Formal Methods in System Design (2024) 62:285–325
https://doi.org/10.1007/s10703-024-00448-z

Dynamic dependability analysis of shuffle-exchange networks

Yassmeen Elderhalli[1] · **Osman Hasan**[1] · **Sofiène Tahar**[1]

Received: 19 March 2021 / Accepted: 6 February 2024 / Published online: 15 May 2024

Abstract
Designing dependable multiprocessor systems requires reliable interconnection networks. Multistage interconnection networks (MINs), including shuffle-exchange networks (SENs), are widely used to establish the desired connection. The failure of these networks can degrade the overall system performance, which may lead to significant losses. In this paper, we propose to formally model and analyze the dynamic dependability aspects of SENs using a combination of dynamic fault trees (DFTs) and dynamic reliability block diagrams (DRBDs) based on higher-order logic (HOL) theorem proving. We propose to integrate these two modeling approaches for efficiently handling the considered formal dependability analysis by leveraging upon the advantages of each method. The soundness of this integration is provided through a formal proof of equivalence between the DFT and DRBD algebras. We utilize the proposed framework to provide the formal DFT and DRBD analyses of three common measures of SENs, namely: terminal, broadcast and network reliability. The proposed approach allowed us to verify generic expressions of probability of failure and reliability of these systems, which can be instantiated with any number of system components and time-to-failure functions.

Keywords Dynamic dependability analysis · Shuffle-exchange networks · Dynamic fault trees · Dynamic reliability block diagrams · Theorem proving · Higher-order logic · HOL4

1 Introduction

Multiprocessor systems are increasingly being used to meet the ongoing demand for intensive processing applications, due to their cost-effectiveness and the feasibility to utilize hundreds of processors. These multiprocessor systems allow parallel computing, to enhance the overall

✉ Yassmeen Elderhalli
y_elderh@ece.concordia.ca

Osman Hasan
o_hasan@ece.concordia.ca

Sofiène Tahar
tahar@ece.concordia.ca

[1] Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada

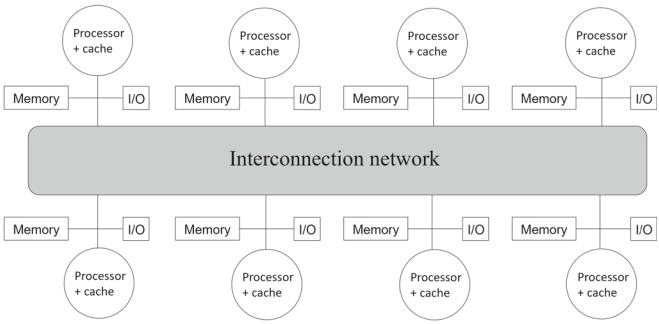


Fig. 1 Overview of multiprocessor system architecture.

system performance, such as throughput. However, memory and I/O peripheral resources are shared among the processors and thus an efficient data routing among system nodes is necessary to maintain high system performance, reliability and low cost. This is of a great importance, particularly in scientific applications, where a huge number of processors are used, i.e., large-scale multiprocessor systems [19]. Therefore, a dedicated interconnection network is used to connect processors and memory modules, as depicted in Fig. 1 [19].

The complexity of interconnection networks ranges from simple networks, such as time-shared bus to complicated ones, such as crossbar switching. The former has a negative impact on the system performance, while the latter has much higher cost as there exists a separate link between each pair of nodes in the systems. For example, for a system of N nodes, i.e., N inputs and N outputs, it is required to have N^2 links or switching elements between each input and output.

Multistage interconnection networks (MINs) have been introduced to reduce the number of required switching elements and hence, reduce the cost while providing better performance than shared-bus networks [20]. The main idea of MINs is to have multiple small stages of crossbar switches that are connected between sources (inputs) and destinations (outputs), which results in a considerable reduction in the number of used switching elements. The number of paths available between each input and output determines the category of the MIN. A single-path MIN has only one path to route information between each source-destination pair. A shuffle-exchange network (SEN) is an example of such type of networks. Each stage has $N/2$ switching elements, where N is the number of inputs and outputs of the network. Usually the switching elements are of size 2×2 to reduce the cost. The number of stages required to establish the single-path MIN is $log_2 N$, which is lower than crossbar networks. For instance, an 8×8 SEN is shown in Fig. 2, where only a single path is available for each input–output pair. However, the reliability of single-path MINs and SENs depends on the switching elements and thus a fault in any of these switches cannot be tolerated.

Enhancing the dependability of MINs, i.e., its ability to provide a trusted service [3], is of great importance in order to maintain high system performance. Therefore, redundant switching elements are used to ensure that the network is able to provide the required switching even after the failure of some of these elements [1, 21]. Thus, multiple-path MINs are used to increase the fault tolerance and hence the network reliability. For instance, SEN+ [33] is a SEN, where an additional stage is added to provide two paths between each input–output pair, as shown in Fig. 3. However, even with the additional path, the failure of some switches can lead to the failure of the connection in some situations. For example, in Fig. 3, if the first two switches in stage 2 fail, then the paths between source 0 and destination 0 are not

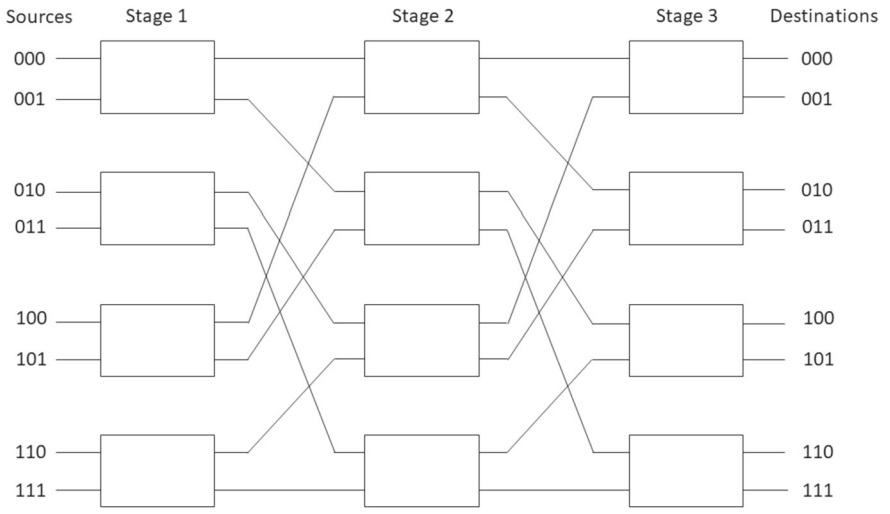


Fig. 2 An 8×8 SEN.

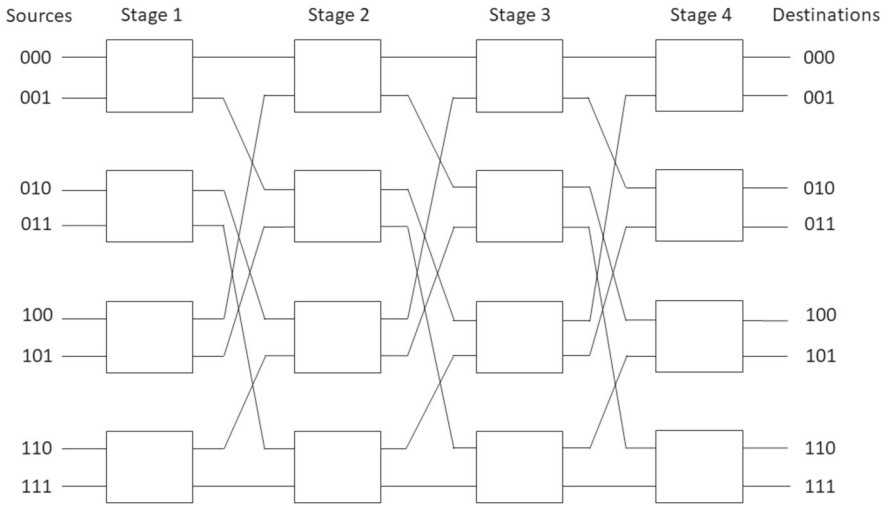


Fig. 3 An 8×8 SEN+.

available anymore. Therefore, spare parts can be used, such as in [20], to replace switches after failure.

Studying the reliability of SENs has been an active research area [5, 28, 33, 34]. The reliability of MINs are commonly analyzed using simulation or analytically. For example, in [17], Monte Carlo simulation was used to analyze the reliability of SENs. However, simulation cannot provide accurate results due to its sampling based nature. Although Continuous Time Markov Chains (CTMCs) can be used to analytically determine the reliability of MINs [30], they cannot be used with large-scale systems since the state space grows exponentially with the increase in the number of system components. On the other hand, when the complexity of the network increases, reliability bounds provide estimate values for the MIN reliability

[16, 35]. Reliability block diagrams (RBDs) have been also used in the analysis of MINs with single and multiple paths. For instance, in [4], the reliability of SEN, SEN+ and SEN+2 (a SEN with two additional stages) is modeled using traditional RBDs. Generic expressions of success rates of the switching elements are provided analytically assuming that all these elements have the same failure rate. However, these generic expressions are not formally verified, which may raise questions about their accuracy. Furthermore, dynamic dependencies among system components, where not only the failure of a component affects the failure of the system, but also the time of failure and the dependency of one component on another, is an important aspect that has not been considered or modeled in the literature. For example, using spare switches can enhance the system reliability.

Dynamic dependability models, such as dynamic fault trees (DFTs) [31] and dynamic reliability block diagrams (DRBDs) [7], capture the dynamic failure and success dependencies, respectively, among system components, and hence are more suitable in modeling real-world systems, such as MINs. Recently, higher-order-logic (HOL) theorem proving has been used in the formal analysis of both models algebraically [11, 13], where generic expressions are formally verified that are independent of the failure distributions of system components. This ensures the soundness of the analysis, which is suitable for safety-critical systems.

Based on the previous discussion, accurate modeling and analysis of these networks is necessary to capture the dynamic behavior as this will provide design engineers with some measures that can help enhancing the performance of the entire multiprocessor system. In a first attempt to use higher-order logic theorem proving for the dependability analysis of MINs, in [11], we formally analyzed the terminal reliability of SEN+ as a case study using higher-order logic theorem proving. We analyzed the reliability of the network using DRBD while adding spare switches to replace the critical ones after failure. However, we did not take into account other aspects of SENs, such as the broadcast and network reliability. Furthermore, in [11], we have not considered generic versions of the given networks. In this paper, we propose to use both formalizations in conducting the dynamic dependability analysis of SENs of multiprocessor systems. We propose to integrate the DFT and DRBD formalizations in a framework that allows the formal analysis of these models using a theorem prover. Furthermore, this framework provides a bidirectional path between both formalizations allowing reasoning about both the failure and success of a given system. In this work, we utilize this framework to formally verify the terminal, broadcast and network reliability of SEN and SEN+ in HOL and provide generic expressions of reliability and probability of failure. In addition, we verify the equivalence of these models, which allows using the analysis results of one model in reasoning about the other one.

The rest of the paper is structured as follows: Sect. 2 provides a brief description of the available DFT and DRBD formalizations in HOL4. Section 3 describes the proposed framework for DFT-DRBD formal analysis. Sections 4, 5 and 6 provide the formal verification of the terminal, broadcast and network reliability analysis, respectively, of SEN and SEN+ along with the formal equivalence of their dynamic dependability models. Finally, we conclude the paper in Sect. 7.

2 Preliminaries

In this section, we provide some preliminaries related to the HOL4 probability, DFT and DRBD theories that are required for understanding the rest of the paper.

2.1 Probability theory in HOL4

A probability space is defined as a triplet $(\Omega, \mathcal{A}, \mathcal{P}r)$, where Ω is the sample space, $\mathcal{A}$ is the set of probability events and $\mathcal{P}r$ is the probability measure [26]. The HOL function `p_space p` returns the sample space (Ω) of the above triplet, while `events p` returns the set of events ($\mathcal{A}$). A random variable is a measurable function that maps the probability space p to another measurable space. It is defined in HOL as [26]:

Definition 1

```
⊢ ∀X p s. random_variable X p s ⇔
    prob_space p ∧ X ∈ measurable (p_space p, events p) s
```

In the definition above, `X` is the random variable, `p` is the probability space and `s` is the space that the random variable maps to. In our work, we use the `borel` space, which is defined over the real line [29].

