 10^{-9}\right)$% confidence level requires $2 \times 10^9$ ten-hour flights without failure. A fleet of 1000 planes flying continuous ten-hour missions would require 24,000 years. Not even a six order magnitude gain in efficiency could make this experiment feasible.

The counter is that the argument above treats the system as an unknown black box. In reality, we know a lot about the system (because we built it). First we know the structure, which permits using arguments-from-design and tests. One option is to concoct the arguments-from-design and then construct the system. Second we know the failure rate of the components, which permits conducting quantitative fault injection experiments. The goal below is to use this structural and stochastic knowledge to bring establishing ultra-reliability within reach.

3. AN ILLUSTRATIVE EXAMPLE

This section determines the fault injection experiments needed and the arguments from design needed to establish the reliability of a simple system. These two activities must fit together to have an authentic procedure. After deciding what experiments and arguments are needed, it is a separate matter to design the experiments and concoct the arguments.

Consider a reconfigurable fourplex where each component has a failure rate of $1.24 \times 10^{-4}$ per hour. The flight time is two hours and there is maintenance after each flight that replaces the failed components. Suppose the requirement is that the probability of failure be less than $1 \times 10^{-6}$, and that this be established at the $100(1- 10^{-6})$% confidence level.

The first step is to imagine conducting a natural life experiment. In a natural life experiment, each operating period is a trial. At the end of each trial (operating period), maintenance is performed, and the system is subjected to another trial. If the system successfully completes a large number of trials without failure, then the system can be said to have the required reliability at the required confidence level. If the requirement is $1 \times 10^{-6}$ reliability at the $10^{-6}$ confidence level, then the number of trials required (without failure) is $13,816,000$. (A more extensive analysis is needed if failures appear during the trials. See section seven.)

The first reduction in effort comes from a probability argument that most of the $13,816,000$ trials do not have a fault occurrence. The product of the (total) failure rate and operating time for this system is $0.001$. Hence, for the $13,816,000$ trials

13,816 trials are expected to have one fault occurrence,
7 trials are expected to have two fault occurrences,
<1 trials are expected to have more than two fault occurrences

The number of trials with exactly $k$ fault occurrences is a random variable. The actual number may differ from the expected number, but the figures above indicate that fault injection experiments are within reach.

It can be seen from the discussion above that three arguments from design are needed.

1) If there are no component failures, the system will perform acceptably.
If the system "successfully recovers" from one failure, then the system will perform acceptably.

(3) If the system "successfully recovers" from two failures, then the system will perform acceptably.

An explanation of the phrase "successfully recovers" requires a discussion of arguments from design.

An argument from design proceeds by demonstrating that if a system begins in a certain state and receives certain inputs, then its outputs will be acceptable. From this point of view a "successful recovery" detects the fault, removes the faulty component, and places the system in a certain state (or set of states). The argument from design then considers the state after recovery as a start state. The argument proceeds by showing that if the system starts in this state and receives certain inputs, the outputs will be acceptable.

The validation procedure for this system is being arranged so that the arguments from design can ignore the details of the reconfiguration process. As far as the argument from design is concerned, the start state can be the result of a reconfiguration or it can be the natural start state at the beginning of the operating period.

For this example, the procedure does not require any argument that a fault is detected or that reconfiguration proceeds correctly. This part of the procedure is handled by lab experiment. This procedure does not establish (or require) a 100% diagnostic level, but a system without an extremely high diagnostic level will not pass this test. This level of diagnostics will have to be part of the original design. Similarly for correct reconfiguration. The reconfiguration algorithm also carries the burden of placing the system in the set of states identified by the arguments from design as acceptable start states.

With these arguments from design in place, the only span of operation not covered is the time from fault occurrence to system recovery. The performance of the system during this span is studied by fault injection experiments. Techniques from probability are used to determine the time of fault occurrence, the component that is faulty, and the external environment at the time of fault occurrence. The system is run for a short time before fault injection in order to produce the required external environment and to set up the related internal states. The fault is injected, and the system observed to check that the fault is detected and removed, that performance requirements are met, and that recovery places the system in an acceptable state for continued operation.

If system recovery is fast, little time is spent on the experiments. Suppose a single trial takes one minute. If the procedure is highly automated, then there are about 50 trials per hour or about 1000 trials per day. This rate gives a time of two weeks. If it takes five minutes to conduct a trial, it will take ten weeks.

Suppose the arguments from design and the 13,823 trials have been successfully completed. Then the entire procedure has established that the probability of system failure or occurrence of an undetected fault during a single operating period is less than 1e-6. This has been established at the 1e-6 confidence level.

