Design of Reliable Software via General Combination of N-Version Programming and Acceptance Testing

Behrooz Parhami
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA
parhami@ece.ucsb.edu

Abstract

N-version programming (NVP) and acceptance testing (AT) are techniques for ensuring reliable computation results from imperfect software. Various symmetric combinations of NVP and AT have also been suggested. We take the view that one can insert an AT at virtually any point in a suitably constructed multi-channel computation graph and that judicious placement of ATs will lead to cost-effective reliability improvement. Hence, as a general framework for the creation, representation, and analysis of combined NVP-AT schemes, we introduce MTV graphs, and their simplified data-driven version called DD-MTV graphs, composed of computation module (M), acceptance test (T), and voter (V) building blocks. Previous NVP-AT schemes, such as consensus recovery blocks, recoverable N-version blocks, and N-self-checking programs can be viewed as special cases of our general combining scheme. Results on the design and analysis of new NVP-AT schemes are presented and the reliability improvements are quantified. We show, e.g., that certain, somewhat asymmetric, combinations of M, T, and V building blocks can lead to higher reliabilities than previously proposed symmetric arrangements having comparable or higher complexities.

1. Introduction

Applications of highly dependable computer systems are no longer limited to exotic space exploration and defense systems. A multitude of information processing and control functions in avionics, high-speed transportation, process monitoring, and transaction-based systems also need ultrareliable computational resources. With the continually increasing complexity of hardware and software systems, and the attendant impossibility of building absolutely defect- and fault-free components, the use of multi-channel computations with *design diversity* may emerge as a practical and cost-effective approach. Design diversity has been found useful for hardware subsystems and for data as well, but its primary application area is in constructing highly reliable software systems based on one of two distinct paradigms: Voting on multiple versions and acceptance testing [2], [5].

N-Version Programming (NVP) was proposed to allow tolerance of software design faults [1]. In NVP, several program modules are independently developed and executed, with the final result is obtained by voting on the module results. Voting, as used here, covers a wide variety of techniques in terms of sophistication, flexibility, and computational complexity and need not be implemented via simple matching and majority rule [7].

Use of acceptance tests (ATs), as proposed in the recovery-block scheme [8], is also based on design diversity. Our confidence in an *accepted* result being actually *correct* depends on the thoroughness (coverage) of the acceptance test and its own reliability. ATs come in many different forms; from simple reasonableness checks to complex, high-coverage validators. Alternate modules can be invoked sequentially in some specified order until one has produced a result passing the AT.

Several researchers have attempted to combine NVP and AT techniques. One such attempt is that of consensus recovery blocks (CRB) in which *n* versions are executed and their results are compared. If there is agreement between two or more versions, then their result is assumed correct and used. Otherwise, the *n* disagreeing results are subjected to an AT in some prespecified order and the first one to pass the test is taken as the correct output [9]. Another example is that of *recoverable N-version blocks* (RNVB) [3], also known as *N-self-checking program* [4], in which module outputs are subjected to ATs and only those that pass the AT are provided to the voter.

Clearly, the above are just examples of the ways in which NVP and AT approaches can be combined. The CRB approach essentially applies NVP (with relaxed 2-out-of-*n* voting) and AT schemes sequentially and one at a time. Because no AT is applied in the case of, say, two agreeing results, there is some chance of an erroneous output being propagated. Furthermore, a common AT is assumed and design diversity is not applied to the AT. The RNVB/NSCP approach applies ATs uniformly to all versions. This increases the complexity considerably since ATs may essentially be duplicates of the computational modules.

We take the view that one can insert an AT at virtually any point in a suitably constructed multi-channel *computation graph* and that judicious placement of ATs will lead to cost-effective reliability improvement.

2. Definitions and Assumptions

The question that we have set out to answer is how to combine the techniques of N-version programming and acceptance testing in an optimal way in order to achieve the best possible results. More specifically, our ultimate goal is to be able to combine diverse software modules, acceptance tests, and voting algorithms in a systematic way in order to maximize the correctness probability of the output with a given overall complexity or to achieve a desired correctness probability at minimal cost.

0-8186-7707-4/96 $5.00 © 1996 IEEE

Unfortunately, due to difficulties in estimating reliability and cost parameters, the above problems are currently intractable when posed in their full generality. So, we begin with a limited set of more specific questions based on simplifying assumptions. The following definitions/ assumptions are needed in our discussions and analyses.

2.1. Definition — *MTV graph*: An MTV (Module-Test-Voter) graph is a directed acyclic graph with one "In", one "Out", and possibly one "Err" (error condition) node, plus any number of nodes of three other types: Module (M), acceptance test (T), and voter (V).

M: Computes a result data object based on its inputs and sends it to some T or V node or to Out.

T: Accepts its input and forwards it along its output arc(s) or rejects it and activates some M or T nodes.

V: Forwards the result of weighted plurality voting to other nodes or activates some M or T nodes.