The probability distribution is defined as the probability that a random variable, X, belongs to a certain set, A [24]:

Definition 2

```
⊢ ∀p X. distribution p X =
     (λs. prob p (PREIMAGE X s ∩ p_space p))
```

The cumulative density function (CDF) is defined as [2]:

Definition 3

```
⊢ ∀p X t. CDF p X t = distribution p X {y | y ≤ (t:real)}
```

In the definition above, `p` is a probability space, `X` is a real-valued random variable and `t` is a variable of type real and represents time.

When two random variables are independent, then the probability of the intersection of their events equals the multiplication of the probabilities of the individual events. This definition is ported from Isabelle/HOL [27] to HOL4 as [29]:

Definition 4

```
⊢ indep_vars p M X ii =
    (∀i. i ∈ ii ⇒
       random_variable (X i) p
          (m_space (M i), measurable_sets (M i))) ∧
    indep_sets p
       (λi.
         {PREIMAGE f A ∩ p_space p |
            (f = X i) ∧ A ∈ measurable_sets (M i)}) ii
```

This definition ensures that a group X is composed of random variables indexed by the elements in set `ii` and that the events represented by the preimage of these random variables are independent using `indep_sets`. Based on Definition 4, `indep_var` is defined to capture the behavior of independence for two random variables [29].

The probabilistic *Principle of Inclusion and Exclusion* (PIE) can be used to express a relationship between the probability of the union of different events as:

$$Pr\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{t\neq\{\},t\subseteq\{1,2,\ldots,m\}} (-1)^{|t|+1} Pr\left(\bigcap_{j\in t} A_j\right) \tag{1}$$

Table 1 HOL4 probability functions

Function	Explanation
`rv_gt0_ninfinity L`	Random variables in list L are greater than 0 and not equal to $+\infty$
`indep_var p lborel (real o X) lborel (real o Y)`	Independence of input random variables defined from the probability space `p` to the Lebesgue Borel measure (`lborel`)
`distributed p lborel (real o X)` f_X	Defines a density function f_X for the real version of random variabe X defined from the probability space p to the Lebesgue–Borel measure
`measurable_CDF p (real o Y)`	Ensures that CDF (F_Y) is measurable
`cont_CDF p (real o Y)`	Ensures that CDF (F_Y) is contiunous
`cond_density lborel lborel p (real o X)(real o Y) y` f_{XY} f_Y $f_{X_a\|Y}$	Defines a conditional density function $f_{X\ a}\|Y$ using the joint density function f_{XY} and the marginal density function f_Y
`den_gt0_ninfinity` f_{X_a} $_Y f_Y$ $f_{X_a\|Y}$	Ensures the proper values for the density functions; joint, marginal and conditional, respictively. $0 \leq f_{X_a Y}$, $0 < f_Y$ and $0 \leq f_{X_a\|Y}$
`indep_sets p X s`	Ensures that the group of sets X indexed by the numbers in set s are independent over the probability space p

It is formally verified in HOL4 in [2], where it is used to express the probability of union of list of events, L:

Theorem 1

```
⊢ ∀p L.
    prob_space p ∧ (∀ x. MEM x L ⇒ x ∈ events p) ⇒
    (prob p (union_list L =
     sum_set {t | t ⊆ set L ∧ t ≠ {}}
      (λt. -1 pow (CARD t+1) * prob p (⋂ t))
```

The Lebesgue integral is defined in HOL4 based on positive functions and functions with positive and negative values [25]. In this work, we use the Lebesgue integral to integrate cumulative density functions and probability density functions, which are always positive. Thus, we use the Lebesgue integral for positive functions, i.e., `pos_fn_integral`. We integrate over the real line and thus we use the Lebesgue-Borel measure (`lborel`) [29] for this purpose. The boundaries of this integral can be identified using an indicator function by specifying the set of elements used in the integration. For example, $\int_A f(x)\,dx$ can be represented as `pos_fn_integral lborel` (λx. `indicator_fn A * f x`). However, for the ease of understanding, we use the regular mathematical expressions, i.e., we use $\int_A f(x)\,dx$ to express the integrals instead. Table 1 lists the probability theory functions used in the rest of the paper, which are explained in [9].

2.2 Dynamic fault trees

A fault tree (FT) is a graphical representation of the the sources of failure that lead to the failure of a given system using fault tree gates, such as AND and OR gates [32]. A failure in this context means that the system will stop delivering its proper functionality. Static fault trees (SFTs) only consider the sources of failure without taking into account the

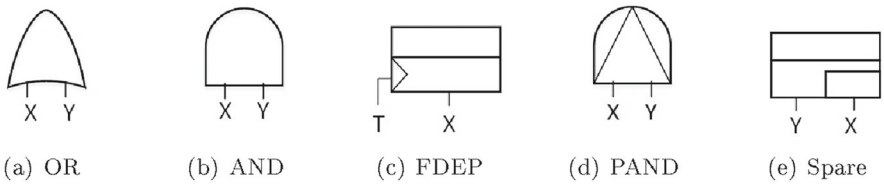


Fig. 4 DFT gates.

Table 2 Definitions of DFT temporal operators

Operator	Mathematical expression	Formalization
`Before`	$A \lhd B = \begin{cases} A, & A < B \\ +\infty, & A \geq B \end{cases}$	`⊢ ∀ A B. D_BEFORE A B = (λs. if A s < B s then A s else PosInf)`
`Simultaneous`	$A \Delta B = \begin{cases} A, & A = B \\ +\infty, & A \neq B \end{cases}$	`⊢ ∀ A B. D_SIMULT A B = (λs. if A s = B s then A s else PosInf)`
`Inclusive Before`	$A \trianglelefteq B = \begin{cases} A, & A \leq B \\ +\infty, & A > B \end{cases}$	`⊢ ∀ A B. D_INCLUSIVE_BEFORE A B = (λs. if A s ≤ B s then A s else PosInf)`

dependencies among system components. However, SFTs cannot capture the behavior of real-world systems, where such failure dependencies are common. For instance, the failure of a main part can activate its spare. Dynamic fault trees (DFTs) can capture these dependencies using DFT gates (Fig. 4) [32]. As an example, consider the Priority AND (PAND) gate (Fig. 4d), which models the sequence of events that can lead to the failure of another event in the system. Such behavior, which is a realistic one, cannot be modeled using the traditional SFT gates.

In order to analyze a given DFT formally within a theorem prover, it is required to have some kind of algebra to mathematically capture the behavior of the gates and thus model the DFT. In [23], a DFT algebra is proposed to enable expressing the output of a given DFT (structure function) based on the system components. The algebraic approach based DFT analysis relies on presenting the basic events, which represent system components, and the output of DFT gates based on their time of failure [23]. Identity elements are defined to express two states of system components. The ALWAYS element represents a component that already failed, i.e., the time of failure equals 0. The NEVER element models a fail safe component, which means that its time of failure equals $+\infty$. Three temporal operators are also introduced in [23], i.e., *Before* ($\lhd$), *Simultaneous* (Δ) and *Inclusive-before* ($\trianglelefteq$), to model the dynamic behavior of one event failing before the other, at the same time and before or at the same time, respectively [23]. In [10], we provided the HOL formalization of these operators (Table 2), where we defined them as lambda abstracted functions that return extended-real numbers (`extreal`), which include real numbers and $\pm\infty$ to model the NEVER element. In [23], several simplification theorems are presented to help reduce the structure function of a given DFT. However, without any formal proofs, the soundness of such reduction cannot be guaranteed. In [10], we formally verified these theorems to ensure their correctness and the soundness of the analysis as well.

Table 3 DFT gates expressions and probability of failure

Gate	Mathematical expression	Probability of failure
AND	$X \cdot Y = max(X, Y)$	$F_X(t) \times F_Y(t)$
OR	$X + Y = min(X, Y)$	$F_X(t) + F_Y(t) - F_X(t) \times F_Y(t)$
PAND	$Q_{PAND} = \begin{cases} Y & X \leq Y \\ +\infty & X > Y \end{cases}$	$\int_0^t f_Y(y)\, F_X(y)\, dy$
FDEP	$X + Y = min(X, Y)$	$F_X(t) + F_Y(t) - F_X(t) \times F_Y(t)$
Spare	$Q_{SP} = Y \cdot (X_d \lhd Y) + X_a \cdot (Y \lhd X_a) + Y \Delta X_a + Y \Delta X_d$	$\int_0^t \left(\int_v^t f_{(X_a \mid Y=v)}(u)du \right) f_Y(v)dv + \int_0^t f_Y(u) F_{X_d}(u)du$

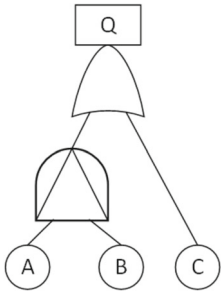


Fig. 5 DFT example.

In [23], the DFT gates, shown in Fig. 4, are modeled based on the time of failure of their output. For instance, the output of the AND gate fails when both inputs fail. Thus, the time of failure of the output equals to the maximum time of failure of both inputs. The Priority-AND (PAND) gate models the sequence of failure dependencies in a system. The Functional DEPendency (FDEP) gate is used to model failure triggers of system components. The time of failure of the triggered component equals to the minimum of the time of failure of the trigger or the component as the latter may fail due to a failure of the trigger or the failure of the component itself. The spare gate models spare parts in a system, where the spare (X) replaces a main part (Y) after its failure. In the general case, the failure distribution of the spare is attenuated by a dormancy factor from the active state. Therefore, in the DFT algebra, two variables are used to distinguish the time of failure of the spare in both its states; active (X_a) and dormant (X_d). Table 3 lists the definitions of these gates, which we formalized in HOL [10].

An example of a DFT is shown in Fig. 5. In this example, there are three inputs to the DFT: A, B, and C, and two gates: OR and PAND. The output of this tree fails if input A fails before B or if input C fails.

In order to verify the probability of failure expression, given in Table 3, we need to define a `DFT_event` to be used in the probabilistic analysis. This event is a set satisfying the condition that the input function is less than or equal to time t, which represents the moment of time until which we are interested in finding the probability of failure. Without this `DFT_event`, there is no possible way to apply the probability directly to DFT gates. We first need to create the `DFT_event` for the time-to-failure function of the output event of any gate and then apply the probability to it. This is formally defined as [10]:

Definition 5

```
⊢ ∀p X t. DFT_event p X t = {s | X s ≤ Normal t} ∩ p_space p
```

In the previous definition, `X` is the time to failure function that can represent inputs and outputs of DFT gates and `t` is the time until which we are interested in finding the probability of failure. The type of `t` is real, while the time to failure functions are of type `extreal` and thus it is required to typecast `t` to `extreal` using the `Normal` function. We verified the probability of failure of all DFT gates based on this event and using their formal definitions, as given in Table 3 [10]. For the rest of the paper, we will omit the typecasting from `real` to `extreal` and vice versa to abstract the presentation.

In [14], we extended the definitions of the AND, OR and PAND gates to n-ary gates in order to be able to verify generic expressions for any number of system components. We defined the n-ary AND as:

Definition 6

```
⊢ ∀ L. n_AND L = FOLDR (λ a b. D_AND a b) ALWAYS L
```

In the previous definition, `D_AND` is the DFT AND gate. `FOLDR` applies a 2-input function, `D_AND` in this case, over a list, `L` that represents the time-to-failure functions of the input events, i.e., the random variables. `ALWAYS` is used in this definition to apply the function to the last element in the list `L`.

We defined the n-ary OR gate as [14]:

Definition 7

```
⊢ ∀ L. n_OR L = FOLDR (λ a b. D_OR a b) NEVER L
```

In the previous definition, `D_OR` is the DFT OR gate. `D_OR` is the 2-input function used with `FOLDR` in this definition. `NEVER` is used here since it will not affect the behavior of the OR gate.