This validation procedure has established something slightly different from a natural life test. A natural life test addresses the probability of failure during a single operating period with no explicit concern about undetected faults carried over from previous operating periods. The simulated-life procedure needs the additional conclusion about fault detection because it relies on some arguments from design. These arguments from design assume that the system begins an operating period fault free. For this reason, the simulated-life test establishes that if the system begins an operating period with no faults, then there is less than a 1e-6 chance that the system will fail during this period or begin the next period with a latent fault. In other words, the arguments-from-design and the fault injection experiment have shown the system satisfies a more stringent set of conditions than the original requirements. This is a price we pay for replacing natural life testing with structural and stochastic arguments.

A general principle is that a system will have to be over-designed in order to pass a validation test. There are two reasons for this. One reason is that a validation test uses arguments from design to improve on the efficiency of natural life testing. The
system must conform to the additional features required by the arguments from design.

Another reason the system needs to be overdesigned is experimental error and the confidence level requirement. The system will have to be more reliable than required for a reasonable test to establish reliability at a high confidence level. An example of this can be seen from the experiment described above. If a system has a probability of failure of $10^{-6}$, then 14 failures can be expected to occur in 14 million trials. Establishing a failure probability of $10^{-6}$ at a $10^{-6}$ confidence level with only 14 million trials requires that no failures occur.

4. FIELD DATA AND FAULT INJECTIONS

One of the difficult areas in the fault tolerant approach to reliability is gathering field data on the occurrence and behavior of faults. This section considers several possibilities and how they relate to fault injection and system design.

An idealized scenario is that the components are bench tested under the proper environmental conditions until errors are observed on the output pins. The failure rate is estimated and used in the probability arguments (as illustrated in section five). The system is designed to detect failures that appear at the output pins of the devices. In the experiments, faults are injected at the output pins. The actual error pattern that appears at the output pins of a faulty device may be hard to obtain. Hence, part of system design may be a worst case analysis for the detection of any error pattern.

It has been suggested that faults are characterized by open or short conditions at the input pins. If this characterization of faults is accepted, then the system must be designed to detect this type of fault, and the experiments must inject this type of fault.

It is apparent that a system will be designed and validated for a given class of faults. This class of faults will have to be stated (and agreed upon) in the initial stages of design.

5. INTEGRATING PERFORMANCE AND RELIABILITY

This validation procedure incorporates performance almost automatically. There are two cases. The first case considers performance when the system begins in an approved start state and the system does not contain any faults. These start states are the states used by the arguments from design discussed in section two. Performance, for this first case, can be established by a combination of arguments and tests.

The second case considers performance during fault recovery. In this case, performance is established by observation. The period of observation is from the time of fault injection to the time of complete and correct system recovery.

6. COMPARISON WITH THE MARKOV MODEL APPROACH

Four common objections to the approach above are: (1) the need to establish the system works correctly if fault free; (2) the need for nearly 100% diagnostics; (3) the need for between flight maintenance to bring the system to a fault free state; and (4) the need to know the nature of faults in order to inject them. The reply is that these requirements are present in other approaches, but often not stated as explicitly as above. Consider the usual "demonstration" of reliability by a Markov model for a reconfigurable fourplex.

The mnemonics for figure 1 are S for a fault free state, R for a recovery mode state, A for an intermediate state, C for a failure state because of nearly coincident faults, and E for failure by exhaustion of parts. The failure rate is $\lambda$ and the reconfiguration rate is $\rho$.

\[
\begin{align*}
S_1 & \quad \rightarrow R \quad \rightarrow C \\
\quad \rho & \quad \downarrow \quad \quad \quad \downarrow \quad \quad \quad \downarrow \\
S_2 & \quad \rightarrow A \quad \rightarrow E \\
\end{align*}
\]

Figure 1: Markov Model for a Fourplex

The transitions to the failure states are component failure transitions. There is no direct transition from a fault-free state to a system failure state. Such a
transition could be included in the model, but it
would not effective since its density function is not
known. Hence, the model assumes that the system
operates correctly if its components are fault free.

Part of the mathematical basis for semi-Markov
models implies that the recovery function $p$ as
depicted is a complete density function which means
the model assumes 100% diagnostics. Compare this
with the experimental procedure that requires
extremely good diagnostics to be successful, but
does not assume anything about diagnostics.

The model above, which describes the behavior of
the system during a single flight, begins in state $S_1$,
the state where all the components are good. Hence,
the model assumes that between flight maintenance
brings the system to a fault free state.

In order to get the parameters for the recovery
function $p$, fault injection experiments must be
conducted, but these fault injection experiments
require knowledge about the nature of faults.

One approach for modeling suggests that the
engineer can provide (perhaps conservative)
parameters for the recovery function because of his
knowledge of system design. If we accept
engineering judgement as valid then there is no
reason to perform the experiment, but we may be
skeptical about our ability to predict one-in-a-million
or one-in-a-billion events. This skepticism, and our
desire for additional assurance, is the motivation for
conducting experiments.

