Journal of Computer Science 2 (6): 473-479, 2006
ISSN 1549-3636

Reliability Evaluation of Distributed Computer Systems Subject to Imperfect Coverage and Dependent Common-Cause Failures

Liudong Xing and Akhilesh Shrestha
Department of Electrical and Computer Engineering, University of Massachusetts Dartmouth

Abstract: Imperfect coverage (IPC) occurs when a malicious component failure causes extensive damage due to inadequate fault detection, fault location or fault recovery. Common-cause failures (CCF) are multiple dependent component failures within a system due to a shared root cause. Both imperfect coverage and common-cause failures can exist in distributed computer systems and can contribute significantly to the overall system unreliability. Moreover they can complicate the reliability analysis. In this study, we propose an efficient approach to the reliability analysis of distributed computer systems (DCS) with both IPC and CCF. The proposed methodology is to decouple the effects of IPC and CCF from the combinatorics of the solution. The resulting approach is applicable to the computationally efficient binary decision diagrams (BDD) based method for the reliability analysis of DCS. We provide a concrete analysis of an example DCS to illustrate the application and advantages of our approach. Due to the consideration of IPC and CCF, our approach can evaluate a wider class of DCS as compared with existing approaches. Due to the nature of the BDD and the separation of IPC and CCF from the solution combinatorics, our approach has high computational efficiency and is easy to implement, which means that it can be easily applied to the accurate reliability analysis of large-scale DCS subject to IPC and CCF. The DCS without IPC or CCF appear to be special cases of our approach.

Key words: Distributed program reliability (DPR), reduced ordered binary decision diagrams (ROBDD), separable approach

INTRODUCTION

A distributed computer system (DCS) is a collection of interconnected independent computers (hosts) that appears to its users as a single coherent system[1]. DCS provide an efficient way to achieve fault-tolerance and share system resources such as processing elements, memory modules, data files, and so on. A successful execution of a distributed program usually requires one or more of the resources that reside on multiple hosts at different geographic sites of the DCS.

It is possible that some faults of hosts or communication links may not be adequately detected and located so that the distributed program cannot be executed successfully despite the presence of adequate redundancies (other operational hosts and links). This phenomenon is known as imperfect coverage (IPC)[2]. The IPC introduces additional failure modes that must be considered for accurate reliability analysis of DCS. In other words, the analysis must allow multiple failure modes including operational (not failed), failed covered, and failed uncovered, rather than the traditional binary designation of operational and failed. This consideration poses unique challenges to existing analysis methods. Because failure to consider IPC in the reliability analysis leads to overestimated system reliability, considerable research have been performed in studying IPC for the reliability analysis of fault-tolerant systems[2-7], but only few of them[5,7] are applicable to DCS and their complexity can increase rapidly as the size of DCS, i.e., the number of hosts and links in a DCS increases.

The challenges increase when common-cause failures are incorporated in the model. Common-cause failures (CCF) are multiple dependent component failures within a system that are a direct result of a common cause (CC) or a shared root cause[8], such as extreme environmental conditions, operation and maintenance errors. Examples abound in the real world. Sabotage, lightning strike, and power outage can cause the simultaneous failure of numerous components in a DCS. It has been shown by many reliability studies that CCF increase a system's joint failure probabilities and thus contributes significantly to the overall unreliability of systems subject to CCF[9]. Therefore, failure to consider CCF in the reliability analysis of such systems leads to underestimated system unreliability measures. Considerable research efforts have been expended in the study of common cause failures for reliability modeling and analysis of computer-based systems. However, the existing CCF models are mainly applicable to non-DCS systems. And they have various limitations, such as being concerned

Corresponding Author: Liudong Xing, University of Massachusetts Dartmouth, USA

with a specific system structure[10-13]; applicable only to systems with exponential time-to-failure distributions[14-16]; limiting analysis to components being affected by at most one common cause, i.e., components belonging to at most a single common-cause group (CCG)[9,17]; having a single common cause (CC) that affects all components of a system[12,18]; or defining CC as being statistically-independent or mutually exclusive. In this study, we seek to address some of these limitations in developing a model for the reliability analysis of DCS subject to CCF by allowing for multiple CC that can affect different subsets of system components, and which can occur statistically-dependently.

