Redundancy Optimization of a System under the Fault-tolerant System

Chetan Kumar Sharma
Assistant Professor, Department of Mathematics
Skyline Institute of Engineering & Technology, Greater Noida, U.P., India
Email: cks26april@gmail.com

Fault-tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. Fault-tolerance is particular sought after in high-availability or life-critical system. A fault tolerant design enables a system to continue its intended operation possibly at a reduced level, rather than falling completely, when some part of the system fails. In this paper we discuss the redundancy optimal system size with the reliability evaluation of a system of components.

Keywords: Fault-tolerant, Reliability Evaluation, Parallel System, Probability, Minimize Average Cost, Hardware & Software Redundant.

1. INTRODUCTION

In many critical applications of digital systems, fault tolerance has been an essential architectural attribute for achieving high reliability. Several techniques can achieve fault tolerance using redundant hardware [1] or software [2]. Typical forms of redundant hardware structures for fault-tolerant systems are of two types: fault masking and fault standby. Redundant software structures for fault-tolerant systems based on the acceptance tests have been proposed by Horning *et.al.* [3]

A monitor and a switch are associated with each redundant unit. The switches and monitors can fail. The monitors have two failure modes: failure to accept a correct result, and failure to reject an incorrect result. We consider a digital circuit module designed to process the incoming messages in a communication system. This module consists of two units: a converter to process the messages, for their accuracy. To guarantee a high reliability of operation at the receiver end, n converters are arranged in "parallel". All, except converter n, have a monitor to determine if the output of the converter is correct. If the output of the converter is not correct, the output is cancelled and a switch is changed so that the original input message is send to next converter. Systems of these kinds have useful application in communication and network control systems and in the analysis of fault-tolerant software systems [4].

We assume that a switch is never connected to the next converter without a signal from the monitor, and the probability that it is connected when a signal arrives is p_s. We next present a general expression for the reliability of the system consisting of n non-identical converters arranged in "parallel" [5]. An optimization is formulated and solved for the minimum average system cost [6]. Let us define the following notation, events, and assumptions.

The notation is as follows [7]:

p_i^c : Pr{converter $i$works}

p_i^s : Pr{switch i is connected to converter $(i+1)$ when a signal arrives}

p_i^{m1} : Pr{monitor i works when converter $i$works}

= Pr{not sending a signal to the switch when converter$i$ works}

p_i^{m2} : Pr{monitor i works when converter $i$has failed}

= Pr{sending a signal to the switch when converter$i$ has failed}

R_{n-k}^k: Reliability of the remaining system of size $n-k$ given that the first k switches work

R_n : Reliability of the system consisting of n converters.

The events [8] are:

C_i^w, C_i^f : Converter i works, fails

M_i^w, M_i^f : Monitor i works, fails

S_i^w, S_i^f : Switch i works, fails

W : System works

The assumptions are:

1. The system, the switches, and the converters are two-state either good or failed [9].
2. The module (converter, monitor, or switch) states are mutually statistical independent.
3. The monitors have three states: good, failed in mode 1, failed in mode 2.
4. The modules are not identical.

2. RELIABILITY EVALUATION

The reliability of the system is defined as the probability of obtaining the correctly processed message at the output [10,11]. To derive a general expression for the reliability of the system, we use an adapted from the total probability as translated into the language of reliability. Let A denoted the event that a system performs as desired. Let X_i and X_j be the event that a component X $(e.g.$ converter, monitor, or switch) is good or failed respectively. Then

Pr{system works} = Pr{system works when unit X is good}×Pr{unit X is good} + Pr{system works when unit X fails }× Pr{ unit X failed}

The above equation provides a convenient way of calculating the reliability of complex systems [12].

When $R_1 = p_i^c$, and for n≥2 [13], the reliability of the system can be calculated as

follows:

R_n= Pr{$W|C_1^w$and M_1^w} Pr{$C_1^w$and M_1^w}+ Pr {$W|C_1^w$and M_1^f} Pr {$C_1^w$and M_1^f}+ Pr {$W|C_1^f$ and M_1^w} Pr {C_1^f and M_1^w}+ Pr {$W|C_1^f$ and M_1^f} Pr {C_1^f and M_1^f}

In order for the system to operate when the first converter works and the first monitor fails, the first switch must work and the remaining system of size $(n-1)$ must work:

$$\text{Pr } \{W|C_1^w \text{and} M_1^f\} = p_1^s R_{n-1}^1$$

Similarly:

$$\text{Pr } \{C_1^f \text{and} M_1^w\} = p_1^s R_{n-1}^1$$

Then

$$R_n p_1^c p_1^{m1} + [p_1^c(1-p_1^{m1}) + (1-p_1^c)p_1^{m2}]p_1^s R_{n-1}^1$$

The reliability of the system consisting of n non-identical converters can be easily obtained:

$$R_n = \sum_{i=1}^{n-1} p_i^c p_i^{m1} \pi_{i-1} + \pi_{n-1} p_n^c \text{For n> } 1$$

and $R_1 = p_1^c$

where $\pi_k^j = \prod_{i=j}^{k} A_i for k \geq 1$

$\pi_k = \pi_k^1$ For all k, and $\pi_0 = 1$

And $A_i = [p_i^c(1-p_i^{m1}) + (1-p_i^c)p_i^{m2}]$ for all $i = 1,2,\dots,n$

$p_i^c = p^c,\ p_i^{m1} = p^{m1},\ p_i^{m2} = p^{m2},\ p_i^s = p^s$ for all i,

Then we obtain a closed form expression for the reliability of system as follows [14]:

$$R_n = \frac{p^c p^{m1}}{1-A}(1-A^{n-1}) + p^c A^{n-1} \quad (1)$$

Where $A = [p^c(1-p^{m1}) + (1-p^c)p^{m2}]p^s$

3. REDUNDANCY OPTIMIZATION

Assume that the system failure costs d units of revenue, and that each converter, monitor and switch module costs a, b and c units respectively. Let T_n be system cost for a system of size n. the average system cost of size $n, E[T_n]$, [15] is the cost incurred when the system has failed, plus the cost of all n converters, $(n-1)$ monitors, and $(n-1)$ switches.