The inner workings of M and T nodes are application-dependent. We make no assumption about these nodes except that they have known reliability parameters. The nodes are connected by directed edges representing data transfers and controls (activations). ∎

2.2. Assumption — *Reliability parameters of computation modules*: Each computation module M_i produces a result which is correct with fixed probability q_i and incorrect with fixed probability p_i, uniformly over its input space. In case $q_i + p_i < 1$, the module may be viewed as (partially) self-checking or fail-safe, abstaining from producing any result with probability $1 - q_i - p_i$. In the rest of this paper, we assume $q_i + p_i = 1$. ∎

2.3. Assumption — *Reliability parameters of acceptance tests*: A correct result passes an acceptance test T_i (outgoing edge labeled "P" is taken) with fixed probability q'_i and fails it (outgoing edge labeled "F" is taken) with fixed probability p'_i, uniformly over the space of correct inputs. Similarly, T_i rejects an incorrect result with fixed probability q''_i and accepts it with fixed probability p''_i, uniformly over the incorrect input space. When $q'_i+p'_i<1$ or $q''_i+p''_i<1$, the AT may be viewed as (partially) self-checking or fail-safe, abstaining from judging an input with probability $1 - q'_i - p'_i$ for correct inputs and $1 - q''_i - p''_i$ for incorrect ones. In the rest of this paper, we will assume $q'_i + p'_i = q''_i + p''_i = 1$. ∎

The reason we do not start by assuming $q'_i = q''_i$ is that ATs behave asymmetrically with respect to correct and incorrect inputs. An AT that is itself defect-free, always accepts a correct input. Thus, p'_i is typically small and is related to the probability of a defect in the AT. On the other hand, even a defect-free AT may accept an incorrect input due to imperfections in the *coverage* of the testing algorithm. Hence, p''_i lumps together two error sources: imperfect coverage and defective design. We typically have $p''_i>p'_i$ (or even, $p''_i>>p'_i$) for simple low-complexity ATs. For a more comprehensive AT, coverage can be very high or even perfect. In such cases, the values of q'_i and q''_i are comparable, although not necessarily equal.

2.4. Assumption — *Independence of module and AT failures*: Each M and each AT fails independently of other Ms and ATs, unless otherwise noted. Hence, the probability of k modules M_i $(1 \le i \le k)$ coincidentally producing erroneous results is $\prod_{i=1}^{k} p_i$. ∎

2.5. Definition — *Weighted plurality voter*: Given n input data objects $x_1, x_2, \ldots, x_n$, with associated non-negative real votes (weights) $v_1, v_2, \ldots, v_n$, a V node computes the output object y and its vote w such that y is "supported by" a number of input data objects with votes totaling w and no other y' is supported by inputs having more votes. If $w \le \frac{1}{2}\sum_{i=1}^{n} v_i$, then y may be non-unique. In such cases, an erroneous voter output will be pessimistically assumed. The output weight w selects one of the outgoing voter edges along which y or an activation signal must be sent. Various definitions of the term "supported by" lead to different voting schemes such as exact, inexact, and approval voting (e.g., with exact voting, an input object x_i supports y iff $x_i = y$). These variations are not discussed in this paper [7]. ∎

2.6. Assumption — *Perfect voters*: Voters are perfect and act instantaneously. This assumption is reasonable since voters are simpler than modules or ATs and are designed just once for use with many different modules and test types. They can be made highly reliable through careful design and extensive validation/testing. ∎

2.7. Example — Consider a system, called ALT1, consisting of 4 computation modules, an AT, and a voter organized as follows. V receives the outputs of M1, M2, and M3. If the 3 results agree, then the common result is sent to Out. If 2 of 3 agree, the result is subjected to T. If the result passes the test, it is sent to Out. If the result fails or if there is no majority, M4 is invoked and its result used directly with no further test or voting. The reliability formulas are given below without proof:

$$Q_{5\mathrm{VP}} = q^2(1 + 2p + 3p^2 - 6p^3)$$

$$Q_{\mathrm{ALT1}} = q^2[1 + 2p + 3p^2 - 3p^2(p' + p'')]$$

Clearly, the reliability of ALT1 is higher than that of 5VP provided that $p'+p'' < 2p$. In the special case of $p' = p'' = s$, the alternate scheme ALT1 offers reliability improvement over 5VP iff $s < p$. In other words, ALT1 is better than 5VP if the acceptance test T is more reliable than each module M_i. In many practical situations, T can be made simpler, and thus more reliable, than M_i. ∎

Next, we define a modified form of MTV graphs in order to simplify the discussion in the remainder of this paper.

2.8. Definition — *Data-driven MTV graph*: A data-driven MTV (DD-MTV) graph is a modified MTV graph with no "Err" node, and single-output M, T, and V nodes.

M_i: Attaches the vote weight w_i to its output y_i.

T_i: Modifies the weight w of y to $w + a_i(w)$ or $w - r_i(w)$ upon acceptance/rejection of its input.

V_i: Produces an output data object y of weight w from its inputs, as detailed in Definition 2.5.

The elimination of control edges and the resultant uniformity of the graph simplifies the enumeration and analysis of various alternatives, while still retaining the power to accurately model most useful hybrid schemes. The weight augmentation function $a_i(w)$ and the weight reduction function $r_i(w)$, are used to adjust the weight of an accepted and rejected input, respectively. ∎

Fig. 1 depicts 4 DD-MTV graphs, each having 6 M/T nodes (alternatives to 6VP). These examples demonstrate the wide variety of multi-channel architectures that can be easily modeled by DD-MTV graphs.