As discussed above, a great deal of work has been done to separately address IPC or CCF in the system reliability analysis. To the best of our knowledge, however, only little work[12,18] has considered both IPC and CCF in solving reliability problems. Moreover, the existing methods did not consider IPC and CCF in a DCS and they share a restrictive assumption that a single elementary CC leads to simultaneous failures of all components of a system. In this study we relax the above restriction by utilizing our generalized CCF model for DCS.

And we propose a separable and efficient reduced ordered binary decision diagram (ROBDD) based approach to the reliability analysis of DCS with both IPC and dependent CCF in an elegant manner.

In this study we use the following acronyms, and we assume the singular and plural of an acronym are always spelled the same:

BDD	Binary Decision Diagram
CC	Common Cause
CCE	Common-Cause Event
CCF	Common-Cause Failure
CCG	Common-Cause Group
DCS	Distributed Computer System
DPR	Distributed Program Reliability
DPUR	Distributed Program UnReliability
DSR	Distributed System Reliability
FST	File Spanning Tree
IPC	Imperfect Coverage
IPCM	Imperfect Coverage Model
MFST	Minimal File Spanning Tree
ROBDD	Reduced Ordered BDD
s-	Implies: statistical(ly)

PROBLEM STATEMENT

This study considers the problem of assessing distributed program reliability for distributed computer systems. Distributed program reliability (DPR) is defined as the probability that at least one minimal file spanning tree (MFST) of a distributed program is operational within the time interval (0, t)[7]. A file-spanning tree (FST) is defined as a spanning tree that connects the root node, i.e., the host running the program under consideration to other nodes such that its vertices hold all the required resources for successful execution of the program. An FST is an MFST if there exists no other FST that is a subset of it. An MFST is said to be operational when all its components are operational[7,19]. The approach developed in this study is also applicable to evaluate distributed system reliability (DSR), which is defined as the probability that at least one MFST for all programs is operational[7].

Assumptions

* The DCS is modeled by a probabilistic undirected graph *G(V,E)*, in which vertices represent the hosts and edges represent the communication links[20]. By probabilistic we mean that failure probabilities are assigned to each node and link in the graph.
* Links or nodes in DCS fail *s*-independently with known probabilities.
* The failure probability for each link or node is given as a fixed probability for a given mission time or in terms of a lifetime distribution.
* The imperfect coverage behavior is described using Dugan *et al*'s imperfect coverage model (IPCM, Fig. 1)[2]. The entry point to the model signifies the occurrence of a fault, and three exits represent three possible and mutually exclusive outcomes. If the offending fault is transient and can be handled without discarding any component, the transient restoration exit (labeled R) is taken. The permanent coverage exit (labeled C) denotes the determination of the permanent nature of the fault and the successful isolation and removal of the faulty component. If the C exit is reached, then a covered component failure is said to occur. An uncovered component failure occurs when a single fault (by itself) causes the system to crash. The single-point failure exit (labeled S) is reached in this case. Within the context of reliability analysis it is required to refer to the exit probabilities only. We assume that the three exit probabilities of the IPCM: transient restoration (*r*), permanent coverage (*c*) and single point of failure (*s*) for each component are given as fixed probabilities.

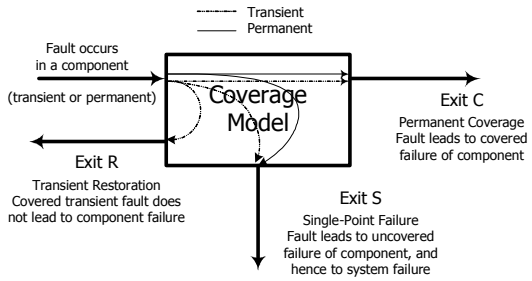

Fig. 1: General structure of the IPCM

CCF model for DCS

* A DCS can be subject to CCF from different common-causes (CC). In general, we express the CC occurring in a DCS as CC_1, CC_2, ……, CC_m, where m is the total number of CC related to the DCS.
* Different CC can occur s-independently, or s-dependently.
* A single component may be affected by multiple CC, i.e., one component can belong to more than one common-cause group (CCG). All components that are caused to fail due to the same elementary CC_i constitute a common cause group CCG_i.