2.3 Dynamic reliability block diagrams

The reliability of a given system is the probability that this system will continue to function as intended. A reliability block diagram (RBD) graphically models the paths that ensure a successful system behavior. These paths are composed of system components that are connected in a series or parallel manner and can be further extended to series–parallel, parallel-series or even more nested hierarchy based on the reliable behavior of the system, as shown in Fig. 6. Usually, systems may encompass dependent failure behaviors among their components. This requires a more advanced modeling approach. Dynamic reliability block diagrams (DRBDs) capture such dependencies using DRBD constructs, like for example the spare and load sharing constructs, shown in Fig. 7. The blocks in a DRBD can be connected in series, parallel, series–parallel and parallel-series, similar to a traditional RBD. Figure 8 shows a simple DRBD, which is composed of a spare construct and an input that are connected in series.

We proposed an algebra that allows expressing the structure function of a given DRBD based on system blocks [11]. The reliability of a given system can be expressed using this DRBD algebra. We defined several operators that enable expressing DRBDs of series and parallel configurations and even more complex structures. Furthermore, the defined operators allow modeling a DRBD spare construct to capture the behavior of spares in a system. We provided the HOL formalization of this algebra to ensure its soundness and enable the formal

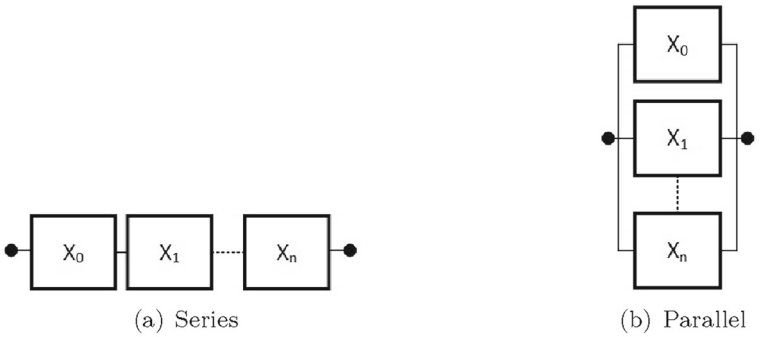


Fig. 6 RBD structures.

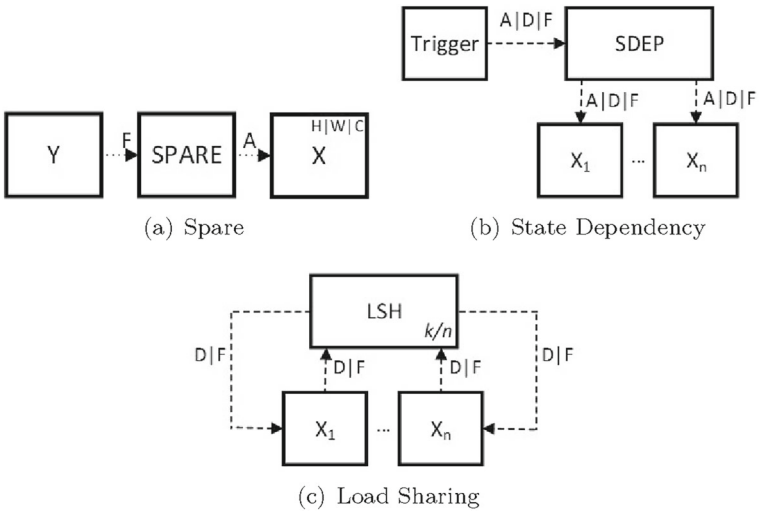


Fig. 7 DRBD constructs.

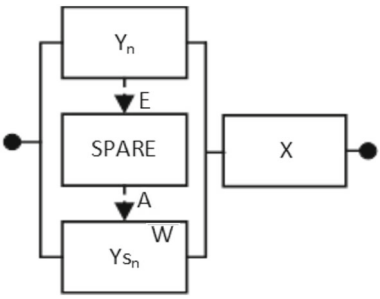


Fig. 8 DRBD example.

Table 4 Definitions of DRBD operators

Operator	Mathematical expression	Formalization
AND	$X \cdot Y = min(X, Y)$	⊢ ∀X Y. R_AND X Y = (λs. min (X s) (Y s))
OR	$X + Y = max(X, Y)$	⊢ ∀X Y. R_OR X Y = (λs. max (X s) (Y s))
After	$X \rhd Y = \begin{cases} X, & X > Y \\ +\infty, & X \leq Y \end{cases}$	⊢ ∀X Y.R_AFTER X Y = (λ s.if Y s < X s then X s else PosInf)
Simultaneous	$X \Delta Y = \begin{cases} X, & X = Y \\ +\infty, & X \neq Y \end{cases}$	⊢ ∀X Y. R_SIMULT X Y = (λs. if X s = Y s then X s else PosInf)
Inclusive After	$X \trianglerighteq Y = \begin{cases} X, & X \geq Y \\ +\infty, & X < Y \end{cases}$	⊢ ∀X Y. R_INCLUSIVE_AFTER X Y = (λs. if Y s≤ X s then X s else PosInf)

analysis using HOL4. We first formally define a DRBD event that creates the set of time until which we are interested in finding the reliability [11]:

Definition 8

```
⊢ ∀p X t. DRBD_event p X t = {s | t < X s} ∩ p_space p
```

In the previous definition, X is the time to failure function of a system component and t is the moment of time until which we are interested in finding the reliability of the system. The probability of this event represents the reliability of the system until time t [11]:

Definition 9

```
⊢ ∀p X t. Rel p X t = prob p (DRBD_event p X t)
```

Then, we verify that its probability is related to the CDF [11].

We introduced DRBD identity elements and operators to model both the combinatorial and dynamic behaviors, as listed in Table 4. The idea is similar to the DFT algebra, where the blocks are modeled based on their time of failure. We need to recall that DRBDs are concerned in modeling the successful behavior, i.e., the “not failing" behavior, and thus we can use the time to failure functions to model the behavior of a given DRBD. We defined two identity elements for DRBD that are similar to the DFT elements, i.e., ALWAYS = 0 and NEVER = $+\infty$. The DRBD operators are listed in Table 4. The AND operator $(\cdot)$ models series DRBD blocks, where it is required that all the blocks are working. The output of the AND operator fails with the first failure of any component of its inputs. On the other hand, the OR operator $(+)$ models parallel structures, where at least one of the blocks should continue to work to maintain the system functionality. To capture the dynamic behavior, we introduced three temporal operators, i.e., *After*, *Simultaneous* and *Inclusive-after* [11]. The after operator $(\rhd)$ models the sequence of events, where the system continues to work as long as one component continues to work after the failure of the other. For example, $(X \rhd Y)$, means that the system continues to work when component X continues to work after the failure of component Y. The simultaneous operator (Δ) is similar to the one of the DFT algebra, where its output fails when both inputs fail at the same time. Finally, the inclusive-after operator

Table 5 Mathematical models and reliability of spare, series and parallel structures

	Mathematical Expression	Reliability
Spare	$(X_a \rhd Y) \cdot (Y \rhd X_d)$	$1 - \int_0^t \int_y^t f_{(X_a \mid Y=y)}(x)\ f_Y(y) dx dy - \int_0^t f_Y(y) F_{X_d}(y) dy$
Series	$\bigcap_{i=1}^{n}(event\ (X_i,\ t))$	$\prod_{i=1}^{n} R_{X_i}(t)$
Parallel	$\bigcup_{i=1}^{n}(event\ (X_i,\ t))$	$1 - \prod_{i=1}^{n}(1 - R_{X_i}(t))$

($\trianglerighteq$) combines the behavior of both after and simultaneous operators. We provided the HOL formalization of these elements and operators based on lambda abstracted functions and `extreal` numbers. The mathematical expressions and the HOL formalization are listed in Table 4. The reliability expressions of these operators are available at [11, 12]. Furthermore, we verified several simplification theorems that enable reducing the structure function of a given DRBD [12].

A spare construct, shown in Fig. 7a, is introduced in DRBDs to model spare parts in systems by having spare controllers that activate the spare after the failure of the main part. In Table 5, Y is the main part and after its failure X is activated. We use two variables $(X_a,\ X_d)$, similar to the space in the DFT algebra to model the active and dormant states, respectively. The spare can have three variants, i.e., hot, warm and cold spare. The cold spare starts working only after the failure of the main part, which means it cannot fail while it is dormant. The hot spare can fail in both the active and dormant states with the same probability. The warm spare can fail in both active and dormant states but with different probabilities. In [12], we formally verified the reliability expressions of these three variants based on the verified reliability expressions of the operators.

DRBD blocks can be connected in series, parallel and more nested structures. We provide here the details of only the series and parallel structures, as listed in Table 5. Details about the nested structures can be found in [11, 12]. The series structure, shown in Table 5, continues to work as long as all the blocks are working. Once one of these blocks stops working, then the entire system stops as well. It can be expressed using the AND operator. Its mathematical model is expressed as the intersection of the individual DRBD events [18]. The parallel structure, shown in Table 5, is composed of several blocks that are connected in parallel. Its structure function can be expressed using the OR operator. Its mathematical model is represented using the union of the individual DRBD events. We developed the HOL formalization of these structures and verified their reliability expressions assuming the independence of the individual blocks [11].

3 Framework for formal dynamic dependability analysis

In order to efficiently use the above formalized DFT and DRBD algebras for conducting the formal dynamic dependability analysis of SENs, we propose in this section a comprehensive framework, which enables the modeling and analysis of DFTs and DRBDs in HOL and produces generic expressions of dependability. Figure 9 depicts the proposed dependability analysis framework, which starts by reading a given system description that can be modeled as a DFT to model the failure or DRBD to model the success. Based on the library of

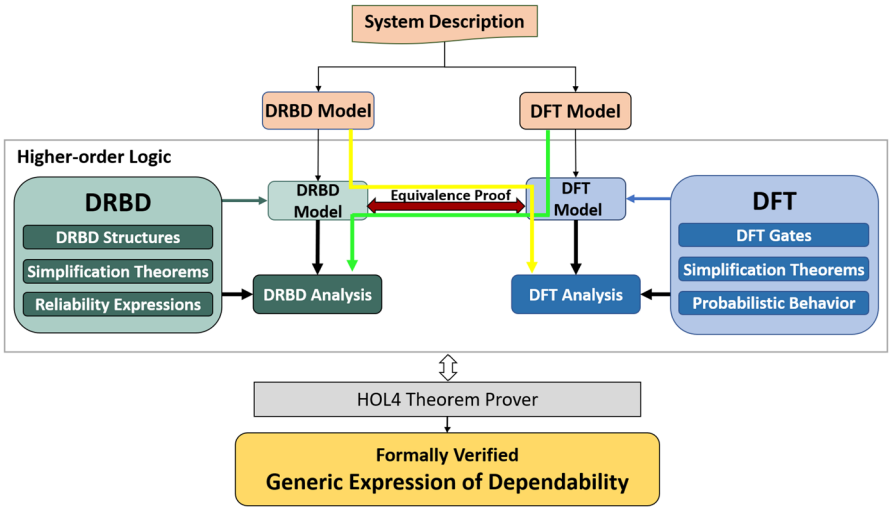


Fig. 9 Framework for formal dynamic dependability analysis using HOL4.

formalized DFT gates and DRBD constructs, a formal dependability model of the given system can be created. The libraries also include the verified simplification theorems for both algebras, which enables formally verifying a reduced dependability model of the given system. This ensures the soundness of the reduction process and allows conducting the analysis of this reduced model instead of the original model. Based on the reduced model, the qualitative analysis of the given system can be performed to determine the sources of failure, in case of a DFT, or success, in case of a DRBD. More specifically, the cut sets and cut sequences are identified, where the former represent the sets of basic events that lead to the failure/success of the top event without considering the order of occurrence, while the latter require determining the exact sequence of occurrences that lead to this failure/success. The formally verified probabilistic behavior of the DFT gates and operators as well as the reliability expressions of the DRBD constructs are included in our library. We perform the quantitative analysis by verifying generic expressions of dependability, i.e., probability of failure for DFTs and reliability for DRBDs. The importance of this framework lies in the fact that these verified expressions are generic and independent of the failure distributions of the basic events. This represents an advantage over model checking based approaches, where only exponential distributions are considered and the failure rates have to be identified in advance before starting the analysis.