2.9. Assumption — *Uniformity of modules and ATs*: In the rest of this paper, modules will be assumed to have identical reliability, complexity, and execution-time parameters. Thus, the subscript i will be omitted form parameters such as p_i, q_i and the weight $w = 1$ is attached to all module outputs. Similarly, the subscript i will be removed for ATs. ATs will be assumed to have perfect coverage and lower or equal complexity compared to modules; thus the assumptions $p' \le p$ and $p'' \le p$. ▮

2.10. Assumption — *Weight augmentation/reduction functions for ATs*: Selection of appropriate weight augmentation and reduction functions, $a(w)$ and $r(w)$, can have important effects on the overall reliability of the system modeled by a DD-MTV graph. Here, we assume $a(w) = r(w) = 1$. These simple constant functions can be intuitively justified when ATs have near-perfect coverage and are of comparable complexity to modules. ▮

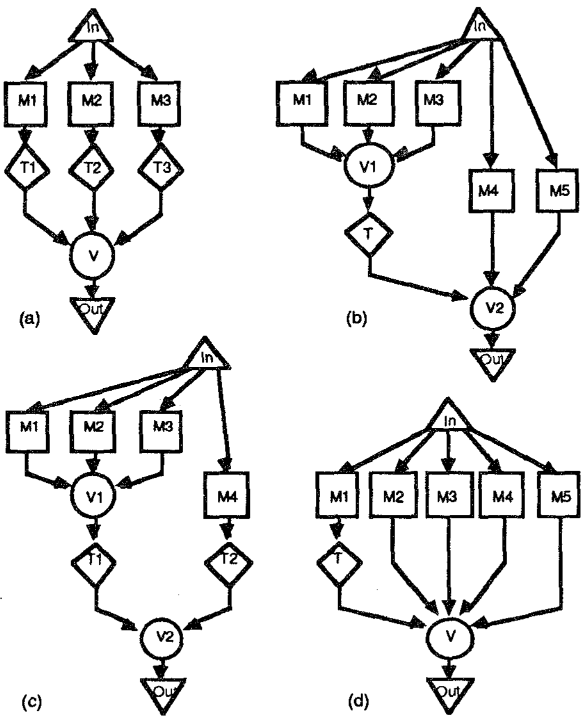

Fig. 1. Examples of DD-MTV graphs with six M/T nodes (alternatives to 6VP).

3. Replacing One Version with an AT

When one module of nVP is replaced by an AT, an asymmetric hybrid architecture is obtained (Fig. 1d for $n = 6$). For $n = 3$, this yields a systems whose reliability is equal to that of the recovery block scheme with 1 alternate. Denoting the system with 2 modules and an AT at the output of M1 as ALT2, we have:

$$Q_{3\mathrm{VP}} = q^3 + 3q^2p \;=\; q(1 + p - 2p^2)$$

$$Q_{\mathrm{ALT2}} = qq' + qp'q + pq''q \;=\; q[1 + p - p(p' + p'')]$$

From the above, we have $Q_{\mathrm{ALT2}} > Q_{3\mathrm{VP}}$ iff $p' + p'' < 2p$. Hence the discussion in Example 2.7 applies here also. Fig. 2 shows the unreliability $P = 1 - Q$ of 3VP and ALT2 schemes when $p' = p''$.

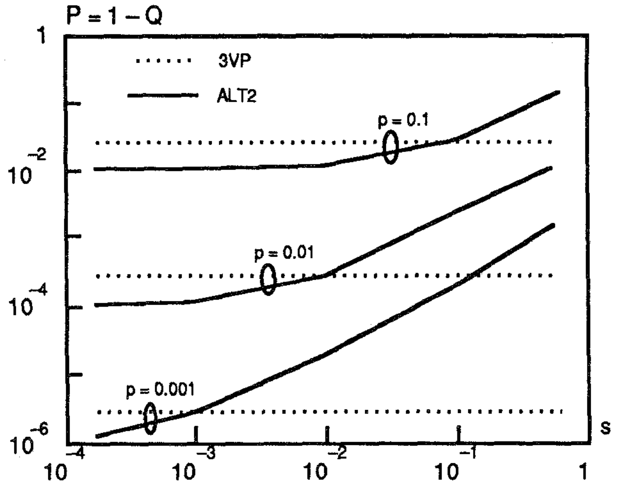

Fig. 2. Unreliability $P = 1-Q$ of 3VP and ALT2, assuming $p' = p'' = s$.