Note that our CCF model is more general and thus more practical than the existing CCF models, which usually require some restrictive assumptions.

Problem inputs: The following lists all the required input parameters for solving the problem:

* DCS configuration in the probabilistic graph
* Mission time t
* Failure parameters of each link and each node
* Fault coverage factors (r_i, c_i, s_i) of each link and each node
* Statistical relationship between elementary CC: s-independent, or s-dependent
* Probabilities of elementary CC occurring or conditional probabilities of CC occurring conditioned on the occurrence of another CC when they are s-dependent.

The accurate reliability analysis of a fault-tolerant DCS heavily depends on the realistic estimate of its input parameters. Fault injection[21,22] is a commonly used technique for estimating the component failure parameters and fault coverage factors. The occurrence probabilities of CC and their statistical relationship can usually be available from sufficient weather data or equipment data[23]. In this study, we consider them as given input parameters of the problem.

AN ILLUSTRATIVE EXAMPLE

We use a simple example (adapted from[7]) to illustrate the proposed methodology for DCS reliability analysis. Figure 2 shows the probabilistic graph of the example DCS.

The links fail s-independently with constant failure rate λ_1=2e-7/hour. The links are subject to uncovered failures with coverage factors r=0, c=0.95, and s=0.05. The nodes fail s-independently with constant failure rate λ_2=1e-7/hour. The nodes are subject to uncovered failures with coverage factors r=0, c=0.99, and s=0.01. Note that for simplicity we assume all the link (node) failures are exponentially distributed with the same failure rate $\lambda_1(\lambda_2)$ and all the links (nodes) have the same coverage factors; our methodology is applicable to arbitrary link (node) failure distributions and coverage factors. In addition, the DCS is subject to CCF from two independent common-causes, earthquakes (denoted by CC_1) and power failures (denoted by CC_2). An earthquake of sufficient intensity would cause links e_2 and e_5 and node n_4 to fail ($CCG_1 = \{e_2, e_5, n_4\}$); a power failure would cause nodes n_1 and n_2 to fail (i.e., $CCG_2 = \{n_1, n_2\}$). We assume that the following information can be extracted from the available weather and power data: the probability of an earthquake is P_{CC1} = 0.001, the probability of a power failure is P_{CC2} = 0.003. The problem is to find DPR for program P_1 in the example DCS for mission time of t = 1000 hours. In the following part, the example will be analyzed to illustrate our method step by step.

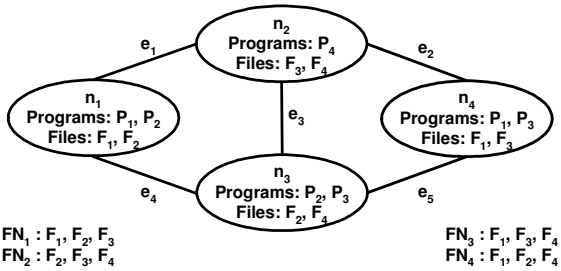

Fig. 2: Probabilistic graph model of the example DCS (adapted from[7]; FN_i denotes the set of files required by a program P_i; system resources are abstracted into files)

SEPARABLE AND EFFICIENT DPR ANALYSIS

We present our separable ROBDD-based approach for analyzing DPR of DCS with both IPC and CCF. The methodology is to separate both IPC and CCF in two phases from the combinatorics of the solution based on the "Total Probability Theorem". The resulted reduced DCS reliability problems are freed from the concern about both CCF and IPC, and can be solved using computationally efficient ROBDD methods. Finally, the results of all reduced sub-problems are integrated to obtain the entire DCS DPR measure.