Our proposed framework also allows formally converting a DFT model into its corresponding DRBD and vice-versa based on the equivalence of both algebras. This implies that a DRBD model can be analyzed as described in Sect. 2.3, where a DRBD event is created and its reliability is verified based on the available DRBD algebra verified theorems. The DRBD model can also be converted to a DFT to model the failure instead of the success so that the model is analyzed using the DFT algebra. Similarly, the DFT model can be analyzed based on the DFT algebra, as described in Sect. 2.2, or by converting it to its counterpart DRBD model.

In order to handle the DFT analysis using DRBD algebra and the DRBD analysis using the DFT algebra, it is required to formally prove the equivalence of both algebras (*Equivalence Proof* in Fig. 9). According to [6], the OR, AND and FDEP gates can be represented using

Table 6 Verified equivalence of DFT gates and DRBD algebra

DFT gate	DRBD operator/construct	Verified theorem
AND	OR	⊢ ∀X Y. D_AND X Y = R_OR X Y
OR	AND	⊢ ∀X Y. D_OR X Y = R_AND X Y
FDEP	AND	⊢ ∀X Y. FDEP X Y = R_AND X Y
PAND	Inclusive After	⊢ ∀X Y. P_AND X Y = R_INCLUSIVE_AFTER Y X
Spare	Spare	⊢ ∀X_a X_d Y. (∀s. ALL_DISTINCT [Y s;X_a s;X_d s])⇒ (WSP Y X_a X_d = R_WSP Y X_a X_d)

series, parallel and series RBDs, respectively. Therefore, they can be modeled using AND and OR operators, while the spare gate corresponds to the spare construct. Finally, the PAND gate can be expressed using the inclusive after operator ($Y \trianglerighteq X$). However, we need to formally verify this equivalence to ensure its correctness. In Table 6, we provide the theorems of equivalence of DFT gates and DRBD operators and constructs, where D_AND, D_OR, FDEP, P_AND and WSP are the names of the AND, OR, FDEP, PAND and spare DFT gates in our HOL formalization [10]. R_WSP is the name of the warm spare DRBD construct in our formalized DRBD [11] and the predicate ALL_DISTINCT [Y $X_a X_d$] ensures that the inputs cannot fail at the same time.

In order to use these verified expressions in Table 6, we need to verify that the DRBD_event and the DFT_event possess complementary sets in the probability space. Based on this, we can verify that the probability of DRBD_event complements the probability of the DFT_event. We formally verify this as:

Theorem 2

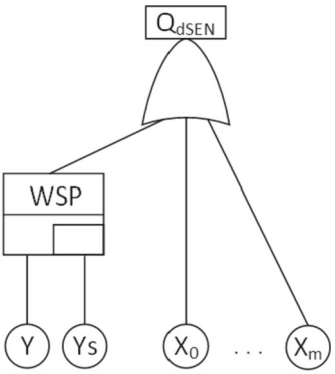
```
⊢ ∀p X t. prob_space p ∧ (DFT_event p X t) ∈ events p ⇒
    (prob p (DRBD_event p X t)= 1 - prob p (DFT_event p X t))
```

In the previous theorem, the conditions ensure that p is a probability space and that the DFT event belongs to the events of the probability space. This theorem can be verified also if we ensure that the DRBD event belongs to the probability space. This theorem means that for the same time to failure function, the DRBD and DFT events are the complements of each other. This way, we can analyze DFTs using the DRBD algebra and vice-versa.

Based on the verification results obtained in Table 6, DFT gates can be formally represented using DRBDs. We show that the amount of effort required by the verification engineer to formally analyze DFTs by analyzing its counterpart DRBD is less than that of analyzing the original DFT model. In Sect. 2.2, a DFT is formally analyzed using the DFT algebra by expressing the DFT event of the structure function as the union of the individual DFT events. If the probabilistic PIE is utilized to formally verify the probability of failure of the top event, then the number of terms in the final result equals $2^n - 1$, where n is the number of individual events in the union of the structure function. Therefore, in the verification process, it is required to verify at least $2^n - 1$ expressions. On the other hand, verifying a DRBD would require verifying a single expression for each nested structure.

In the following sections, we utilize this framework, to conduct the dynamic dependability analysis of shuffle-exchange networks, i.e., terminal, broadcast and network reliability analysis.

Fig. 10 DFT of SEN.



4 Terminal reliability analysis of shuffle-exchange networks

The terminal reliability is the reliability of the connection between a given source and destination, i.e., the probability of having a reliable connection between one source-destination pair. In this section, we analyze the terminal reliability of the SEN and SEN+ using both DFT and DRBD models and verify their equivalence.

4.1 DFT analysis of SEN and SEN+

We model the sources of failure of both SEN and SEN+ using DFTs. We use n-ary gates, which enable verifying expressions of the probability of failure for a generic number of system components.

Figure 10 shows the DFT model of the SEN system. Since SENs are single path MINs, the failure of any of the switches in the path between a given source and destination leads to losing the connection. Therefore, adding spare parts will lower the probability of failure. For illustration purposes, we use a spare part (WSP) to replace the main switch Y after failure. The DFT consists of an n-ary OR gate, which means that the failure of any of the switches interrupts the connection between the source and the destination.

Since the top event is an n-ary OR gate, we need first to verify that the `DFT_event` of the n-ary OR is equal to the union of the individual events as:

Theorem 3

```
⊢ ∀ p X t s. FINITE s ⇒
  (DFT_event p (n_OR (MAP_SET_LIST X s)) t =
   ⋃ᵢ∈ₛ {rv_to_devent p X t i})
```

In Theorem 3, `s` is a set of numbers that are used to represent the indices of the system components. `X` is a group of random variables that represent the time-to-failure of the switches in the system. We need to recall that `n_OR` accepts a list of random variables as an argument. Therefore, we create this list using `MAP_SET_LIST X s`, where set `s` is first converted to a list, then a list is created by mapping the random variable `X` over the list. In Theorem 3, `rv_to_devent` creates the DFT events of the random variables. It is defined as:

Definition 10 *rv_to_devent*

```
⊢ ∀ p X t. rv_to_devent p X t = (λi. DFT_event p (X i) t)
```

This way, we can use this function to create a group of DFT events for a set of indexed random variables. Then, we verify the probability of the n-ary OR gate in a way similar to the probability of the DRBD parallel structure in Tables 5, which is defined as the union of events. We formally verify this as:

Theorem 4

```
⊢ ∀ p X t s. FIN_NonEmpty s
  indep_sets p (λi. {rv_to_devent p X t i}) s ∧
  (∀ i. i ∈ s ⇒ rv_gt0_ninfinity [X i]) ⇒
  (prob p (DFT_event p (n_OR (MAP_SET_LIST X s)) t) =
   1 - ∏_{i∈s} (1 - F_{X_i}(t)))
```

In Theorem 4, it is required that the set of indices, `s`, is nonempty and finite, which is a realistic condition as in any system the number of components is finite. This is asserted using `FIN_NonEmpty s`. The last condition of Theorem 4 ensures that the values of the random variables of `X` are greater than or equal to 0 and not equal to $+\infty$, which is required to be able to use the CDF of the random variable as given in [10].

We express the structure function of the DFT of SEN as:

$$
\begin{aligned}
&Q_{\texttt{dSEN\_Terminal}} = \\
&\quad \texttt{n\_OR (MAP\_SET\_LIST (}\lambda\texttt{i. }\textbf{if}\ \texttt{i = 0}\ \textbf{then}\ \texttt{WSP Y Ys}_a\ \texttt{Ys}_d \\
&\qquad\qquad \textbf{else}\ \texttt{X i) (\{0\} } \cup \texttt{ L))}
\end{aligned}
\tag{2}
$$

We notice that the structure of the DFT is defined using the indices in `{0}` ∪ `L`. 0 is the index of the spare gate and `L` has the indices of the rest of the switches in the system.

Finally, we verify the probability of failure of this top event as:

Theorem 5

```
⊢ ∀p X Y Ys_a Ys_d t L.
 let
  WSPY = WSP Y Ys a Ys_d;
  F0= (λi. {event_set [(DFT_event p WSPY t, 0)]
        (rv_to_devent p X t) i});
  probl = (prob p (DFT_event p Q_dSEN_Terminal t);
  prob0= 1- prob p (DFT_event p WSPY t);
  probr = 1 - prob0 * ∏_{i∈L} (1-F_{X_i}(t))
 in
  DISJOINT {0} L ∧ FIN_NonEmpty L ∧
  indep_sets p F0({0} ∪ L) ∧
  (∀ i. i ∈ L ⇒ rv_gt0_ninfinity [X i]) ⇒
  probl = probr
```

In Theorem 5, `event_set` is a function that accepts a list of pairs in which each pair is composed of a DFT event with its index. In this case, `DFT_event p(WSP Y Ys`$_a$ `Ys`$_d$`) t` is the event and `0` is its index. Theorem 5 requires that (1) the indices of the elements in set `L` are unique and do not include `0`, which is characterized by `DISJOINT {0} L`; (2) the set `L`, which has the indices, is finite and not empty, which is ensured using `FIN_NonEmpty L`; and (3) the independence of the events, which is ascertained using `indep_sets`. Based on this theorem, we are able to verify the probability of the structure function of the terminal reliability of SEN. Details of the proof can be found at [8].

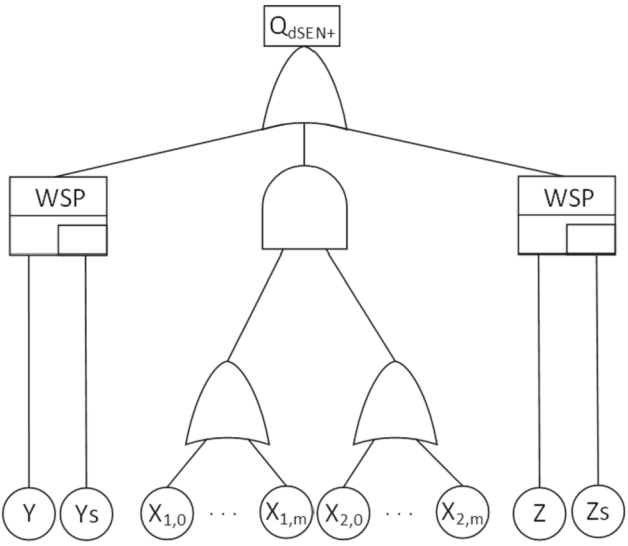


Fig. 11 DFT of SEN+ terminal connection.

Theorem 5 can be further rewritten based on the probability of the spare gate [10]. However, the required conditions of the latter should be satisfied, such as the continuity of the distributions. Since we need a group of indexed sets in `indep_sets`, we define a function `event_set` that accepts a list of pairs in which each pair is composed of a DFT event with its index. This function also accepts the remaining blocks of the DFT that have their indices embedded in a set (that can be generic of any size).