We next generalize the above analysis to the comparison of nVP and the alternative hybrid scheme ALT3 (similar to Fig. 1d, but with $n-1$ modules). But we first need some notation. Let $R_{k,m}$ be the reliability of a homogeneous k-out-of-m system in which each module fails with probability p (the parameter p is not explicitly shown).

$$R_{k,m} = \sum_{j=k}^{m} \binom{m}{j} q^j p^{m-j}$$

$R_{k,m}$ is defined to be 0 for $k > m$ and 1 for $k \le 0$. We now write reliability equations for nVP and ALT3 as follows. To simplify the reliability expressions, let $h = \lfloor n/2 \rfloor$. Each of the following expressions is written by considering the four possible cases with respect to the presence of faults in two modules (nVP) or in one module and its associated AT (ALT3) and for each case figuring out how many of the remaining $n - 2$ modules must be fault-free in order to guarantee a correct result.

$$Q_{n\mathrm{VP}} = R_{h+1,n} = q^2R_{h-1,n-2} + 2pqR_{h,n-2} + p^2R_{h+1,n-2}$$

$$Q_{\mathrm{ALT3}} = qq'R_{h-1,n-2} + qp'R_{h,n-2} + pq''R_{h,n-2} + pp''R_{h+1,n-2}$$

The difference of reliabilities, $\Delta Q = Q_{\mathrm{ALT3}} - Q_{n\mathrm{VP}}$, is:

$$q(p-p')R_{h-1,n-2} + [p(p-p'') - q(p-p')]R_{h,n-2} - p(p-p'')R_{h+1,n-2}$$

$$= q(p-p')[R_{h-1,n-2} - R_{h,n-2}] + p(p-p'')[R_{h,n-2} - R_{h+1,n-2}]$$

Since each of the two terms within square brackets is positive, a sufficient condition for reliability improvement over nVP is immediately obtained as $\max(p', p'') < p$, which always holds by Assumption 2.9. To continue the analysis, we note that:

$$R_{k-1,m} - R_{k,m} = \binom{m}{k-1} q^{k-1} p^{m-k+1}$$

$$= m!\, q^{k-1} p^{m-k+1} / [(k-1)!\,(m-k+1)!]$$

After some manipulation, we can rewrite ΔQ as:

$$(n-2)!/[h!(n-h-1)!]\; q^h p^{n-h-1} [(n-1)p - hp' - (n-h-1)p'']$$

Therefore, the sign of ΔQ depends on the sign of the last expression within square brackets. For n odd, $h = (n-1)/2$ and $\Delta Q > 0$ iff $p' + p'' < 2p$. For n even, we have $h = n/2$ and $\Delta Q > 0$ iff $(n/2-1)(p'+p'') + p' < (n-1)p$. In this case, p' is somewhat more important than p''. As an example, for $n=6$, we must have $3p' + 2p'' < 5p$ if the alternative with one AT (Fig. 1d) is to be more reliable than 6VP.

4. Replacing k Versions with ATs

We now consider the case where k of the n modules are removed ($k \leq n/2$) and replaced by ATs following k of the remaining $n - k$ modules. As in Section 3, let $R_{k,m}$ be the reliability of a homogeneous k-out-of-n system in which each module fails with probability p and use $h = \lfloor n/2 \rfloor$ for notational convenience. We can then write:

$$Q_{n\mathrm{VP}} = R_{h+1,n} = \sum_{i=h+1}^{n} \binom{n}{i} q^i p^{n-i}$$

$$Q_{\mathrm{ALT4}} = \sum_{i=0}^{k} \sum_{j=0}^{k-i} \binom{k}{i} \binom{k-i}{j} (pp'')^i (pq''+qp')^j (qq')^{k-i-j} R_{h+2i+j-2k+1,n-2k}$$

The reliability expression Q_{ALT4} is derived as follows. Assume that of the k branches containing ATs, i have faults in both the module and in the AT, j have a fault in either the module or the AT but not both, and $k - i - j$ are fault-free. The i branches with double faults all potentially produce incorrect results with weight 2. The j branches containing single faults produce results with weight 0, whether the fault is in M or in T. Finally, the $k - i - j$ fault-free branches produce correct results of weight 2. If c of the remaining $n - 2k$ modules produce correct results, the following condition must be met for the output to be guaranteed correct:

$c + 2(k-i-j) > (n-2k-c) + 2i$ or $c \geq h+2i+j-2k+1$

This justifies the term $R_{h+2i+j-2k+1,n-2k}$ in the expression for Q_{ALT4}. The other terms are simply the probabilities of the indicated number of faults raised to appropriate powers. For example, the probability that M is faulty but T catches the error or M is fault-free but T erroneously rejects its output is $pq'' + qp' = p + p' - pp' - pp''$.

As written, the expressions for $Q_{n\mathrm{VP}}$ and Q_{ALT4} are difficult to compare without resorting to numerical calculation. To facilitate comparison, we rewrite $Q_{n\mathrm{VP}}$ in the following way. Divide the set of n modules into k module pairs plus $n - 2k$ individual modules. Each of the k pairs can have 2, 1, or 0 faults. Let i, j, and $k - i - j$ be the number of such pairs, respectively. The k module pairs contribute incorrect results with total weights of up to $2i + j$ and correct results with total weights of at least $2k - 2i - j$. Again, if the remaining $n - 2k$ modules produce c correct results and $n - 2k - c$ incorrect ones, we must have $c + 2k - 2i - j > (n - 2k - c) + 2i + j$, or $c \geq h + 2i + j - 2k + 1$, to guarantee a correct output. The probabilities of having 2, 1, ore 0 faulty modules in a pair of modules are p^2, $2pq$, and q^2, respectively. Thus:

$$Q_{n\mathrm{VP}} = \sum_{i=0}^{k} \sum_{j=0}^{k-i} \binom{k}{i} \binom{k-i}{j} (p^2)^i (2pq)^j (q^2)^{k-i-j} R_{h+2i+j-2k+1,n-2k}$$