Separating IPC: Consider two mutually exclusive and complete events E_1 (1 or more components including links and nodes in the DCS fail uncovered) and E_2 (no component experiences an uncovered failure). According to the "Total Probability Theorem", for event E, the failure of a given distributed program whose occurrence probability is distributed program unreliability ($DPUR$), we have:

$$DPUR = \Pr(E) = \Pr(E \mid E_1) \bullet \Pr(E_1) + \Pr(E \mid E_2) \bullet \Pr(E_2)$$
$$= 1 - \Pr(E_2) + \Pr(E \mid E_2) \bullet \Pr(E_2) \quad (1)$$

According to Dugan *et al*'s IPCM[2], we have:

$u[i]$ = Pr (SF_i: component i fails uncovered) = $s_i \bullet q_i(t)$,

$c[i]$ = Pr (CF_i: component i fails covered) = $c_i \bullet q_i(t)$,
$n[i]$ = Pr (NF_i: component i does not fail)
$= 1 - q_i(t) + r_i \bullet q_i(t)$ (2)

$q_i(t)$ is the failure probability of the link/node i within time interval $(0,t)$, which can be obtained directly or calculated from the input failure parameters; r_i, c_i, s_i are fault coverage factors given as input parameters. Based on Eq. (2), we can calculate $\Pr(E_2)$ in Eq. (1) as

$$\begin{aligned}\Pr(E_2) &= \prod_{\forall i}(1-u[i]) \cdot \\ &= \prod_{\forall i}(1-u[i]) \\ &= \prod_{\forall i}(1-s_i \bullet q_i(t)) = \prod_{\forall i}(1-(1-r_i-c_i)\bullet q_i(t)) \\ &= (1-0.05\bullet(1-e^{-1000\bullet 2e-7}))^5 \bullet (1-0.01\bullet(1-e^{-1000\bullet 1e-7}))^4 \\ &= 0.999946\end{aligned}$$

$\Pr(E|E_2)$ in Eq. (1) is the unreliability of corresponding perfect coverage DCS that ignores IPC. It should be evaluated given that no link or node experiences an uncovered failure. Therefore, before calculating $\Pr(E|E_2)$ we modify each node/link's failure function $q_i(t)$ to a conditional probability $\tilde{q}_i(t)$ conditioned on no uncovered failure occurring: $\tilde{q}_i(t) = \Pr(CF_i \mid NF_i \text{ or } CF_i) = \frac{c[i]}{n[i]+c[i]} = \frac{c[i]}{1-u[i]}$, which is valued as 9.8995e-5 for the nodes and 1.8998e-4 for the links of the example DCS. Using these modified component failure probabilities, we can calculate $\Pr(E|E_2)$ by any approach that ignores IPC but considers CCF.

Separating CCF: Based on our CCF model there exist m elementary CC in a DCS. The m CC partition the sample space into the following 2^m disjoint subsets, each called a common-cause event (CCE).

$$\begin{aligned}CCE_1 &= \overline{CC_1} \cap \overline{CC_2} \cap \ldots \cap \overline{CC_m}, \\ CCE_2 &= CC_1 \cap \overline{CC_2} \cap \ldots \cap \overline{CC_m}, \\ &\ldots\ldots, \\ CCE_{2^m} &= CC_1 \cap CC_2 \cap \ldots \cap CC_m.\end{aligned}$$

We build a space called "CCE space" over this set of collectively exhaustive and mutually exclusive common-cause events that can occur in a DCS, that is, $\Omega_{CCE} = \{CCE_1, CCE_2, \ldots, CCE_{2^m}\}$. If $\Pr(CCE_j)$ denotes the probability of CCE_j occurring, then we have $\sum_{j=1}^{2^m} \Pr(CCE_j) = 1$ and $CCE_i \cap CCE_j = \phi$ for any $i \neq j$.

For our example DCS presented before, the CCE space is composed of $2^2 = 4$ CCE, that is, $\Omega_{CCE} = \{CCE_1, CCE_2, CCE_3, CCE_4\}$, because there are 2 elementary common-causes CC_1 (earthquakes) and CC_2 (power failures). Each CCE_i is a distinct and disjoint combination of elementary CC, as defined in the first column of Table 1.

Let A_{CCE_i} denote a set of components, which are the only ones affected by the CCE_i. A_{CCE_i} is simply the union of those CCG whose corresponding elementary common-causes occur, as shown in the second column of Table 1.