In SEN+, an additional path is added to increase the redundancy in the system. Therefore, for the connection between a given source and a destination to be broken, it is required that these two paths must be disconnected. The DFT of the SEN+ is shown in Fig. 11, where two spares are added to replace the main switches `Y` and `Z` after failure. Switch `Y` is the input switch connected to the source and switch Z is connected to the destination. This DFT is composed of three levels. Therefore, in order to verify the probability of the top event, we need first to verify that the `DFT_event` of the n-ary AND gate is equal to the intersection of the input events. We formally verify this in HOL as:

Theorem 6

```
⊢ ∀ p X t s. FIN_NonEmpty s ∧ 0 ≤ t ⇒
  (DFT_event p
   (n_AND (MAP_SET_LIST X s)) t =
    ⋂_{i∈s} {rv_to_devent p X t i})
```

Then, we verify the probability of failure of the AND gate top event as:

Theorem 7

```
⊢ ∀ p X t s. FIN_NonEmpty s ∧ 0 ≤ t ∧
  indep_sets p (λi. {rv_to_devent p X t i}) s
  (∀ i. i ∈ s ⇒ rv_gt0_ninfinity [X i]) ⇒
  (prob p
   (DFT_event p
    (n_AND (MAP_SET_LIST X s)) t) = ∏_{i∈s} (F_{X_i}(t)))
```

The first three conditions in Theorem 7 are needed to be able to use Theorem 6, while `indep_sets` ensures the independence of the events.

We use Theorems 4 and 7 to verify the probability of OR of AND of OR, which is required for the probability of the top event of SEN+. We express the structure function of the DFT of Fig. 11, Q_{dSEN+} as:

```
Q_dSEN+_Terminal =
 n_OR (MAP_SET_LIST (λi. if i = 0 then WSP Y Ys_a Ys_d
                         else if i = 1 then
                           ((n_OR (MAP_SET_LIST X L1)) ·
                            (n_OR (MAP_SET_LIST X L2)))
                         else  WSP Z Zs_a Zs_d)
       {0; 1; 2})
```
(3)

In the previous equation, {0; 1; 2} indicates that the OR gate has three inputs with indices 0 for the first spare, 1 for the AND of ORs, and 2 for the second spare. `L1` and `L2` contain the indices of the switches in the two redundant paths (for the two lower ORs).

The DFT top event can be expressed using union and intersection of events, which can be useful in reusing the existing theorems of probability of union of intersections and intersection of unions. We verify this relationship as:

Theorem 8

```
⊢ ∀ p Y Ys_a Ys_d Z Zs_a Zs_d X L1 L2 t.
 let
  WSPY = WSP Y Ys_a Ys_d;
  WSPZ = WSP Z Zs_a Zs_d;
  s_events_Y_Z = {event_set
     [(DFT_event p WSPY t,0);
      (DFT_event p WSPZ t,3)]
    (rv_to_devent p X t) i |
    i ∈ ind_set [{0}; L1; L2; {3}] a};
  BU0= {BIGUNION s_events_Y_Z | a | a ∈ ind_set
          [{0}; {1; 2}; {3}] j};
  BI0= {BIGINTER BU0| j | j ∈ {0; 1; 2}}
 in
  FINITE L1 ∧ FINITE L2 ∧
  disjoint_family_on (ind_set [{0}; L1; L2; {3}])
    {0; 1; 2; 3} ⇒
  (DFT_event p (Q_dSEN+_Terminal)t =
   BIGUNION BI0)
```

In Theorem 8, `disjoint_family_on (ind_set [{0}; L1; L2; {3}]) {0;1;2;3}` ensures that the sets {0}, `L1`, `L2` and {3} are disjoint, i.e., each switch has a unique index. We also define `ind_set`, which accepts a list of sets and returns a group of indexed sets. This is required to be able to create the hierarchy of the DFT using sets. More details can be found at [8].

Finally, we verify the probability of failure of Q_{dSEN+}:

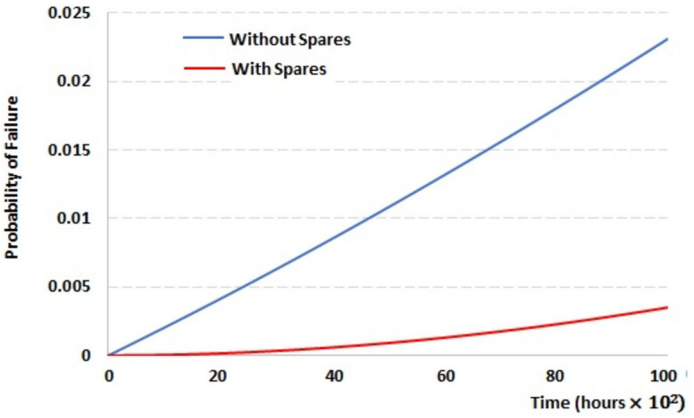


Fig. 12 Probability of failure of the terminal connection of a 128 × 128 SEN+ with and without spares.

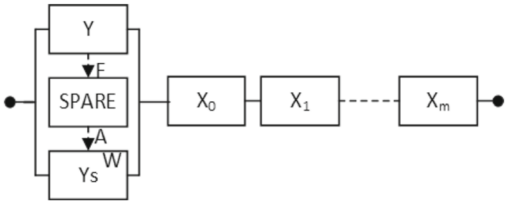
Theorem 9

```
⊢ ∀ p X Y Ys_a Ys_d Z Zs_a Zs_d t L1 L2.
 let
  WSPY = WSP Y Ys_a Ys_d;
  WSPZ = WSP Z Zs_a Zs_d;
  events_YZ = event_set [(DFT_event p WSPY t,0);
    (DFT_event p WSPZ t,3)]
   (rv_to_devent p X t);
  prob0= (1 - prob p (DFT_event p WSPY t))
  prob1 = (1 - ∏_{i∈L1} (1 - F_{X_i}(t))) * (1 - ∏_{i∈L2} (1 - F_{X_i}(t)))
  prob2 = (1 - prob p (DFT_event p WSPZ t))
  probl = prob p (DFT_event p Q_{dSEN+_Terminal t});
  probr = 1 - prob0  * (1 - prob1) * prob2 ;
 in
 0≤ t ∧
 SEN_set_req p L1 L2 (ind_set [{0}; L1; L2; {3}])
   (ind_set [{0}; {1; 2}; {3}]) {0; 1; 2} events_YZ ∧
 (∀ i. i ∈ (L1 ∪ L2) ⇒ rv_gt0_ninfinity [X i]) ⇒
 (probl = probr)
```

In the previous theorem, it is required to ensure that the sets are finite and nonempty. It is also required to ascertain the independence of the input events over the probability space. We use the function `SEN_set_req` to combine these conditions into an abstracted format.

In order to use the above verified generic probability of failure expressions on a concrete instance of SEN+, we take as an example the probability of failure of the terminal connection of a 128 × 128 SEN+, where each OR gate of the first level of Fig. 11 has 6 inputs. We assume that the failure rate of each switching element is 1×10^{-5}. We evaluate in MATLAB [22] the probability of failure for the SEN+ system with and without spare parts with a dormancy factor of 0.1, as shown in Fig. 12. This result shows that considering the spares in the analysis leads to having more reliable and realistic system than the traditional fault trees. Note that MATLAB is only used to calculate the probability of failure and not to implement the whole approach.

Fig. 13 DRBD of SEN.



4.2 DRBD analysis of SEN and SEN+

For SENs (single-path MIN), the terminal reliability is modeled as a series RBD. For illustration purposes, we use a spare part to replace the first input switch, and thus increase the reliability. The DRBD of the modified SEN is shown in Fig. 13, where Y is the main switch that will be replaced by Ys after failure and the series structure has $m + 1$ elements.

Using the proposed DRBD algebra in [11], we express the structure function of the SEN DRBD as:

$$\begin{aligned} &Q_{\texttt{SEN\_Terminal}} = \\ &\quad \texttt{nR\_AND}\ (\lambda \texttt{i}.\ \textbf{if}\ \texttt{i}\ =\ 0\ \textbf{then}\ \texttt{R\_WSP}\ \texttt{Y}\ \texttt{Ys}_\texttt{a}\ \texttt{Ys}_\texttt{d} \\ &\qquad\qquad \textbf{else}\ \texttt{X}\ \texttt{i})\ \{0\} \cup \texttt{L} \end{aligned} \tag{4}$$

In the previous equation, X is a group of indexed time-to-failure functions that represent the blocks of the series structure and L is a set with their indices. L can be instantiated with any group of numbers, which makes this function generic to represent the reliability model of any SEN with any size.

Then, we verify that the DRBD_event of $Q_{\texttt{SEN}}$ can be represented using the series structure as:

Theorem 10

```
⊢ ∀ p X Y Ys_a Ys_d t L.
 let
  R_WSPY = R_WSP Y Ys_a Ys_d;
  events_Y = (λi. event_set [(DRBD_event p R_WSPY t,0)]
   (rv_to_event p X t) i);
 in
  DISJOINT {0} L ∧ FIN_NonEmpty L ⇒
  (DRBD_event p Q_SEN_Terminal t =
   DRBD_series events_Y ({0} ∪ L) )
```

We use event_set and ind_set to create the events, similar to the DFTs. Since we are dealing with a series structure, we only need to specify the heirarchy of the architecture in one direction using $\{0\} \cup L$. We verify Theorem 10 using the relationship between nR_AND and DRBD_series (verified in [11]) and some set-related theorems.

Based on Theorem 10, we verify a generic expression for the reliability of the SEN system:

Theorem 11

```
⊢ ∀ p X Y Ys_a Ys_d t L.
 let
  R_WSPY = R_WSP Y Ys_a Ys_d;
```

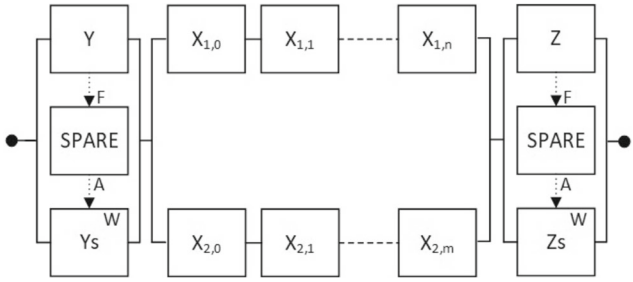


Fig. 14 Terminal reliability DRBD of SEN+.

```
  events_Y = (λi. {event_set [(DRBD_event p R_WSPY t, 0)]
   (rv_to_event p X t) i});
  indep_event_Y = indep_sets p events_Y ({0} ∪ L);
  probl = prob p (DRBD_event p Q_SEN_Terminal t);
  probr = Rel p (R_WSPY) t * ∏_{l∈L} (Rel p (X l) t);
 in
  DISJOINT {0} L ∧ FIN_NonEmpty L ∧ indep_event_Y ⇒
  (probl = probr)
```

In a similar manner, the SEN+ is modeled as a series–parallel-series structure. To further enhance the reliability, we use spare constructs as shown in Fig. 14, where Y and Z are the main single switches that are connected to the source and destination with their spares Ys and Zs, respectively. The parallel structure in the middle represents the reliability model of the two alternative paths between the source and the destination. Therefore, this DRBD consists of a series of two spare constructs and one parallel structure that consists of two series structures.