In summation, the requirements for conducting the
experiment, although stringent, appear to be
common to other approaches.

7. PROBABILITY DERIVATIONS

The experiment consists of a number of trials where
each trial represents (and simulates) one operating
period. The system either completes a trial or fails
(during) a trial. If the system successfully completes
almost all of the trials, and if the trials realistically
correspond to an operating period, then the system
can be said to have the specified reliability at the
required confidence level. The system successfully
completes a trial if it detects and recovers from the
faults while maintaining its performance.

This section discusses five items: (1) the number of
trials, (2) the number of faults per trial, (3)
superfluous fault injections, (4) the time of fault
occurrences, the independence of occurrence time
and fault type, and (5) conditioning to restrict the
number of faults injected during a trial.

The Number of Trials for Binomial Sampling

We wish to establish that the probability of failure
for a system during an operating period is small.
Furthermore, we wish to establish this small
probability of failure at a confidence level of $100(1-
\gamma)\%$. That is, there is less than a $100\gamma\%$ chance that
the experiment has misled us.

Since any one trial for the system has an outcome of
either success or failure, the probability distribution
is binomial. Let the probability of failure during a
single trial be $p$. We wish to show that $p$ is small
at a high confidence level. Let $NF$ be the number
of failures observed during $n$ trials. The probability
that there are no failures during an experiment
consisting of $n$ trials is

$$P\{NF = 0\} = (1 - p)^n$$

The larger the value of $p$, the smaller the probability
that no failures will be observed. Hence, this
expression will yield a one-sided confidence
interval. That is, if no failures appear during a large
number of trials, then the likelihood that the
probability of failure $p$ is greater than or equal to a
certain quantity is very small. For a given
probability $p$ and a given confidence level $1 - \gamma$, the
number of trials required is given by the equation

$$(1 - p)^n = \gamma$$

which gives

$$n = (\log \gamma) / \log(1 - p)$$

For extremely high reliability requirements, it seems
appropriate to request a high confidence level.
We'll require that the experiment have a confidence
level equal to the reliability requirement. Some
sample calculations follow.

For $p = 1e-6$ and $\gamma = 1e-6$, $n = 1.3816 \times 10^7$;
For \( p = 1\times10^{-9} \) and \( \gamma = 1\times10^{-9} \), \( n = 2.0723 \times 10^{10} \).

The binomial expression can be extended to handle failures during the trials. For example, the probability of zero or one failure during \( n \) trials is

\[
P\{NF = 0 \text{ or } 1\} = (1 - p)^n + np(1 - p)^{n-1} \quad (4)
\]

For zero or one failure, establishing reliability when \( p = 1\times10^{-6} \) and \( \gamma = 1\times10^{-6} \) requires that \( n = 16.7\times10^6 \).

**Number of Faults During a Single Trial**

This part of the design for the experiment determines how many faults to inject during a single trial. There are several methods of determining this number, but the theory of Poisson renewal processes is the one used below because of its simplicity. Suppose there are \( N \) classes of faults where the total fault rate within each class is \( \lambda_1, \ldots, \lambda_N \). If \( \alpha \) is the total fault rate for the entire system, then

\[
\alpha = \alpha_1 + K + \alpha_N \quad (5)
\]

The Poisson formula gives the probability of \( m_1 \) faults of type \( i \) (\( i = 1, \ldots, N \)) occurring in the interval \([0, T]\) as

\[
\frac{(\alpha_1 T)^{m_1} e^{-\alpha_1 T}}{(m_1)!} L \frac{(\alpha_N T)^{m_N} e^{-\alpha_N T}}{(m_N)!} \quad (6)
\]

where some (or all) of the \( m_1 \) can be zero.

The actual number of faults is a random number chosen according to the Poisson process, but it is possible to compute the expected number of faults to judge the experimental effort required. If there are \( n \) trials, then the expected number of trials with \( m_1 \) faults of type \( i \) (\( i = 1, \ldots, N \)) is \( n \) times the expression in (6).

**Superfluous fault injections**

The fault injection procedure will have to be adapted to the assumption of a constant rate used by the Poisson process. Since the system removes failed components, the failure rate of the system does not remain constant. One method of handling this is to treat the removed components as virtual components. This means that the component is theoretically subject to later fault injections, but in practice these faults will not be injected if the system has already removed the component. If the system has not yet removed the faulty component, then the second fault can be injected into the same component. This double injection checks that the occurrence of a second fault does not interfere with the detection and removal of a faulty component.