Therefore:

$$E[T_n] = an + (b + c)(n - 1) + d(1 - R_n)$$

Where R_n is given in equation (1). The minimum value of $E[T_n]$ is attained at

$$n^* = \begin{cases} 1 & if A \leq 1 - p^{m1} \\ \lfloor n_0 \rfloor & if\ otherwise \end{cases}$$

Where

$$n_0 = \frac{\text{In (a + b + c)} - \text{In}[dp^c(A + p^{m1} - 1)]}{\text{In A}} + 1$$

4. NUMERICAL DISCUSSION

A converter works and p^c= 0.75 gives and a monitor works when converter works with p^{m1}=0.89 but monitor works when converter fails p^{m2}= 0.95 and the system costs$d$= 1000units of the revenue and that each converter, monitor and switch model cost $a = 2.5, b = 2.0$ and $c = 1.5$ units respectively. Now we discuss the corresponding average cost (minimized).

We have $A = [0.75(1 - 0.89) + (1 - 0.75)95]0.89$

$$R_n = \frac{0.75 \times 0.89}{1 - A}(1 - A^{n-1}) + 0.75A^{n-1}$$

$$E[T_n] = 2.5n + (2.0 + 1.5)(n - 1) + 1000(1 - R_n)$$

Where $n_0 = \frac{\log 6 - \log\,[1000 \times 0.75(A + 0.89 - 1)]}{\log A}$ +1

Then the optimal system size is n^* and the corresponding average cost will be minimized.

5. CONCLUSION

Fault tolerant systems must be used whenever a failure can result in loss of life or loss of a high value asset. In general faults tolerance considered as a study of faults/failures behavior is the reasonable starting point of stopping their effects as any system defects, the technique and tools are developed towards how to probe this behavior and further how to stop the propagation. To optimize the system of minimize average cost, worked will be focused of a system when it fails.

REFERENCES

[1] F.P. Mathur and P.T. de Sousa; “Reliability modeling and analysis of general modular redundant systems”, IEEE Trans Reliability, Vol. 24(5), pp. 296-299, 1975.

[2] J.J. Horning, H.C. Lauer, P.M. Melliar-Smith and B. Randell; “A program structure for error detection and recovery”, Lecture Notes in Computer Science, Springer, Vol. 16(2), pp. 177-187, 1974. DOI: 10.1007/BFb0029359

[3] B. Balasubramanian and V. Garg; “A fusion-based approach for handling multiple faults in data structures”, Technical Report ECE-PDS-2009-001, Parallel and distributed systems Laboratory, ECE Dept. University of Texas at Austin, 2009

[4] Cinzia Bernardeschi and Andrea Domenici; “Application of Model checking to fault tolerance analysis”, Software Engineering to Formal Methods and Tools and Back, Vol. 4(2), pp. 531-547, 2019. https://doi.org/10.1007/978-3-030-30985-5_31

[5] Won Young Yun, Sang Hee Lee and Han Chung; “Standby Redundancy Optimization of Multi-Level Systems,” Journal of the Korean Institute of Industrial Engineers, Vol. 45(6), pp 484-490, 2019.

[6] L. Lamport; “Using Time instead of timeout for fault-tolerant distributed systems”, ACM Trans. Program.Lang. system, Vol. 6(2), pp. 254-280, 1984.

[7] F. Benjamin, Jones and Lee Pike; “Modular model-checking of a Byzantine fault-tolerant protocol”, Conf. on NASA formal methods symposium, pp. 163-177, April 2017.

[8] P. Bokor, J. Kinder, M. Serafini and N. Suri; “Efficient model checking of fault tolerant distributed protocols”, IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), Hong Kong, China, pp. 73-84, 2011. doi: 10.1109/DSN.2011.5958208

[9] Jun-Ming Xu, Qiang Zhu and Min Xu; “Fault-tolerant analysis of a class of networks”, Information Processing Letters, Vol. 103(6), pp. 222-226, 2007.

[10] Ayan Palchaudhuri and Anindya Sundar Dhar; “Fault localization and testability approaches for FPGA fabric aware canonic signed digit recording implementations”, Journal of electronic Testing, Vol. 35(1), pp. 779-796, 2019.

[11] A.K.S. Pundir and O.P. Sharma; “Fault tolerant reconfigurable hardware design using BIST on SRAM: A review”, International Conference on Intelligent Computing and Control (I2C2), Coimbatore, India, pp. 1-16, 2017. doi: 10.1109/I2C2.2017.8321907.

[12] K.K. Aggarwal; “Redundancy Optimization in General Systems”, IEEE Transaction on Reliability, Vol. R-25(5), pp. 330-332, 1976.

[13] M. Staroswiecki, G. Hoblos and A. Aitouche; “Fault tolerance analysis of sensor systems”, Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304), Phoenix, AZ, USA, Vol. 4(5), pp 3581-3586, 1999.

[14] Ting Wang, Tieming Chen, Yang Liu and Ye Wang; “Anti chain based algorithms for timed /probabilistic refinement checking”, Sci. China. Inf. Sci., Vol. 61(5), pp. 213-222, 2018. https://doi.org/10.1007/s11432-017-9133-4

[15] F.B. Schneider; “Implementing fault-tolerant services using the state machine approach: A tutorial”, ACM computing Surveys, Vol. 22(4), pp. 299-319, 1990.