Using our DRBD operators, we formally express the structure function of this DRBD as:

$$
\begin{aligned}
&Q_{\text{SEN+\_Terminal}} = \\
&\quad \text{nR\_AND } (\lambda i.\ \textbf{if } i = 0 \textbf{ then } \text{R\_WSP Y } Ys_a\ Ys_d \\
&\qquad\qquad \textbf{else if } i = 1 \textbf{ then } \big((\text{nR\_AND X L1}) + (\text{nR\_AND X L2})\big) \\
&\qquad\qquad \textbf{else } \text{ R\_WSP Z } Zs_a\ Zs_d)\ \{0;\ 1;\ 2\}
\end{aligned}
\tag{5}
$$

Thus, the outer series structure is expressed using the nR_AND operator over the set {0; 1; 2} as this structure contains three different substructures; i.e., two spare constructs and one parallel structure. In order to re-utilize the verified expressions of reliability, it is required to express this DRBD using the series and parallel structures. Therefore, we verify that the DRBD event of the $Q_{\text{SEN+}}$ is equal to a nested series–parallel-series structure as:

Theorem 12

```
⊢ ∀ p X Y Ys_a Ys_d Z Zs_a Zs_d t L1 L2.
 let
  R_WSPY = R_WSP Y Ys_a Ys_d;
  R_WSPZ = R_WSP Z Zs_a Zs_d;
  events_YZ = (λi. event_set
    [(DRBD_event p R_WSPY t,0);
```

```
    (DRBD_event p R_WSPZ t,3)]
   (rv_to_event p X t) i);
  DRBD_s0= (λa. DRBD_series events_YZ (ind_set
           [{0}; L1; L2;{3}] a));
  DRBD_p0= (λj. DRBD_parallel DRBD_s0(ind_set
           [{0}; {1; 2};{3}] j));
  DRBD_s1 = DRBD_series DRBD_p0{0; 1; 2};
 in
  disjoint_family_on (ind_set [{0; 3}; L1; L2]) {0;1;2} ∧
  FIN_NonEmpty L1 ∧ FIN_NonEmpty L2 ⇒
  (DRBD_event p Q_SEN+_Terminal t = DRBD_s1)
```

In Theorem 12, `disjoint_family_on (ind_set [{0; 3}; L1; L2]) {0;1;2}` ensures that each switch has a unique index. Since we are dealing with a series–parallel-series structure, we need three sets to identify the hierarchy of this nested structure. Set {0; 1; 2} in Theorem 12 indicates that the outer series structure has three elements, i.e., three parallel structures. In addition, `ind_set [{0}; {1;2}; {3}]` indicates that the first parallel structure has only one series structure with index 0, the second parallel structure has two series structures with indices 1 and 2, and the third parallel structure has only one series structure with index 3. Finally, `ind_set [{0}; L1; L2; {3}]` implies that the first series structure has only one element with index 0, the second and third series structures have an arbitrary number of blocks indexed by `L1` and `L2`. The last series structure has one element with index 3. We verify Theorem 12 using the relationship between the event of `nR_AND` and the `DRBD_series`, and the equivalence of the event of OR with the union of events. Some basic set-related theorems were used in the proof as well. More details can be found at [8].

Based on Theorem 12, we verify a generic expression for the reliability of the SEN+ system:

Theorem 13

```
⊢ ∀ p X Y Ys_a Ys_d Z Zs_a Zs_d t L1 L2.
 let
  R_WSPY = R_WSP Y Ys_a Ys_d;
  R_WSPZ = R_WSP Z Zs_a Zs_d ;
  ind_set0= ind_set [{0}; L1; L2; {3}];
  ind_set1 = ind_set [{0}; {1; 2}; {3}];
  events_YZ = event_set [(DRBD_event p R_WSPY t,0);
    (DRBD_event p R_WSPZ t,3)]
   (rv_to_event p X t);
  probl = prob p (DRBD_event p Q_SEN_Terminal t ;
  prob0= 1 - ∏_{l∈L1} (Rel p (X l) t);
  prob1 = 1 - ∏_{l∈L2} (Rel p (X l) t);
  prob2 = 1 - prob0  * prob1;
  probr = Rel p R_WSPY t * Rel p R_WSPZ t * prob2;
 in
  SEN_set_req p L1 L2 ind_set0  ind_set1 {0; 1; 2} events_YZ ⇒
  (probl = probr)
```

In Theorem 13, `SEN_set_req` is the same function that we use with DFTs. We first rewrite the goal using Theorem 12, then we use the reliability of the series–parallel-series

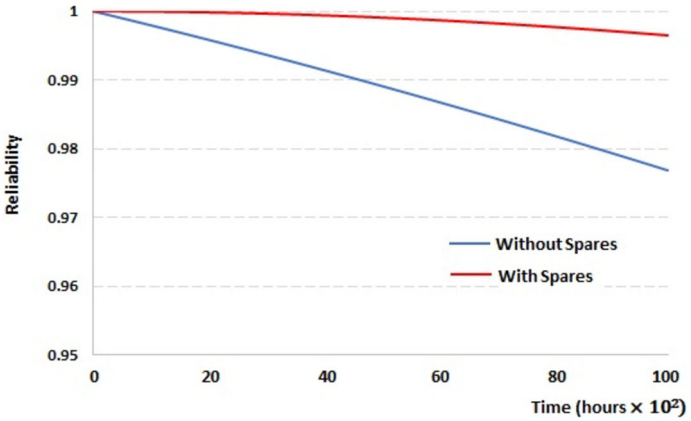


Fig. 15 Terminal reliability of 128 × 128 SEN+ with and without spares.

to verify the final expression. The reliability of the spare constructs can be further rewritten using the probability of the spare construct, verified in [11], given that the required conditions are ensured, such as the continuity of the CDFs. It can be noticed that the DRBD and the DFT models possess the same hierarchy represented by the sets of indices, which makes it easy to be used when going from one model to the other. More details about the proof steps can be found at [8].

Similar to the DFT analysis, we evaluate the terminal reliability of a 128 × 128 SEN+ in MATLAB, where each inner series structure of Fig. 14 has 6 blocks. We assume that the failure rate of each switching element is 1×10^{-5}. We evaluate the reliability for the SEN+ system with and without spare parts with a dormancy factor of 0.1, as shown in Fig. 15. Note that the results obtained here are the complement of the ones presented in Fig. 12 since reliability is the complement of failure.

4.3 Equivalence of DFT and DRBD models

In Sect. 3, we described how a DFT model can be formally analyzed using the DRBD algebra and vice versa. To illustrate the utilization of the proposed methodology, we formally verify the equivalence of the DRBD and the complement of the DFT events for terminal and broadcast reliability of SEN and SEN+. Proving this equivalence allows verifying the probability of one model and directly use the equivalence proof to provide the probability of the other model. In this section, we present the equivalence theorems of the terminal reliability and the remaining theorems for broadcast reliability will be presented in the following sections.

We verify the equivalence of the DRBD and DFT models of the terminal reliability of both SEN and SEN+. The main idea of the proof is to verify that the DFT event of SEN/SEN+ is the complement of the DRBD event of SEN/SEN+. Since these events are part of the probability space `p_space`, we need to use `p_space p DIFF` to express the complement of the DFT event in the probability space `p_space`.

Theorem 14 *Terminal/Broadcast SEN*

```
⊢ ∀ p X Y Ys_a Ys_d t L.
 let
  F0=nR_AND (λi. if i=0 then R_WSP Y Ys_a  Ys_d else  X i) {0} ∪ L;
  F1 = n_OR
       (MAP_SET_LIST (λi. if i = 0 then WSP Y Ys_a Ys_d else X i) ({0} ∪ L));
 in
  FINITE L ∧ (∀ s. ALL_DISTINCT [Y s; Ys_a s; Ys_d s])⇒
  (DRBD_event p F0  t = p_space p DIFF DFT_event p F1 t)
```

Theorem 15 *Terminal SEN+*

```
⊢ ∀ p X Y Ys_a Ys_d Z Zs_a Zs_d t L1 L2.
 let
  R_WSPY = R_WSP Y Ys_a Ys_d;
  R_WSPZ = R_WSP Z Zs_a Zs_d ;
  WSPY = WSP Y Ys_a Ys_d;
  WSPZ = WSP Z Zs_a Zs_d;
  F0= (nR_AND X L1) + (nR_AND X L2);
  F1 = nR_AND (λi. if i = 0then R_WSPY
                   else if i = 1 then F0
                   else R_WSPZ) {0; 1; 2};
  F2 = (n_OR (MAP_SET_LIST X L1)) · (n_OR (MAP_SET_LIST X L2));
  F3 =  n_OR (MAP_SET_LIST (λi. if i = 0then WSPY
                                else if i = 1 then F2
                                else WSPZ) {0; 1; 2});
 in
  FINITE L1 ∧ FINITE L2 ∧
  (∀s. ALL_DISTINCT [Y s; Ys_a s; Ys_d s; Z s; Zs_a s; Zs_d s])⇒
  (DRBD_event p F1 t = p_space p DIFF DFT_event p F3 t)
```

Based on these theorems, the terminal reliability analysis of SEN/SEN+ can be conducted using one model and the analysis of the other model can be performed based on this equivalence.

5 Broadcast reliability analysis of shuffle-exchange networks

The broadcast reliability represents the probability of having a working connection between one source and all destinations. This is required when one of the processors in the system needs to transmit information to all destinations in the network. We present in this section, the broadcast reliability of the SEN and SEN+ using both DFT and DRBD models.

5.1 DFT analysis of SEN and SEN+

Since in SENs there exists a single path between each source and destination, it is required to have a successful transmission through all these paths for a proper broadcast. Therefore, the DFT can be modeled using an OR gate. We further lower the probability of failure by adding an additional spare gate, as shown in Fig. 10. However, the number of DFT inputs, which represent the switches, varies between the terminal and broadcast reliability models.

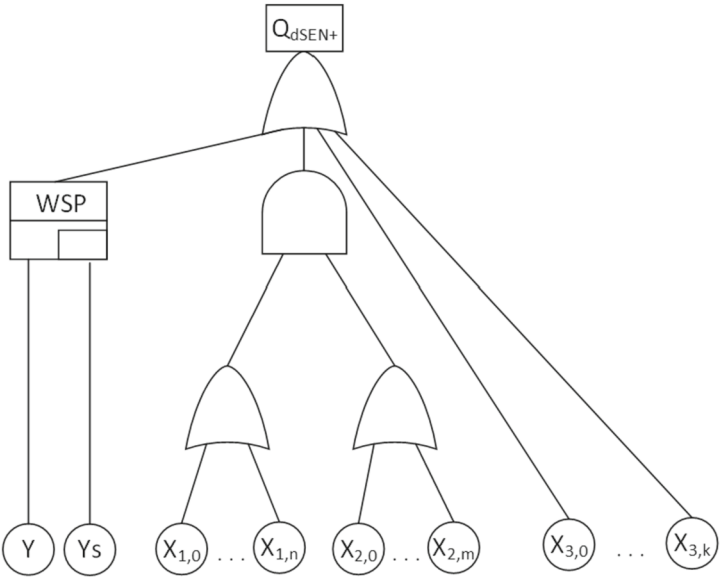


Fig. 16 DFT of broadcast SEN+.

For example, consider an 8×8 SEN. The number of inputs for the terminal DFT is 3, i.e., $log_2 8$, while the broadcast DFT requires seven inputs, i.e., $\sum_{i=1}^{log_2 8}(\frac{8}{2^i})$ [4]. Therefore, we can also use Theorem 5 for the broadcast, since this theorem is verified for any number of system blocks with their indices in the set s. This highlights the importance of having generic verified expressions for any number of system blocks, which enables the re-utilization of the theorems in different contexts.

The DFT model of the broadcast SEN+ is shown in Fig. 16. Its top event is modeled using an OR gate that is connected to a spare gate for the input switch, AND of OR to model the two alternative paths and finally, the rest of the destination switches in order to have a proper broadcast transmission.

We formally express the structure function of the top event as:

$$
\begin{aligned}
&Q_{\texttt{dSEN+\_Broadcast}} = \\
&\texttt{n\_OR (MAP\_SET\_LIST (}\lambda\texttt{i. }\textbf{if}\texttt{ i = 0 }\textbf{then}\texttt{ WSP Y Ys}_a\texttt{ Ys}_d \\
&\qquad\qquad \textbf{else if}\texttt{ i = 1 }\textbf{then} \\
&\qquad\qquad\quad \big(\texttt{(n\_OR (MAP\_SET\_LIST X L1)) } \cdot \\
&\qquad\qquad\quad\ \texttt{(n\_OR (MAP\_SET\_LIST X L2))}\big) \\
&\qquad\qquad \textbf{else}\texttt{ (n\_OR (MAP\_SET\_LIST X L3)))} \\
&\qquad\texttt{\{0; 1; 2\})}
\end{aligned}
\tag{6}
$$

The hierarchy of the DFT is divided using the sets of indices. `MAP X (SET_TO_LIST L1)`, `MAP X (SET_TO_LIST L2)` and `MAP X (SET_TO_LIST L3)` are used to create the lists of the group of random variables for the n-ary gates. $L1$ and $L2$ have the indices of the switches in the two alternative paths, i.e., the inputs of the two lower OR gates in the DFT of Fig. 16, while $L3$ has the indices of the remaining inputs of the top OR gate. The set $\{0; 1; 2\}$ indicates that the top OR gate has three inputs, which is similar to the terminal DFT model.