**Time and order of fault injection**

In addition to the convenient formula above for the number of faults, the Poisson renewal process also has a nice distribution for the conditional arrival times. To begin consider ordered uniform sampling. First choose \( n \) samples from the uniform distribution on the interval \([0, T]\), say \( x_1, \ldots, x_n \). Second, order the sample such that \( x_{(1)} < \ldots < x_{(n)} \). The probability that \( x_{(1)} < s_1, \ldots, x_{(n)} < s_n \) for \( 0 < s_1 < \ldots < s_n < T \) is

\[
\frac{n!}{T^n} \int_0^T \int_0^{s_1} \ldots \int_0^{s_{n-1}} L \int_0^{s_n} K \ dt_n K \ dt_2 K \ dt_1 \quad (7)
\]

We will show that the (properly conditioned) arrival time for any combination of faults during the operating time \( T \) is given by the ordered uniform distribution.

Suppose \( m_i \) faults of type \( i \) have occurred in some specified order with \( m_1 + \ldots + m_K = K \). Let \( \alpha_i \) be the rate of occurrence for faults of type \( i \). Let \( \delta_1 \) be the rate of the \( j \)-th fault that occurs. Hence,

\[
\delta_1 \delta_2 L \delta_K = (\alpha_1)^m_1 (\alpha_2)^m_2 L (\alpha_N)^m_N \quad (8)
\]

The probability that the \( K \) designated faults occur in the designated order and that the \( j \)-th fault occurs before time \( s_j \) is given by

\[
\int_0^{s_1} \int_0^{s_{1-1}} \ldots \int_0^{s_{n-1}} e^{-(\alpha_1 t_1)} \int_0^{s_{n+1}} e^{-(\alpha_N t_n)} M \quad (9)
\]

\[
\int_0^{s_1} \int_0^{s_{1-1}} \ldots \int_0^{s_{n-1}} e^{-(\alpha_i t_i)} \ dt_i \ dt_{n-1} \ dt_1
\]

where, as before, \( \alpha \) is the total fault rate of the system. The expression (9) is equal to
The probability that \( m_i \) faults of type \( i \) have occurred is given by (6). Hence the probability that \( m_i \) faults of type \( i \) have occurred in some specified order given that \( m_i \) faults of type \( i \) have occurred is given by dividing (10) by (6) and equals

\[
\frac{m_i! \cdot m_2! \cdot m_N!}{T^K} \prod_{k=1}^{K} \int_{t_1}^{t_{K-k+1}} e^{-\alpha_k t} \ dt_k \ dt_2 \ dt_1
\]

This expression is independent of the order in which the faults occurred which implies all orderings are equally likely. Furthermore, there are

\[
K! \quad m_1! \ m_2! \cdot L \ m_N! \quad (12)
\]
different orderings.

As an illustrative computation, the example in the next section will need the probability that there are three faults present in the system given that two permanents and one transient fault have occurred during the operating period. Suppose the operating period has been normalized to 1 time unit, and that any transient lingers for at most \( \delta \) time units. The possible orderings are PPT, PTP, and TPP, and the required probability is given by the total sum of

\[
\begin{align*}
\text{[PPT]} & \quad 2! \ m_1! \ m_2! \ m_3! \\
\text{[PTP]} & \quad 2! \ m_1! \ m_2! \ m_3! \\
\text{[TPP]} & \quad 2! \ m_1! \ m_2! \ m_3!
\end{align*}
\]

which in unreduced form is equal to

\[
\frac{1}{3} + \delta(1-\delta)^2 + \delta^2(1-\delta) + \frac{\delta^3}{3} + \delta^2(1-\delta) + \frac{\delta^3}{3} \quad (14)
\]

Independence of time and type

As before, let \( \alpha_i \) be the total rate of occurrence for all faults of type \( i \). Let \( \delta_i \) be the rate for the \( j^{th} \) fault that occurs. The probability that \( m_i \) faults of type \( i \) have occurred in any order whatsoever (with \( m_1 + \ldots + m_N = K \)) is equal to

\[
\frac{K!}{(m_1)! \cdot \ldots \cdot (m_N)!} \prod_{k=1}^{K} \int_{t_1}^{t_{K-k+1}} e^{-\alpha_k t} \ dt_k \ dt_2 \ dt_1
\]

The first factor gives the number of combinatorial arrangements that are indistinguishable. The integral expression gives the probability that a single arrangement occurs in the specified order before the designated times. The integrands simplify to give

\[
\begin{align*}
\text{[PPT]} & \quad 2! \ m_1! \ m_2! \ m_3! \\
\text{[PTP]} & \quad 2! \ m_1! \ m_2! \ m_3! \\
\text{[TPP]} & \quad 2! \ m_1! \ m_2! \ m_3!
\end{align*}
\]

which is algebraically identical to

\[
\frac{K!}{(m_1)! \cdot \ldots \cdot (m_N)!} \left\{ \frac{(\alpha_1 T)^{m_1}}{(m_1)!} e^{-\alpha_1 T} \right\} \left\{ \frac{(\alpha_N T)^{m_N}}{(m_N)!} e^{-\alpha_N T} \right\}
\]