Comparing the corresponding ij terms in the expression for Q_{ALT4} to the above expression for $Q_{n\mathrm{VP}}$ provides some insight. For example, for $p' = p'' = s$, corresponding terms become identical and the two schemes are equivalent with respect to reliability. For $p' = p'' = s$, the ij term in Q_{ALT4} divided by the ij term in $Q_{n\mathrm{VP}}$ yields the ratio:

$(s/p)^i \, [(p + s - 2ps)/(2pq)]^j \, [(1 - s)/q]^{k-i-j}$

For particular values of s and p satisfying $s < p$, the first and the second term above are always less than 1 while the third term is always greater than 1. Hence, the above ratio can be less than or greater than 1 depending on the values of i and j and no conclusion can be drawn based on this naive term-by-term comparison.

For $n = 3$, the only acceptable value for k is 1 and Fig. 2 depicts the corresponding changes in the unreliability $P = 1 - Q$. To observe the effect of changing k, $Q_{n\mathrm{VP}}$ and Q_{ALT4} have been evaluated for $n = 5$, with $k = 1$ or 2, and $p = 0.1$, 0.01, or 0.001, assuming $p'=p''=s$. Fig. 3 depicts the resulting unreliabilities as functions of s. As expected, the unreliabilities are identical when $s = p$ (crossover points in Fig. 3). The 5VP scheme is uniformly better for $s > p$. In the case of $s < p$, both alternates are uniformly better than 5VP and the alternate with $k = 2$ is better than the one with $k = 1$. It is worth noting that the improvement in reliability for $s < p$ is smaller than the degradation for $s > p$, particularly for larger k. Therefore, modules must be replaced with ATs only if the condition $s < p$ is reasonably certain.

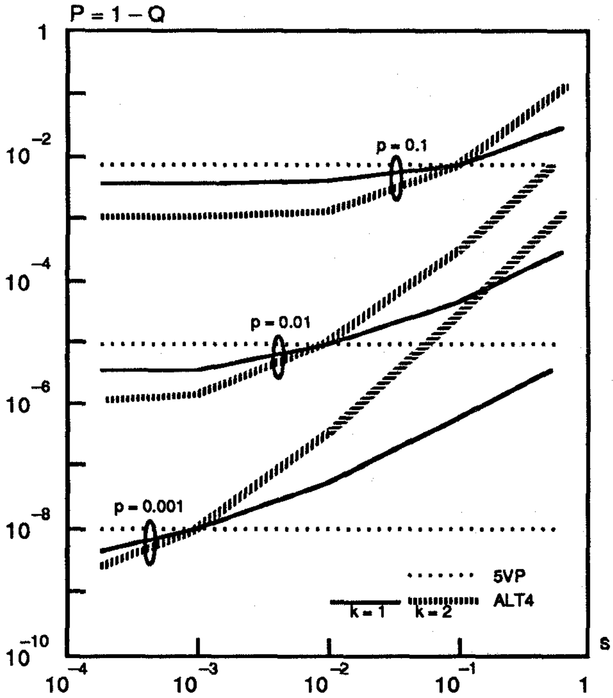

Fig. 3. Unreliability $P = 1-Q$ of 5VP and ALT4, with $k = 1$ or 2, assuming $p' = p'' = s$.

5. More General Schemes

As seen from the examples in Fig. 1, DD-MTV graphs and the systems they model can be composed in many different ways. The systems discussed and analyzed in Sections 3 and 4 all involve a single level of voting. In this section, we discuss a system involving two levels of voting as an example of more general architectures.

Consider the DD-MTV graph ALT5 which is similar to Fig. 1b, except that the 1st level of voting involves k modules and the 2nd level involves n–k–1 other modules. This may be viewed as a generalized recovery block scheme in which the primary computation is a k-way voted block and the alternate consists of an (n–k–1)-way parallel block. In an actual implementation, modules in the alternate block may be executed sequentially until sufficient votes are collected, given the outcome of the primary voting block. The reliability of ALT5 is:

$$Q_{\mathrm{ALT5}} = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} q^i p^{k-i} \left[q'' R_{\lfloor (n-i)/2 \rfloor, n-k-1} + p'' R_{\lfloor (n-i)/2 \rfloor + 1, n-k-1} \right]$$

$$+ \sum_{i=\lfloor k/2 \rfloor + 1}^{k} \binom{k}{i} q^i p^{k-i} \left[q' R_{\lfloor (n-k-i)/2 \rfloor, n-k-1} + p' R_{\lfloor (n-k-i)/2 \rfloor + 1, n-k-1} \right]$$

The above expression for Q_{ALT5} is derived as follows. Let there be i correct results among the k channels of the primary voting block; this event has probability $\binom{k}{i} q^i p^{k-i}$. Now if $i \le \lfloor k/2 \rfloor$, the plurality voting result must be assumed incorrect in the worst case. If T rejects this incorrect result (probability q''), its weight becomes $i-1$ and a correct final output will be produced as long as $\lfloor (n-i)/2 \rfloor$ of the remaining $n-k-1$ modules are fault-free. On the other hand, if T erroneously accepts the incorrect result (probability p''), increasing its weight to $i+1$, then at least $\lfloor (n-i)/2 \rfloor + 1$ of the remaining $n-k-1$ modules must be fault-free for the final result to be guaranteed correct. Recall that $R_{j,m} = 0$ for $j > m$. Similarly, if $i \ge \lfloor k/2 \rfloor + 1$, then the voter output is correct and has weight i. The second half of the expression for Q_{ALT5} is similarly justified.