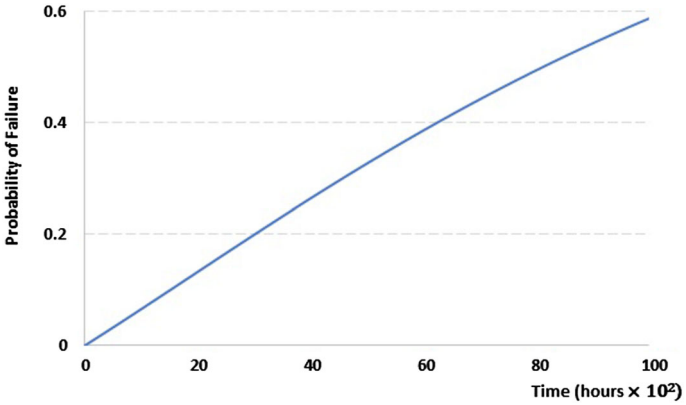


Fig. 17 Probability of failure of the broadcast of a 128 × 128 SEN+.

We use this structure function to verify the probability of failure of the top event utilizing the probability of the n-ary OR gate and the AND gate:

Theorem 16

```
⊢ ∀ p X Y Ys_a Ys_d t L1 L2 L3 s.
 let
  ind_set0= ind_set [{0}; L1; L2; L3];
  ind_set1 = ind_set [{0}; {1; 2}; {3}];
  WSPY = WSP Y Ys_a Ys_d ;
  DFT_WSPY = DFT_event p WSPY t;
  probl = prob p (DFT_event p Q_dSEN+_Broadcast  t);
  prob0= 1 - prob p DFT_WSPY;
  prob1 = 1 - ∏_{i∈L1} (1 - F_{X_i}(t));
  prob2 = 1 - ∏_{i∈L2} (1 - F_{X_i}(t));
  prob3 = 1 - prob1 * prob2;
  prob4 = ∏_{i∈L3} (1 - F_{X_i}(t));
  probr = 1 - prob0  * prob3 * prob4;
 in
  SEN_broad_set_req p L1 L2 L3 ind_set0  ind_set1 {0; 1; 2}
   (event_set [(DFT_WSPY,0)]
     (rv_to_devent p X t)) ∧ 0 ≤ t ∧
  (∀ i. i ∈ (L1 ∪ L2 ∪ L3) ⇒ rv_gt0_ninfinity [X i]) ⇒
  (probl = probr)
```

In Theorem 16, `SEN_broad_set_req` ascertains the conditions required for the sets, such as finiteness. It also ensures the independence of the events. More details about the proof can be found at [8].

Figure 17 shows the evaluation results of the probability of failure of the DFT of Fig. 16 for a 128 × 128 SEN+. This SEN+ has 63 inputs for each first level OR gate and the top level OR gate has 66 inputs. As with the terminal SEN+, we assume that the failure rate of each switching element is 1×10^{-5} with a dormancy factor of 0.1.

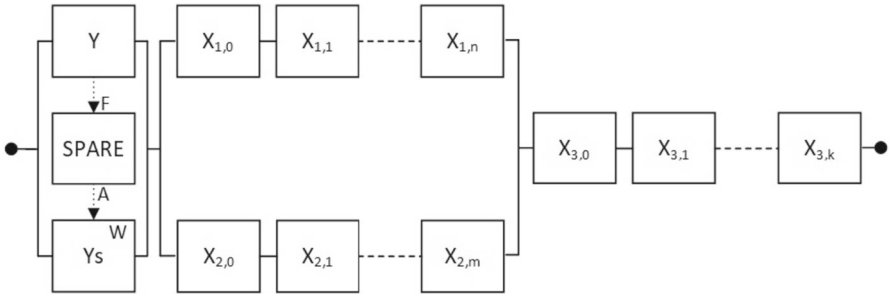


Fig. 18 Broadcast DRBD model of SEN+.

5.2 DRBD analysis of SEN and SEN+

Similar to the DFT SEN broadcast model, we can use the model in Fig. 13. However, as mentioned previously, the number of the blocks is different. Therefore, we can also use Theorem 11 for the broadcast reliability, since this theorem is verified for any number of system blocks using set s.

The DRBD of the SEN+ is depicted in Fig. 18. The first block (with the spare) represents the input switch that is connected directly to the source. The failure of this switch can interrupt the broadcast transmission. Therefore, we add a spare part to replace it after failure. The series structure on the right side of the figure models the switches of all destinations, as they are all receiving the transmission. Finally, the parallel-series structure in the middle, represents the two alternative paths that are available for each broadcast transmission. For example, for the SEN+ shown in Fig. 3, the number of switches connected to the destinations are four, while each one of the alternative paths has three switches.

In order to formally verify the reliability of the broadcast of the SEN+, we first express it using our operators as:

$$
\begin{aligned}
&Q_{\texttt{SEN+\_Broadcast}} = \\
&\texttt{nR\_AND}\ (\lambda \texttt{i}.\ \textbf{if}\ \ \texttt{i = 0}\ \textbf{then}\ \ \texttt{R\_WSP Y Ys}_{\texttt{a}}\ \texttt{Ys}_{\texttt{d}} \\
&\qquad\qquad \textbf{else if}\ \texttt{i = 1}\ \ \textbf{then}\ \big(\texttt{(nR\_AND X L1)} + \\
&\qquad\qquad\qquad\qquad\qquad \texttt{(nR\_AND X L2)}\big) \\
&\qquad\qquad \textbf{else}\ \texttt{(nR\_AND X L3))}\ \ \texttt{(\{0; 1; 2\})}
\end{aligned}
\tag{7}
$$

In the previous equation, $L1$ and $L2$ are the sets that have the indices of the inner series structures of the parallel-series structure in the middle. The set {0; 1; 2} indicates that the outer series structure consists of three main components. The first spare construct has index 0, while the parallel-series structure has index 1. Finally, the series structure on the left side of Fig. 18 has index 2, and $L3$ has the indices of the blocks in this series structure. We verify the reliability of this DRBD utilizing the probability of series and parallel structures as:

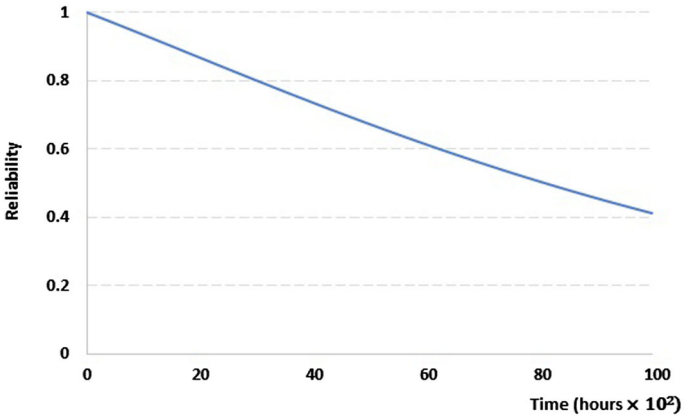


Fig. 19 Broadcast reliability of a 128 × 128 SEN+.

Theorem 17

$\vdash \forall$ p X Y Ys_a Ys_d t L1 L2 L3.
let
ind_set0= ind_set [{0}; L1; L2; L3];
ind_set1 = ind_set [{0}; {1; 2}; {3}];
R_WSPY = R_WSP Y Ys_a Ys_d;
probl = prob p (DRBD_event p $Q_{SEN+\_Broadcast}$ t);
prob0= Rel p R_WSPY t;
prob1 = $\prod_{i \in L3}$ (Rel p (X l) t);
prob2 = 1 - $\prod_{l \in L1}$ (Rel p (X l) t);
prob3 = 1 - $\prod_{l \in L2}$ (Rel p (X l) t);
prob4 = 1 - prob2 * prob3;
probr = prob0* prob1 * prob4;
in
SEN_broad_set_req p L1 L2 ind_set0 ind_set1 {0; 1; 2}
(event_set [(DRBD_event p R_WSPY t,0)]
(rv_to_event p X t)) ⇒
(probl = probr)

We evaluate the broadcast reliability, in Fig. 19, of a 128 × 128 SEN+, where each inner series structure of Fig. 18 has 63 blocks and the series structure on the right hand side of the figure has 64 blocks. We use the same failure rates of 1×10^{-5} for each switching element with a dormancy factor of 0.1.

5.3 Equivalence of DFT and DRBD models

Since the terminal and broadcast reliability models of the SEN are similar, Theorem 14 can be used for the equivalence of the SEN in the broadcast reliability of both models since they both share the same structure.

In a similar manner to the equivalence of the terminal reliability models, we verify the equivalence of the DRBD and DFT models of the SEN+ broadcast reliability. More precisely,

we verify that the DRBD event of the broadcast reliability of SEN+ is the complement of the DFT event.

Theorem 18 *Broadcast SEN+*

```
⊢ ∀ p X Y Ys_a Ys_d t L1 L2 L3.
 let
  R_WSPY = R_WSP Y Ys_a Ys_d;
  nR_AND_XL1 = nR_AND X L1;
  nR_AND_XL2 = nR_AND X L2;
  nR_AND_XL3 = nR_AND X L3;
  F0= nR_AND_XL1 + nR_AND_XL2;
  F1 = (λi. if i = 0 then R_WSPY else if i = 1
then F0 else nR_AND_XL3);
  F2 = nR_AND F1 {0; 1 2};
  WSPY = WSP Y Ys_a Ys_d;
  n_OR_XL1 = n_OR (MAP_SET_LIST X L1);
  n_OR_XL2 = n_OR (MAP_SET_LIST X L2);
  n_OR_XL3 = n_OR (MAP_SET_LIST X L3);
  F3 = n_OR_XL1 · n_OR_XL2;
  F4 = (λi. if i = 0 then WSPY else if i = 1
 then F3 else n_OR_XL3);
  F5 = n_OR (MAP_SET_LIST F4 ({0; 1 2}));
 in
  FINITE L1 ∧ FINITE L2 ∧ FINITE L3 ∧
  (∀ s. ALL_DISTINCT [Y s; Ys_a s; Ys_d s]) ⇒
  (DRBD_event p F2 t = p_space p DIFF DFT_event p F5 t)
```

6 Network reliability analysis of shuffle-exchange networks

According to [4], the network reliability of SENs can be defined as the reliability of all connections between sources (inputs) and destinations (outputs). In other words, we are looking at the reliability of the overall network. This is usually modeled using RBDs. In this section, we use both DFT and DRBD models in different scenarios to model the reliability of the network.

6.1 DFT analysis of SEN and SEN+

In the SEN, it is required that all switching elements must work properly in order to maintain a successful behavior of the network. Thus, the system fails with the failure of any of the switching elements. The behavior can be further enhanced by using spares. The DFT of the SEN network can be modeled as in Fig. 10. However, to further enhance the system reliability, the reliability engineer may suggest to use more spares to replace the switching elements. Therefore, we present a generic model, where the number of switching elements that have spares is generic, as shown in Fig. 20. This model can be also used with both the terminal and broadcast models, when more spares are required.

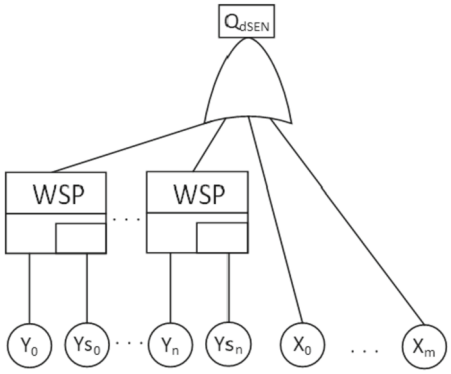


Fig. 20 DFT of SEN network with multiple spares.