The first factor is the probability that \( m_i \) faults of type \( i \) occur. The second factor is the probability that \( K \) events occur before times \( s_1 \) through \( s_K \) in the ordered uniform distribution. Hence, the time of occurrence is independent of the type of faults that occur.
Conditioning to Improve Efficiency

This subsection discusses using knowledge about system structure and performance to improve the efficiency of the experiment. As described above, for each trial the number of injected faults is chosen by random sampling. Likewise, the time and place for fault injection.

The fact that the number of trials with \( k \) fault occurrences is a random variable introduces two problems. The first, more serious, problem is that it is possible to randomly choose a large number of faults for a single trial. For the fourplex example, any trial with three or more faults has the potential for causing the system to fail. A few such incidents can be handled by sequential testing, but it is more efficient to eliminate them. The second problem is that the random selection can produce a much larger number of trials than expected, increasing the time and expense of the experiment. It can also produce a much smaller number, but this can cause concern about the entire experiment being marginal because of an unusual run of random numbers. Both these problems are handled by partitioning and conditioning.

Conditioning can also be used to exclude nearly coincident fault occurrences. The conservative approach used here is to treat the excluded occurrences as system failures, just as too many occurrences are treated as system failure.

The major concern is to preserve the confidence level of the experiment while conditioning. We need the standard result below on combining confidence intervals.

**Classic Theorem** [Wilkes [1]] Suppose each of the estimators \( \sigma_1, ..., \sigma_k \) has the confidence interval \([a_i, b_i]\) at the \(100(1-h_i)\%\) level, then the multivariable estimator \((\sigma_1, ..., \sigma_k)\) has the multivariable confidence interval \([a_1, b_1] \times ... \times [a_k, b_k]\) at the \(100(1-h_1 - ... - h_k)\%\) confidence level.

The procedure divides fault occurrences during a trial (a simulated operating period) into three classes. The first class, labeled BF for benign faults, are those that will not cause system failure. Examples: (1) the occurrence of zero faults during an operating period, (2) the occurrence of a few faults in a nonreconfigurable, Byzantine resilient, system, and (3) the occurrence of transient faults sufficiently far apart. The second class, labeled EF for experimental faults, are those to be studied by experiment. The third class, labeled FF for fatal faults, are those we declare to cause system failure upon occurrence. Examples: (1) an extremely large number of fault occurrences during an operating period, and (2) nuisance faults that are unlikely to occur, may or may not cause system failure, but are difficult to include in the experiment. Of course, the total probability of all faults in class FF must be very small.

There are three comments about this partition. First, all the experimental error is concentrated in the second class. The first class has zero chance of failure at the 100\% confidence level. The third class will be assumed to have 100\% chance of failure at the 100\% confidence level. Second, more refined partitions have been examined, but (for the examples to date) there has been no gain in efficiency. The problem is that the number of trials for the larger number of faults is too small to get good confidence intervals. Third, the description of the partition is a little optimistic about the division between the second and third classes. It assumes that the system is well designed and that the occurrence of more faults than the system can handle is extremely unlikely. The approach taken in this initial attempt to construct a validation experiment is that the experiment will be conducted only if the preliminary probability analysis shows the system is well designed—that the occurrence of more faults than the system can handle is extremely unlikely. Otherwise, the system will be redesigned before the experiment is undertaken.

To begin the derivation, let \( P\{\text{Failure}\} \) be the probability of system failure that we are trying to establish at the \(100[1-h]\%\) confidence level. Suppose \( P\{\text{BF}\} \), \( P\{\text{EF}\} \) and \( P\{\text{FF}\} \) are the probabilities of the benign, experimental, and fatal fault occurrences as described above. Partitioning gives the conditional probability expansion

\[
P\{\text{Failure}\} = P\{\text{Failure} \mid \text{BF}\} P\{\text{BF}\} + P\{\text{Failure} \mid \text{EF}\} P\{\text{EF}\} + P\{\text{Failure} \mid \text{FF}\} P\{\text{FF}\} \tag{18}
\]
Since \( P(\text{Failure} | BF) = 0 \) and \( P(\text{Failure} | FF) = 1 \),
\[
P(\text{Failure}) = \frac{P(\text{Failure} | BF) P(BF)}{P(BF)} + P(\text{FF}). \tag{19}
\]

The 0 and the \( P(\text{FF}) \) on the right side have 100% confidence. The \( P(\text{Failure}) \) on the left is the reliability requirement. Hence, it is necessary to establish that
\[
P(\text{Failure} | EF) = \frac{P(\text{Failure}) - P(\text{FF})}{P(EF)} \tag{20}
\]
is less than \( P(\text{Failure}) \) at the 100(1 - \( h \))% level.