To compare the above reliability expression to $Q_{n\mathrm{VP}}$, we divide the n modules into 3 groups of k, 1, and $n - k - 1$ modules. Let i be the number of fault-free modules in the first group. Then, $Q_{n\mathrm{VP}}$ can be written as:

$$Q_{n\mathrm{VP}} = \sum_{i=0}^{k} \binom{k}{i} q^i p^{k-i} \left[(q R_{\lfloor (n-k-i)/2 \rfloor, n-k-1} + p R_{\lceil (n-k-i)/2 \rceil, n-k-1} \right]$$

The two terms within square brackets correspond to the case of the single module in the second group being fault-free (probability q) or faulty (probability p), respectively, leading to different requirements for the number of fault-free modules in the third group ($i+1+c > n-k-1-c$ in the first case and $i+c > n-k-1-c$ in the second, where c is the required number of fault-free modules in the third group). Again comparison of the expressions for Q_{ALT5} and $Q_{n\mathrm{VP}}$ leads to no general conclusion. For example, in the case of $p' = p'' = p$, we note that of the corresponding pairs of terms in the expressions for Q_{ALT5} and $Q_{n\mathrm{VP}}$, some are larger in Q_{ALT5} and others are larger in $Q_{n\mathrm{VP}}$.

To get an intuitive feeling for the relative values of Q_{ALT5} and $Q_{n\mathrm{VP}}$ and conditions under which the alternative scheme offers a higher reliability than NVP, consider the special case depicted in Fig. 1b ($n = 6$, $k = 3$). The relevant reliability equations in this case are:

$$R_{6\mathrm{VP}} = q^2(1 + 2p + 3p^2 - 16p^3 + 10p^4)$$

$$R_{\mathrm{ALT5:6,3}} = q^2[1+2p+3p^2-3p^3-p^2(1+2p)p' - 3p^2(1-p)p'']$$

To compare the above reliabilities, let us compute their difference $\Delta Q = Q_{\mathrm{ALT5:6,3}} - Q_{6\mathrm{VP}}$:

$$\Delta Q = q^2 p^2 [p\,(13 - 10p) - (1 + 2p)p' - 3(1-p)p'']$$

Hence for the alternative scheme to be better than 6VP, we must have $(1+2p)p' + 3(1-p)p'' < p(13-10p)$. Observe that p'' is somewhat more important than p' in that it is multiplied by a larger term. In the special case of $p' = p'' = s$, The above condition becomes $s < p\,(13-10p)/(4-p)$ or $s < 3p + p(1-7p)/(4-p)$. Thus, for p reasonably small, reliability improvement is guaranteed as long as $s \le 3p$.

One cannot draw general conclusions from one example, but it is interesting to find the cause of the improved reliability. Both 6VP and the scheme of Fig. 1b produce the correct result when 4 or more M/T nodes are fault-free. To see this for Fig. 1b, consider the following 5 cases which exhaust all possible double failures.

Case 1: 2 failures in $\{M_1,M_2,M_3\}$: Fault-free T rejects the incorrect voter output, reducing its weight from 2 to 1. Correct output is produced since M_4 and M_5 are fault-free.

Case 2: 1 failure in $\{M_1,M_2,M_3\}$ and 1 in T: T rejects the correct voter output, reducing its weight from 2 to 1. The output would be correct even if M_4 or M_5 were faulty.

Case 3: 1 failure in $\{M_1,M_2,M_3\}$ and 1 in $\{M_4,M_5\}$: T accepts the correct voter output, increasing its weight from 2 to 3. The output is independent of M_4 or M_5.

Case 4: 1 failure in T and 1 in $\{M_4,M_5\}$: T rejects the correct voter output, reducing its weight from 3 to 2. The healthy module in $\{M_4,M_5\}$ creates a correct majority.

Case 5: 2 failures in $\{M_4,M_5\}$: T is fault-free and accepts the unanimous voter output, increasing its weight from 3 to 4. M_4 and M_5 cannot affect the correct output.

Cases 2 and 3 above show that some triple failures are also tolerated by ALT5; hence the improved reliability. In certain instances, as in Cases 3 and 5 above, the output of T obviates the need for executing M_4 or M_5. These cases correspond to up to one fault in $\{M_1,M_2,M_3\}$, with T fault-free, and have a probability of $q^2(1 + 2p)(1 - s)$.

One should note that if there were no AT between the two voting levels in Figure 1b, reliability would actually degrade compared to a single-level scheme with the same number of modules. The reason is that correct minority results in the first level would be discarded whereas they may help establish a correct majority if combined with correct outputs from the remaining modules. So the AT is a key element in this multi-level voting configuration.