Because the two CC occurring in the example DCS are independent, we can calculate the occurrence probability of each CCE as follows: $\Pr(CCE_1) = (1-P_{CC1})(1-P_{CC2})$, $\Pr(CCE_2) = P_{CC1}(1-P_{CC2})$, $\Pr(CCE_3) = (1-P_{CC1})P_{CC2}$, and $\Pr(CCE_4) = P_{CC1}P_{CC2}$. The values of $\Pr(CCE_i)$ for the example DCS are shown in the third column of Table 1.

Table 1: CCE, affected components, probabilities

CCE_i	A_{CCE_i}	$\Pr(CCE_i)$
$CCE_1 = \overline{CC_1} \cap \overline{CC_2}$	ϕ	0.996003
$CCE_2 = CC_1 \cap \overline{CC_2}$	$CCG_1=\{e_2,e_5,n_4\}$	0.000997
$CCE_3 = \overline{CC_1} \cap CC_2$	$CCG_2=\{n_1, n_2\}$	0.002997
$CCE_4 = CC_1 \cap CC_2$	$CCG_1 \cup CCG_2=$ $\{n_1,n_2,n_4,e_2,e_5\}$	0.000003

As an illustration, we also show the calculation procedure for $\Pr(CCE_i)$ in case of two CC being mutually exclusive or being s-dependent. If elementary common-causes CC_1 and CC_2 are mutually exclusive, then $\Pr(CCE_1) = 1 - P_{CC1} - P_{CC2}$, $\Pr(CCE_2) = P_{CC1}$, $\Pr(CCE_3) = P_{CC2}$, and $\Pr(CCE_4) = 0$. The calculation is slightly different when the two CC are s-dependent. For example, floods (CC_2) often occur in conjunction with hurricanes (CC_1). Suppose that the probability of CC_1 occurring is P_{CC1}, that $\Pr\{CC_2 \mid CC_1\} = p$, and that $\Pr\{CC_2 \mid \text{no } CC_1\} = q$. The CCE occurrence probabilities $\Pr(CCE_i)$ can be calculated as: $\Pr(CCE_1) = (1 - P_{CC1}) \cdot (1 - q)$, $\Pr(CCE_2) = P_{CC1} \cdot (1 - p)$, $\Pr(CCE_3) = (1 - P_{CC1}) \cdot q$, and $\Pr(CCE_4) = P_{CC1} \cdot p$.

Based on the CCE space we developed and the Total Probability Theorem, we calculate the $\Pr(E|E_2)$ in Eq. (1) as:

$$\begin{aligned}\Pr(E \mid E_2) &= \Pr\{\text{Program fails} \mid \text{no uncovered failure}\} \\ &= \sum_{i=1}^{2^m} [\Pr\{\text{Program fails} \mid CCE_i \cap \text{no uncovered failure}\} \\ &\qquad \bullet \Pr(CCE_i)] \\ &= \sum_{i=1}^{2^m} [DPUR_i \bullet \Pr(CCE_i)] \qquad (3)\end{aligned}$$

As described above, $\Pr(CCE_i)$ in Eq. (3) can be obtained based on the statistical relationship between the elementary common-causes and the occurrence probabilities of elementary CC (P_{CC}), which are given

as input parameters. $DPUR_i$ is a conditional probability that the distributed program fails conditioned on the occurrence of CCE_i and no uncovered failure. The evaluation of $DPUR_i$ is actually a reduced DCS reliability problem in which the components affected by CCE_i (A_{CCE_i}) do not appear and no further attention to IPC and CCF is required. Since both IPC and CCF are out of the picture, traditional DCS reliability analysis approaches that ignore both IPC and CCF can now be applied to solve those reduced reliability problems $DPUR_i$. In the following, we present an efficient ROBDD-based approach for solving $DPUR_i$.

Solving reduced problems $DPUR_i$: It has been shown by many studies that in most cases, ROBDD-based algorithms require less memory compared with other methods and can perform exact and efficient calculation for large system reliabilities[2,7,24]. In the following we present a four-step ROBDD-based approach to the evaluation of $DPUR_i$ in Eq. (3).

Step 1: Obtain the set of MFST using the algorithm based on a breadth-first search, described in[20].

Step 2: Order all the DCS components including nodes and links using a good variable ordering heuristic. A heuristic is good in the sense that it yields a compact BDD[25].