**Example**—For the fourplex in section two,
\[
P(\text{Failure}) = 1 \times 10^{-6}
\]
Confidence level = 100(1 - 1e-6)%
\[
P(BF) = P(0 \text{ faults}) = 0.999
\]
\[
P(FF) = P(3 \text{ or more faults}) = 1.66e-10
\]
\[
P(EF) = 1 - [P(BF) - P(FF)]
\]
Solving for \( n \) in the equation \((1-r)^n = h\) where
\[
h = 1 \times 10^{-6}
\]
\[
r = \frac{P(\text{Fail}) - P(\text{FF})}{P(EF)}
\]
Gives \( n = 1.3809e+4 \) trials. This number is smaller than the expected number of trials computed in section two because this last computation includes the information that zero faults produces no system failures. More important, there will be exactly 1.3809e+4 trials with faults. (The number of trials with one fault and the number with two faults are still random.)

### 8. Second Example

This section proposes an architecture and conducts the probability analysis for a system whose requirement is less than one chance in a billion of failure during a ten hour flight. The analysis includes transient faults. The architecture is a nonreconfigurable sevenplex where each module consists of a computer-on-a-chip plus six transmission lines to the other modules. The system is assumed to have Byzantine resilience, which means the system can tolerate and correctly identify up to two faulty components. The system is also assumed to have transient recovery that removes any effects of a transient fault within 15 seconds. The permanent and transient rates are

| Device | Perm Rate | Tran Rate |
|--------|-----------|-----------|
| computer | 1 e-6 per hour | 1 e-5 per hour |
| line | 1 e-5 per hour | 1 e-4 per hour |

These are nominal rates for illustrative purposes. A typical figure for a chip is 1 e-6 per hour. The transmission link consists of several devices in series, and is given a rate an order of magnitude greater. A rule of thumb is that transients are an order of magnitude more frequent than permanents. The permanent fault rate for the system is 0.00427/(10 hours); the transient rate is 0.0427/(10 hours); and the total rate is 0.04697/(10 hours).

The requirement is to establish that there is less than one chance in a billion of failure during a ten hour flight, and to establish this at the 100(1 - le9)% confidence level. That is, there is less than one chance in a billion that the experiment has mislead us. The probability analysis concentrates on the number of fault injections required since reducing this number is necessary for feasibility.

**Computational procedure and example**

We use two features of the system (together with the derivations in section seven) to reduce the number of trials, and hence the number of fault injections. First the Byzantine algorithm implies we need consider only those trials that have three or more faults present in the system at one time. Second the transient fault recovery (of at most 15 seconds) implies most transient fault occurrences will be isolated.

As an example of the latter, we’ll compute the probability of three coincident faults given three transients have occurred. Normalizing the operating period to 1 gives the transient recovery time of \( \delta = (15 \text{ seconds})/(10 \text{ hours}) = 1/2400 \). The expression is
\[
3! \frac{l_{i_1}^{l_{i_1}+\delta}}{l_{i_1}^{l_{i_1}+\delta}} \frac{l_{i_2}^{l_{i_2}+\delta}}{l_{i_2}^{l_{i_2}+\delta}} + 3! \frac{l_{i_1}^{l_{i_1}+\delta}}{l_{i_1}^{l_{i_1}+\delta}} \frac{l_{i_2}^{l_{i_2}+\delta}}{l_{i_2}^{l_{i_2}+\delta}}
\]
\[
= 3\delta^2 - 2\delta^3 = 5.21 \times 10^{-7} \tag{21}
\]
Hence, the probability of three coincident faults given three transients is small. The computation for three coincident faults given 2 permanents and 1 transient is given in section seven by equation (13). The other computations are similar, and will not be displayed.

The trial reduction procedure for this system decomposes the faults into the three classes described in the previous section: (1) the benign occurrences, BF, where the occurrences cannot cause system failure because of the Byzantine algorithm, (2) the experimental occurrences, EF, where it must be established by experiment that the occurrences do not cause system failure, and (3) the fatal occurrences, FF, where it is assumed the occurrences will cause system failure. This decomposition will be carried out in stages.

Large differences among the probabilities require retaining their numerical values to many decimal places. This accuracy is preserved in the tables. The narrative uses approximations.