6. Dealing with Correlated Failures

Analysis of various hybrid redundancy schemes with correlated failures becomes significantly more complex. In this section, we present a simplified analysis based on a highly pessimistic view of correlated failures: that they affect a set of modules and ATs in the worst possible way, causing the modules to produce identical incorrect results and an AT to reject (accept) any correct (incorrect) result. We obtain lower bounds for the reliabilities of pure and hybrid schemes and show the bounds for certain hybrid schemes to be higher. This does not necessarily imply that the hybrid schemes are more reliable ($a>b$, $a'>b'$, and $b>b'$ do not imply $a>a'$). On the other hand, the exact reliability of a complex system is not computable and we usually settle for lower bound guarantees. From this viewpoint, a system for which the lower-bound or guaranteed reliability is higher must be considered better.

In what follows, we compare nVP and ALT3 schemes with regard to correlated failures and show ALT3 to be superior. Since nVP and ALT3 differ only in the use and placement of M_1, M_n, and T, our model postulates the occurrence of correlated failures in c modules among $\{M_2, \ldots, M_{n-1}\}$ and includes probability parameters relating to how M_1, M_n, and/or T may be affected. The parameters β, β', σ, μ, τ, ν, ν', defined below, should be interpreted as "probability of *event*, given that c modules in $\{M_2, \ldots, M_{n-1}\}$ contain correlated/common failures". Nodes unaffected by correlated failures can still suffer from random failures, with corresponding parameters as defined in Section 2. The events associated with the conditional probabilities for nVP and ALT3 are:

NVP	β	Both M_1 and M_n are affected
	σ	M_1 (M_n) is affected but M_n (M_1) is not
	ν	Neither M_1 nor M_n is affected
ALT3	β'	Both M_1 and T are affected
	μ	M_1 is affected but T is not
	τ	T is affected but M_1 is not
	ν'	Neither M_1 nor T is affected

Clearly, we have $\beta+2\sigma+\nu = \beta'+\mu+\tau+\nu' = 1$. Also, given that two modules are more similar than a module and an AT, the following are reasonable assumptions:

$$\tau \le \sigma \le \mu \quad \text{and} \quad \beta' \le \beta \le \sigma \le \nu \le \nu'$$

These inequalities are essentially the crux of our comparison, in much the same way that the condition $p' + p'' < 2p$ was essential in proving improvements with independent failures. One last item of notation: Since the k-out-of-(n–c–2) reliability, $R_{k,n-c-2}$, is used repeatedly in the following analysis, we denote it by R_k for brevity. Recall that h was defined as $\lfloor n/2 \rfloor$.

We next derive upper bounds on the reliability reduction due to correlated failures in nVP and ALT3. The "≈" sign denotes proportionality rather than equality.

$$\begin{aligned} \Delta Q_{n\mathrm{VP}} &\approx \beta(1 - R_{h+1}) + 2\sigma[q(1 - R_h) + p(1 - R_{h+1})] \\ &\quad + \nu[q^2(1 - R_{h-1}) + 2pq(1 - R_h) + p^2(1 - R_{h+1})] \\ &= 1 - [(\beta+2\sigma p+\nu p^2)R_{h+1} + 2q(\sigma+\nu p)R_h + \nu q^2 R_{h-1}] \end{aligned}$$

$$\begin{aligned} \Delta Q_{\mathrm{ALT3}} &\approx \beta'(1 - R_{h+1}) + \mu[q''(1-R_h) + p''(1-R_{h+1})] \\ &\quad + \tau[q(1-R_h) + p(1-R_{h+1})] + \nu'[qq'(1 - R_{h-1}) \\ &\quad + p'q(1 - R_h) + pq''(1 - R_h) + pp''(1 - R_{h+1})] \\ &= 1 - [(\beta' + \mu p'' + \tau p + \nu' pp'')R_{h+1} \\ &\quad + (\mu q'' + \tau q + \nu' p'q + \nu' pq'')R_h + \nu' qq' R_{h-1}] \end{aligned}$$

$$\begin{aligned} \Delta Q_{n\mathrm{VP}}-\Delta Q_{\mathrm{ALT3}} &\approx (\beta'-\beta+\mu p''+\tau p-2\sigma p+\nu' pp''-\nu p^2)R_{h+1} \\ &\quad +(\mu q''+\tau q-2\sigma q+\nu' p'q+\nu' pq''R_h-2\nu pq)R_h+q(\nu' q'-\nu q)R_{h-1} \end{aligned}$$

To simplify the above expression for $\Delta Q_{n\mathrm{VP}} - \Delta Q_{\mathrm{ALT3}}$, note the following equalities:

$$R_h = R_{h+1} + \binom{n-c-2}{h} q^h p^{n-c-2-h}$$

$$R_{h-1} = R_{h+1} + \binom{n-c-2}{h} q^h p^{n-c-2-h} + \binom{n-c-2}{h-1} q^{h-1} p^{n-c-1-h}$$