Step 3: Generate the ROBDD for the failure function of a DPR from the MFST using an algorithm similar to the one described in[7].

Step 4: Evaluate $DPUR_i$ recursively from the ROBDD using the modified failure probability $\tilde{q}_i(t)$. The evaluation algorithm is the same as the traditional BDD evaluation[6].

For the example DCS, the set of MFST for program P_1 includes: $MFST_1 = \{n_1, e_1, n_2\}$, $MFST_2 = \{n_4, e_5, n_3\}$, $MFST_3 = \{n_1, e_4, n_3, e_3, n_2\}$, $MFST_4 = \{n_4, e_2, n_2, e_3, n_3\}$. We use ordering of $n_1 < n_2 < e_1 < e_4 < n_3 < n_4 < e_5 < e_3 < e_2$ to generate the ROBDDs.

There are four reduced problems for the example DCS: $DPUR_i$, i=1,2,3,4. The $DPUR_4$ is simply 1 because when CCE_4 occurs, all MFST for program P_1 fail. According to the components affected by CCE_i (i.e. A_{CCE_i}), the ROBDD of $DPUR_1$ is generated from all the four MSFT; ROBDD of $DPUR_2$ is generated from $MFST_1$ and $MFST_3$, and ROBDD of $DPUR_3$ is generated from $MFST_2$. Figure 3 and Figure 4 show the ROBDD for the first three reduced problems. Evaluation of them gives: $DPUR_1$ = 7.6825e-8, $DPUR_2$ = 1.9807e-4 and $DPUR_3$ = 3.8793e-4.

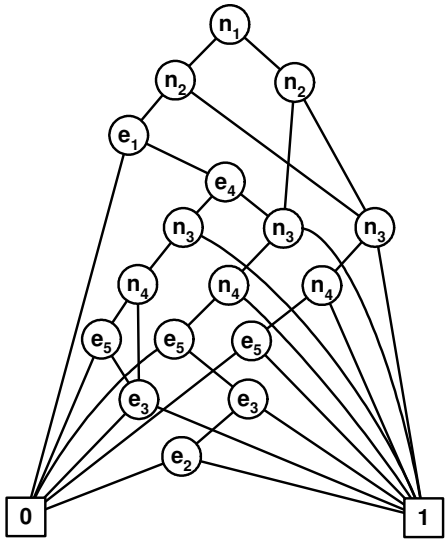

Fig. 3: ROBDD for $DPUR_1$

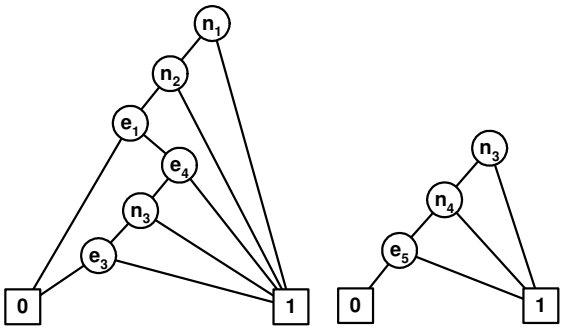

Fig. 4: ROBDD for $DPUR_2$ and $DPUR_3$

Integrating results: Based on the discussion, we integrate the results of $DPUR_i$ with Pr(CCE_i) using Eq. (3) to generate Pr($E|E_2$). Then we integrate the result of Pr($E|E_2$) with Pr(E_2) using Eq. (1) to fulfill the task of distributed program reliability analysis for program P_1. Table 2 summarizes the integration process and the results. As a comparison, the distributed program unreliability for program P_1 without considering IPC and CCF is 8e-8 for the example DCS, which shows that IPC and CCF contribute significantly to the system unreliability and must be properly considered for the accurate analysis of DCS reliability.