First step: overall computation

Table 2 displays the probabilities for the occurrence of (i permanents, j transients) during a trial (a ten hour flight). All the entries labeled P{ fault occurrences } are rare events that are placed in the fatal fault, FF, category. All entries with less than three faults are placed in the benign fault, BF, category. The rest receive closer examination in the subsections below.

Second step: analysis for three faults

This step considers the cases (2p,1t), (1p,2t), and (0p,3t). We need only run a trial if there are three faults present in the system at one time.

The probability that three faults are present at one time for (0p,3t) is displayed in (*) with the value \(=5.2 \times 10^{-7}\). Hence, \(P(BF)=1.2 \times 10^{-5}\), \(P(EF)=6.4 \times 10^{-12}\), and \(P(FF)=0\).

The probability that three faults are present at one time for (1p,2t) is \(=4.2 \times 10^{-4}\). Hence, \(P(BF)=3.7 \times 10^{-6}\), \(P(EF)=1.6 \times 10^{-9}\), and \(P(FF)=0\).

The computation for the probability that three faults are present at one time for (2p,1t) is given by (14).

The numerical value is \(=0.33\). Hence, \(P(BF)=2.5 \times 10^{-7}\), \(P(EF)=1.2 \times 10^{-7}\), and \(P(FF)=0\).

Table 2: Global Occurrence Probabilities

| (p,t)  | Probability     |
|--------|-----------------|
| (0p,0t)| 0.954116020641642 |
| (1p,0t)| 0.00407407540814  |
| (2p,0t)| 8.698150996378502e-6 |
| (3p,0t)| 1.238036825151207e-8 |

| (p,t)  | Probability     |
|--------|-----------------|
| (0p,1t)| 1.322733762798567e-11 |
| (1p,1t)| 0.040740754081398 |
| (2p,1t)| 1.7396301992757e-4 |
| (3p,1t)| 3.71411047545362e-7 |

| (p,t)  | Probability     |
|--------|-----------------|
| (0p,2t)| 5.286417243395653e-10 |
| (1p,2t)| 5.647807152780061e-13 |
| (2p,2t)| 8.698150996378502e-4 |

| (p,t)  | Probability     |
|--------|-----------------|
| (0p,3t)| 1.12985594508703e-11 |
| (1p,3t)| 8.10607496885537e-12 |
| (2p,3t)| 5.286417243395653e-8 |

| (p,t)  | Probability     |
|--------|-----------------|
| (0p,4t)| 1.128650081464972e-10 |
| (1p,4t)| 1.238036825151207e-5 |
| (2p,4t)| 3.71411047545362e-7 |
| (3p,4t)| 5.643250407324859e-10 |

| (p,t)  | Probability     |
|--------|-----------------|
| (0p,5t)| 1.128650081464972e-9 |
| (1p,5t)| 1.238036825151207e-5 |
| (2p,5t)| 8.11607496885537e-12 |
| (3p,5t)| 2.5e-7 |
| (4p,5t)| 1.2e-7 |
| (5p,5t)| 0 |

Third step: analysis for four faults

The computation for the probability that three faults are present at one time for (2p,1t) is given by (14).
We consider the cases \((0p,4t), (1p,3t), (2p,2t),\) and \((3p,1t)\). A set of faults is called coincident if they all occur within a time span of \(\delta\). By a multiple integral calculation, the probability of three or four coincident faults given four faults is \(\approx 2 \times 10^{-6}\).

For the cases \((0p,4t), (1p,3t),\) and \((2p,2t),\) if there are only two coincident faults, then there are at most two faults in the system at any one time, which can be handled by the Byzantine and transient recovery algorithms.

The case \((0p,4t)\) has probability \(1.3 \times 10^{-7}\). Hence, \(P(BF) = 1.3 \times 10^{-7}, P(EF) = 0,\) and \(P(FF) = 2.6 \times 10^{-13}\).

The case \((1p,3t)\) has probability \(5.3 \times 10^{-8}\). Hence, \(P(BF) = 5.3 \times 10^{-8}, P(EF) = 0,\) and \(P(FF) = 1.1 \times 10^{-13}\).

The case \((2p,2t)\) has probability \(7.9 \times 10^{-9}\). Hence, \(P(BF) = 7.9 \times 10^{-9}, P(EF) = 0,\) and \(P(FF) = 1.6 \times 10^{-14}\).

The case \((3p,1t)\) is handled differently. The probability of four coincident faults is \(\approx 0.252\). Hence, \(P(BF) = 0, P(EF) = 4.0 \times 10^{-10}, P(FF) = 1.3 \times 10^{-10}\). That is, if the transient occurs after the third permanent (or 15 seconds before the third permanent), then the trial belongs to the FF category because there will be four faults present in the system. Otherwise, the trial will be performed.