Substituting the above in the expression for $\Delta Q_{n\mathrm{VP}} - \Delta Q_{\mathrm{ALT3}}$, the coefficient for R_{h+1} becomes 0. Dividing both sides by $(n-c-2)!q^h p^{n-c-2-h}/[h!(n-c-1-h)!]$, yields:

$$\begin{aligned} \Delta Q_{n\mathrm{VP}} - \Delta Q_{\mathrm{ALT3}} &\approx (n-c-1-h)[q(\mu q''/q+\tau-2\sigma+\nu'-\nu) \\ &\quad + p(\nu' q''-\nu q)] + hp(\nu' q'-\nu q) \end{aligned}$$

Since by our assumptions both $\nu' q'' - \nu q$ and $\nu' q' - \nu q$ are non-negative, a sufficient condition for $\Delta Q_{n\mathrm{VP}} - \Delta Q_{\mathrm{ALT3}}$ to be non-negative is to have $\mu q''/q+\tau-2\sigma+\nu'-\nu \ge 0$.

$$\begin{aligned} \mu q''/q+\tau-2\sigma+\nu'-\nu &= \mu(q''-q)/q + (\mu+\tau+\nu') - (2\sigma+\nu) \\ &= \mu(q''-q)/q + (1-\beta') - (1-\beta) = \mu(q''-q)/q+\beta-\beta' \end{aligned}$$

This last expression is non-negative by our assumptions; hence $\Delta Q_{n\mathrm{VP}} \ge \Delta Q_{\mathrm{ALT3}}$ and the conclusion that correlated failures affect nVP less favorably than ALT3.

7. Conclusions

We have introduced a general framework for the creation, representation and analysis of combined NVP-AT schemes which covers previously proposed NVP-AT combinations, such as CRB, RNVB, and NSCP, as special cases. MTV graphs, and their simplified version called DD-MTV graphs in this paper, facilitate the discussion and analyses of various fault-tolerant system architectures for critical applications. Preliminary results on new hybrid NVP-AT schemes demonstrate the potential for significant reliability improvement with judicious placement of the various component types.

Continued research in this area will enhance the utility of the proposed general framework for the study of hybrid NVP-AT schemes, leading to specific design techniques, performance comparisons, and tradeoff guidelines. Results of such extended studies will contribute both to a basic understanding of voting and acceptance testing as *dependability enhancement* mechanisms and to their application in realizing ultrareliable computations. Specific problems for future investigation include:

- Optimal weight augmentation and reduction policies; the $a(w)$ and $r(w)$ functions of Definition 2.8.
- Effects of unequal module complexities and reliabilities as well as imperfect voters.
- Optimal number of modules to be replaced by ATs (the parameter k of Section 4).
- Optimal partitioning of n modules for 2-level voting.
- More general multi-level voting schemes and their attendant design tradeoffs.
- Effect of combined correctness/timeliness requirements in highly reliable real-time systems.

The ultimate goal is to solve the following problem: Given values for p, p', p'', as well as other system cost and reliability parameters, what is the most cost-effective configuration of computation modules, ATs, and voters? As this problem is quite difficult, any approach to its solution will necessarily use a number of intermediate problems. For example: What is the best arrangement of n M/T modules to maximize the overall reliability?

References

[1] A. Avizienis,"The *N*-Version Approach to Fault-Tolerant Software", *IEEE Trans. Software Engineering*, Vol. 11, No. 12, pp. 1491-1501, Dec. 1985.

[2] D.E. Eckhardt et al, "An Experimental Evaluation of Software Redundancy as a Strategy for Improving Reliability", *IEEE Trans. Software Engineering*, Vol. 17, No. 7, pp. 692-702, July 1991.

[3] R.E. Gantenbein, S.Y. Shin, and J.R. Cowles, "Evaluation of Combined Approaches to Distributed Software-Based Fault Tolerance", *Proc. of the Pacific Rim Int'l Symp. Fault-Tolerant Systems*, Kawasaki, Japan, Sep. 1991, pp. 70-75.

[4] J.-C. Laprie, J. Arlat, C. Beounes, and K. Kanoun, "Definition and Analysis of Hardware- and Software-Fault-Tolerant Architectures", *Computer*, Vol. 23, No. 7, pp. 39-51, July 1990.

[5] N.G. Leveson, S.S. Cha, J.C. Knight, and T.J. Shimeall, "The Use of Self Checks and Voting in Software Error Detection: An Empirical Study", *IEEE Trans. Software Engineering*, Vol. 16, No. 4, pp. 432-443, Apr. 1990.

[6] B. Parhami, "A Data-Driven Dependability Assurance Scheme with Applications to Data and Design Diversity", in *Dependable Computing for Critical Applications*, Springer-Verlag, 1991, pp. 257-282.

[7] B. Parhami, "Voting Algorithms", *IEEE Trans. Reliability*, Vol. 43, No. 4, pp. 617-629, Dec. 1994.

[8] B. Randell, "System Structure for Software Fault Tolerance", *IEEE Trans. Software Engineering*, Vol. 1, No. 2, pp. 220-232, June 1975.

[9] K. Scott, J.W. Gault, and D.F. McAllister, "The Consensus Recovery Block" *Proc. of the Total System Reliability Symp.*, 1983, pp. 74-85.