Summary of the DCS analysis approach: Our DPR analysis approach for DCS subject to IPC and CCF first separates the consideration of IPC from the solution combinatorics and then decompose the resulted simplified problem into a number of reduced problems according to Total Probability Theorem. The effects of both IPC and CCF are factored out through the above

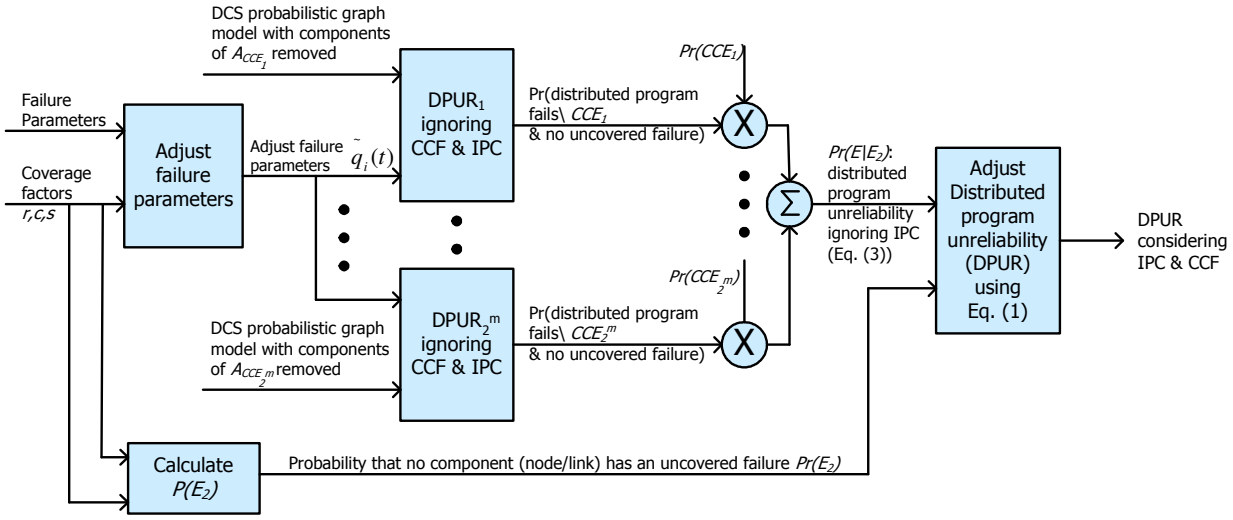

Fig. 5: A conceptual overview of the proposed DPR analysis approach

two-phase reductions. The reduced problems are solved using efficient ROBDD based method. Figure 5 shows a conceptual overview of the proposed separable approach.

Table 2: Results of the example DCS

$\Pr(E_2)$	0.999946
$DPUR_i$	$DPUR_1 = 7.6825e\text{-}8$ $DPUR_2 = 1.9807e\text{-}4$ $DPUR_3 = 3.8793e\text{-}4$ $DPUR_4 = 1$
$\Pr(E\|E_2)$	$\sum_{i=1}^{4}[DPUR_i \bullet \Pr(CCE_i)]$ $= 4.4366e-6$
$DPUR(P_1)$	$1-\Pr(E_2)+\Pr(E\mid E_2)\bullet\Pr(E_2)$ $= 5.843e-5$

The advantages of our approach are that it allows reliability engineers to use their favorite software package that ignores both IPC and CCF for computing distributed program reliability, and adjust the input and output of the program slightly to produce the DPR measure considering both IPC and CCF. As shown through the example, due to the nature of the ROBDD and the separation of IPC and CCF from the solution combinatorics, our approach has higher computational efficiency and is easier to implement than other potential methods such as Markov chain based methods, which can accommodate IPC and CCF by expanding the state space and number of transitions, worsening the state explosion problem.

CONCLUSION

In this study, we presented a separable and efficient ROBDD-based approach for DPR analysis of DCS with IPC and dependent CCF. Our approach enables the analysis of multiple common-causes that can affect different subsets of system components, and which may be *s*-dependent. We illustrate the proposed approach by considering the DPR analysis of a DCS subject to two common-causes. The efficiency of our approach means that it can be easily applied to the accurate reliability (including both DPR and DSR) analysis for large-scale DCS subject to IPC and CCF.