**Fourth step: analysis for five faults**

We examine the cases: \((0p,5t), (1p,4t),\) and \((2p,3t)\) and say (as before) that faults are isolated if their occurrence times are further apart than \(\delta\). The probability that all five faults are isolated given that five faults have occurred is \(\approx 0.9979\).

Applying this to the cases \((0p,5t)\) and \((1p,4t)\), we will say that there is benign fault occurrence if all faults are isolated since this implies two or less faults present at any time. Otherwise it is considered a fatal fault occurrence. For the case \((0p,5t)\) this gives \(P(BF) = 1.1 \times 10^{-9}, P(EF) = 0,\) and \(P(FF) = 2.2 \times 10^{-12}\). For \((1p,4t)\) it gives \(P(BF) = 5.6 \times 10^{-10}, P(EF) = 0,\) and \(P(FF) = 1.1 \times 10^{-12}\).

Applying this to the case \((2p,3t)\), we will consider any occurrence where the faults are not isolated as fatal. We will still conduct trials, but only accept isolated fault occurrences. Hence there will never be more than three faults present in the system. For \((2p,3t)\) this gives \(P(BF) = 0, P(EF) = 1.1 \times 10^{-10},\) and \(P(FF) = 2.2 \times 10^{-13}\).

**Summary for analysis**

Table 3 summarizes the three subsections above. Table 4 gives the fault category probabilities.

| Case    | BF                | EF                | FF                |
|---------|-------------------|-------------------|-------------------|
| \((3p,0t)\) | 0                 | 1.238036825151207e-8 | 0                 |
| \((2p,1t)\) | 2.488454018553925e-7 | 1.225656456899695e-7 | 0                 |
| \((1p,2t)\) | 3.712550549053929e-6 | 1.5599239969052e-9  | 0                 |
| \((0p,3t)\) | 1.238036181372058e-5 | 6.437791490786276e-12 | 0                 |
| \((3p,1t)\) | 3.954240098059948e-10 | 1.332177145335705e-10 | 0                 |
| \((2p,2t)\) | 7.929610005841749e-9  | 0                 | 1.585925173018696e-14 |
| \((1p,3t)\) | 1.321601667640291e-7 | 0                 | 5.286406670561166e-8 |
| \((0p,4t)\) | 2.643208621697826e-13 | 0                 | 1.057283448679131e-13 |
| \((2p,3t)\) | 1.126279916293896e-10 | 2.370165171076441e-13 | 0                 |
| \((1p,4t)\) | 5.631399581469477e-10 | 0                 | 1.18508258553822e-12 |
| \((0p,5t)\) | 1.126279916293896e-9  | 2.370165171076441e-12 | 0                 |

**Table 4: Fault Category Probabilities**

| Category | Probability   |
|----------|--------------|
| BF       | 0.99999986280277 |
| EF       | 1.370204301340983e-7 |
| FF       | 1.767996391850492e-10 |
Using formula (20) gives the total number of trials as \( n = 3439 \). The expected number of trials for each fault condition is given in table 5. Only a moderate effort is required because the system was designed for validation.

| Condition | Expected number of trials |
|-----------|---------------------------|
| (3p,0t)   | 311                       |
| (2p,1t)   | 3076                      |
| (1p,2t)   | 39                        |
| (0p,3t)   | 0                         |
| (3p,1t)   | 10                        |
| (2p,3t)   | 3                         |

Table 5: Expected Number of Trials

9. SUMMARY

This paper outlines a validation procedure that consists of integrating various elements: field data, system diagnostics, proofs and tests, and fault injection experiments. There has been considerable effort in all these areas, but no previous attempt to rigorously establish ultra reliability, perhaps because such an effort has appeared impossible. The exposition above demonstrates it may be within reach. It depends on a careful integration of the various elements. The main topics above are a discussion of integrating the elements and a derivation of results to reduce the fault injection effort.

The major point is dividing validation into two parts: structural and stochastic. The structural part consists of proofs and tests that the system operates correctly if all the components are fault-free. It also includes proofs and tests that the system can, up to a point, identify and tolerate misbehaving components. It appears, however, to be beyond the reach of structural arguments to predict successful reconfiguration or demonstrate the system can handle a large number of faults. These items are more suitable for demonstration by experiment. It is shown how to demonstrate this at an extremely high confidence level.

An example illustrates that if a system is designed for validation then a moderate fault injection effort can establish ultra reliability at an equivalent confidence level.

REFERENCES

[1] S. Wilks, *Mathematical Statistics*, Wiley, New York, 1963.

[2] S. Karlin and H. Taylor, *A First Course in Stochastic Processes*, Academic Press, New York, 1975.

Allan L. White is a research engineer at NASA Langley. His interests and publications include probability models, design of experiments, and stochastic control.