REFERENCES

1. Tanenbaum, A.S. and M.V. Steen, 2002. Distributed Systems: Principles and Paradigms, Prentice Hall.
3. Amari, S.V., J.B. Dugan and R.B. Misra, 1999. A separable method for incorporating imperfect coverage into combinatorial models. IEEE Trans. on Reliability, 48: 267-274.
2. Dugan, J.B. and S.A. Doyle, 1996. New results in fault-tree analysis. tutorial notes: Annual Reliability & Maintainability Symposium.
4. Boyd, M.A., 1989. Generating and solving Markov models with imperfect coverage in one step. Technical Report, Dept. of Computer Science, Duke University.

5. Lopez-Benitez, N., 1994. Dependability modeling and analysis of distributed programs. IEEE Trans. on Software Eng., 20: 345-352.
6. Xing, L. and J.B. Dugan, 2002. Analysis of generalized phased-mission system reliability, performance and sensitivity. IEEE Trans. on Reliability, 51: 199-211.
7. Zang, X., H. Sun and K.S. Trivedi, 1999. Dependability Analysis of Distributed Computer Systems with Imperfect Coverage. In Proc. 29th Ann. Intl. Symp. Fault-Tolerant Computing, pp: 330-337.
8. Hoyland, A. and M. Rausand, 1994. System Reliability Theory: Models and Statistical Methods. John Wiley and Sons, 12.
9. Vaurio, J.K., 1998. An implicit method for incorporating common-cause failures in system analysis. IEEE Trans. on Reliability, 47: 173-180.
10. Bai, D.S., W.Y. Yun and S.W. Chung, 1991. Redundancy optimization of k-out-of-n systems with common-cause failures. IEEE Trans. on Reliability, 40: 56-59.
11. Levitin, G., 2001. Incorporating common-cause failures into nonrepairable multistate series-parallel system analysis. IEEE Trans. on Reliability, 50: 380-388.
12. Pham, H., 1993. Optimal cost-effective design of triple-modular-redundancy-with-spares systems. IEEE Trans. on Reliability, 42: 369-374.
13. Sharma, G.C., L.R. Goel and P. Gupta, 1985. Stochastic analysis of a parallel system with common cause failures, preventive maintenance and two types of repair. Microelectronics and Reliability, 5: 1035-1039.
14. Anderson, P.M. and S.K. Agarwal, 1992. An improved model for protective-system reliability. IEEE Trans. on Reliability, 41: 422–426.
15. Chae, K.C. and G.M. Clark, 1986. System reliability in the presence of common-cause failures. IEEE Trans. on Reliability, R-35, pp: 32-35.
16. Fleming, K.N., N. Mosleh and R.K. Deremer, 1986. A systematic procedure for incorporation of common cause events into risk and reliability models. Nuclear Eng. and Design, 93: 245-273.
17. Tang, Z. and J.B. Dugan, 2004. An integrated method for incorporating common cause failures in system analysis. In Proc. Annual Reliability and Maintainability Symp., pp: 610-614.
18. Amari, S.V., J.B. Dugan and R.B. Misra, 1999. Optimal reliability of systems subject to imperfect fault-coverage. IEEE Trans on Reliability, 48: 275-284.
19. Lin, M., D. Chen and M. Horng, 1999. The reliability analysis of distributed computing systems with imperfect nodes. The Computer J., 42: 129-141.
20. Kumar, V.K.P., S. Hariri and C.S. Raghavendra, 1986. Distributed program reliability analysis. IEEE Trans. on Software Engg., 12: 42-50.
21. Powell, D., E. Martins, J. Arlat and Y. Crouzet, 1995. Estimators for fault tolerance coverage evaluation. IEEE Trans. on Computers, 44: 261-274.
22. Smith, D.T., B.W. Johnson, J.A. Profeta III and D.G. Bozzolo, 1995. A fault list generation algorithm for the evaluation of system coverage. In Proc. Annual Reliability & Maintainability Symp., pp: 425-432.
23. Page, L.B. and J.E. Perry, 1989. A model for system reliability with common-cause failures. IEEE Trans. on Reliability, 38: 406-410.
24. Bryant, R., 1986. Graph-based algorithms for boolean function manipulation. IEEE Trans. on Computers, C-35, 8: 677-691.
25. Bouissou, M., 1986. An ordering heuristic for building binary decision diagrams from fault-trees. In Proc. Annual Intl. Reliability and Maintainability Symp., pp: 208-214.