The top event of the DFT of Fig. 20 can be expressed using the DFT operators as:

$$
\begin{aligned}
&\mathtt{Q_{dSEN\_Network}} = \\
&\quad \mathtt{n\_OR} \\
&\quad\quad (\mathtt{MAP\_SET\_LIST} \\
&\quad\quad\quad (\lambda \mathtt{i.\ \textbf{if}\ \ i} \in \mathtt{L1\ \ \textbf{then}\ WSP\ (Y\ i)\ (Ys_a i)\ (Ys_d\ i)} \\
&\quad\quad\quad\quad\quad \mathtt{\textbf{else}\ \ X\ i)\ (L1 \cup L2))}
\end{aligned}
\tag{8}
$$

We verify the probability of failure of the top event in a similar way to Theorem 5 as:

Theorem 19

$$
\begin{aligned}
&\vdash \forall\ \mathtt{p\ X\ Y\ Ys_a\ Ys_d\ t\ L1\ L2.} \\
&\ \mathtt{\textbf{let}} \\
&\quad \mathtt{F0 = (\lambda i.\ \ i} \in \mathtt{L1\ then\ WSP\ (Y\ i)\ (Ys_a\ i)\ (Ys_d\ i)\ \ else\ X\ i);} \\
&\quad \mathtt{F2\ =\ (\lambda i.\ \{rv\_to\_devent\ p\ F0\ \ t\ i\});} \\
&\quad \mathtt{probl\ =\ prob\ p\ (DFT\_event\ p\ Q_{dSEN\_Network}\ t);} \\
&\quad \mathtt{F3\ =\ } \textstyle\prod_{i \in L1} \mathtt{(1\ -\ prob\ p\ (DFT\_event\ p\ (WSP\ (Y\ i)\ (Ys_a\ i)\ (Ys_d\ i))t));} \\
&\quad \mathtt{F4\ =\ } \textstyle\prod_{i \in L2} \mathtt{(1\ -\ F_{X_i}\ (t));} \\
&\quad \mathtt{probr\ =\ 1\ -\ F3\ *\ F4;} \\
&\ \mathtt{\textbf{in}} \\
&\quad \mathtt{DISJOINT\ L1\ L2} \wedge \mathtt{FIN\_NonEmpty\ L1} \wedge \mathtt{FIN\_NonEmpty\ L2} \wedge \\
&\quad \mathtt{(\forall\ i.\ i} \in \mathtt{L2} \Rightarrow \mathtt{rv\_gt0\_ninfinity\ [X\ i])} \wedge \\
&\quad \mathtt{indep\_sets\ p\ F2\ (L1} \cup \mathtt{L2)} \Rightarrow \\
&\quad \mathtt{(probl\ =\ probr)}
\end{aligned}
$$

In Theorem 19, $\mathtt{Y}$, $\mathtt{Ys_a}$ and $\mathtt{Ys_d}$ are groups of indexed random variables that represent the main and spare switches. Theorem 19 provides a generic scenario for the SEN, where `L1` and `L2` can be instantiated with any number of distinct indices that represent the system switches, with and without spares.

The DFT model of the SEN+ network is shown in Fig. 21. It consists of a spare gate for one of the switches in the input stage. The rest of the input switches ($X_{1,0}$ to $X_{1,r}$) are connected directly to the n-OR gate of the top event. Therefore, the failure of any of these switches leads to the failure of the network. The series of ANDs and ANDs of ORs are used to model the two available paths. Finally, all destination switches ($X_{4,0}$ to $X_{4,k}$) are required to function and thus they are all connected to the output of the OR gate. This DFT is composed of three

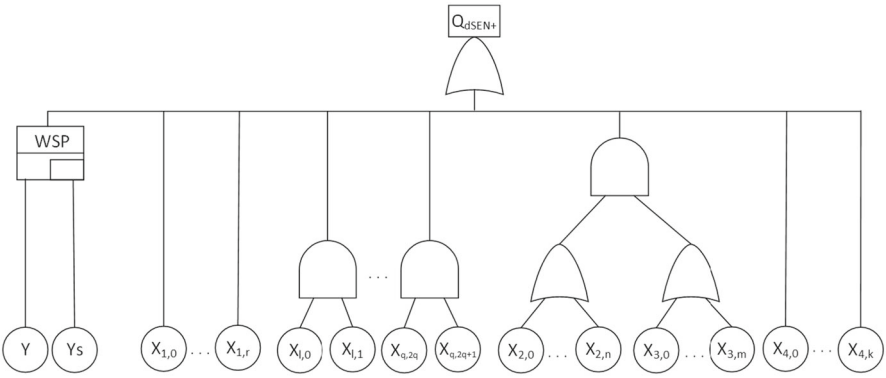


Fig. 21 DFT of SEN+ network.

levels; OR of ANDs of ORs, and thus we can use the theorems of union of intersections of unions to verify its probability of failure if the sets of indices are handled properly.

We first express the top event using the DFT operators as:

$$
\begin{aligned}
&Q_{\texttt{dSEN+\_Network}} = \\
&\texttt{n\_OR} \\
&\quad \texttt{(MAP\_SET\_LIST} \\
&\qquad \texttt{(}\lambda\texttt{i. }\textbf{if}\ \texttt{ i = 0 }\textbf{then}\ \texttt{ WSP Y Ys}_{\texttt{a}}\ \texttt{Ys}_{\texttt{d}} \\
&\qquad\qquad \textbf{else if}\ \texttt{i = 1 }\textbf{then}\ \texttt{n\_OR (MAP\_SET\_LIST X L1)} \\
&\qquad\qquad \textbf{else if}\ \texttt{i = 3 }\textbf{then}\ \texttt{(n\_OR (MAP\_SET\_LIST X L2))} \cdot \\
&\qquad\qquad\qquad\qquad\qquad \texttt{(n\_OR (MAP\_SET\_LIST X L3))} \\
&\qquad\qquad \textbf{else if}\ \texttt{i = 4 }\textbf{then}\ \texttt{n\_OR (MAP\_SET\_LIST X L4)} \\
&\qquad\qquad \textbf{else}\ \texttt{(X (2 * i))} \cdot \texttt{(X (2 * i + 1)))} \\
&\qquad \texttt{(\{0; 1; 3; 4\}} \cup \texttt{L))}
\end{aligned}
\tag{9}
$$

In the previous equation, the spare gate is assigned index 0. The second group of switches has index 1, while the indices of these switches, $X_{1,0}$ to $X_{1,r}$, are in set `L1`. They are represented as `n_OR (MAP_SET_LIST X L1`. The output of the AND of ORs is assigned index 3 and is modeled as `(n_OR (MAP_SET_LIST X L2))` $\cdot$ `(n_OR (MAP_SET_LIST X L3))`, which is similar to both the terminal and broadcast models. The group of switches, $X_{4,0}$ to $X_{4,k}$, has index 4 and is represented using `n_OR (MAP_SET_LIST X L4)`. Thus, we have the indices {0; 1; 3; 4} for the outer groups in the DFT. However, the last part of the DFT, which is the series of ANDs in the middle of Fig. 21, has a generic number of AND gates and cannot be assigned a specific index. Therefore, we use set `L` to get a unique index for the output of each AND gate. We use this unique number to create the indices of the inputs of each AND gate. For example, for an index `j` in set `L`, we create two indices for the inputs of the AND gate as `(2*j)` and `(2*j+1)`. This is modeled as `(X (2 * i))` $\cdot$ `(X (2 * i + 1)))` and set `L` is used with the set of indices in the outer level as `(SET_TO_LIST ({0; 1; 3; 4} ∪ L))`. It is important to highlight that the indices of the individual inputs should be unique.

We then verify that the `DFT_event` of $Q_{\texttt{dSEN\_Network}}$ is equal to the union of intersection of union of events as in the following theorem:

Theorem 20

```
⊢ ∀ p L1 L2 L3 L4 L X Y Ys_a Ys_d t.
 let
  s0= {2 * i | i ∈ L};
  s1 = {2 * i + 1 | i ∈ L};
  WSPY = WSP Y Ys_a Ys_d;
  s_events_WSP = {event_set [(DFT_event p (WSPY) t,0)]
                  (rv_to_devent p X t) i |
                    i ∈ if a ∈ s0 ∪ s1 then {a}
                         else ind_set [{0}; L1; L2; L3; L4] a}
  BU0= {BIGUNION s_events_WSP |
         a ∈ if j ∈ L then {2 * j; 2 * j + 1}
             else ind_set [{0}; {1}; {}; {2; 3}; {4}] j};
  BI0= {BIGINTER BU0| j ∈ {0; 1; 3; 4} ∪ L};
  BU1 = BIGUNION BI0;
 in
  FIN_NonEmpty L1 ∧ FIN_NonEmpty L2 ∧ FIN_NonEmpty L3 ∧
  FIN_NonEmpty L4 ∧ FINITE L ∧
  DISJOINT {0; 1; 3; 4} L ∧
  (∀ i. i ∈ L ⇒ DISJOINT {2 * i; 2 * i + 1} {0; 1; 2; 3; 4}) ∧
  disjoint_family_on
    (ind_set [{0}; L1; L2; L3; L4; s0 ∪ s1]) {0; 1; 2; 3; 4; 5} ⇒
  (DFT_event p (Q_dSEN_Network) t = BU1)
```

In Theorem 20, the conditions are required to ensure that the sets are finite, nonempty and that at each level of the DFT the indices are unique. It is clear from the theorem how the hierarchy of the DFT is structured using the sets. For example, "`if j ∈ L then {2 * j; 2 * j + 1} else ind_set [{0}; {1}; {}; {2; 3}; {4}] j`" determines the indices of the second level of the DFT (the ORs) based on the value of `j` in the outer level. The first part "`if... then`" is for the series of ANDs, while the "`else`" is for the rest of the parts in the second level. Although some of the parts of the DFT have no intermediate OR gates, like the spare, we implicitly assume that there are OR gates with single inputs to maintain the consistency. The indices of the second level indicates the indices of the output of these gates. This can be obvious for the AND of ORs in Fig. 21, where the OR gates have indices 2 and 3. We use an empty set ({}) in the indices of the second level due to the fact that there is no index 2 in the outer level, and thus we assign an empty set in the second level for this index.

We verify the probability of failure of $Q_{dSEN\_Network}$ as:

Theorem 21

```
⊢ ∀ p L1 L2 L3 L4 L X Y Ys_a Ys_d t.
 let
  s0= {2 * i | i ∈ L};
  s1 = {2 * i + 1 | i ∈ L};
  s2 = (λi. if i ∈ s0 ∪ s1 then {i}
            else ind_set [{0}; L1; L2; L3; L4] i);
  s3 = (λj. if j ∈ L then {2 * j; 2 * j + 1}
            else ind_set [{0}; {1}; {}; {2; 3}; {4}] j);
  WSPY = WSP Y Ys_a Ys_d;
  events_WSPY = event_set [(DFT_event p (WSPY) t,0)];
  probl = prob p (DFT_event p (Q_dSEN_Network) t);
```

```
  prob0= 1 - prob p (DFT_event p (WSPY) t);
  prob1 = ∏_{l∈L1} (1 - F_{X_1}(t));
  prob2 = 1 - ∏_{l∈L2} (1 - F_{X_1}(t));
  prob3 = 1 - ∏_{l∈L3} (1 - F_{X_1}(t));
  prob4 = 1 - prob2 * prob3;
  prob5 = ∏_{l∈L4} (1 - F_{X_1}(t));
  prob6 =∏_{j∈L} (1 - F_{X_{2*j}}(t) * F_{X_{2*j+1}}(t));
  probr = 1 - prob0* prob1 * prob4 * prob5 * prob6;
 in
  SEN_network_set_req p L1 L2 L3 L4 L
    s2 s3 ({0; 1; 3; 4} ∪ L)
     (events_WSPY (rv_to_devent p X t)) ∧
  (∀ i. i ∈ L1 ∪ L2 ∪ L3 ∪ L4 ∪ s0 ∪ s1 ⇒
      rv_gt0_ninfinity
[X i]) ⇒   (probl = probr)
```

In Theorem 21, `SEN_network_set_req` ensures that all sets are finite, nonempty and distinct. It also ensures the independence of the input events. It accepts all sets of the indices of the three levels. The second condition (`rv_gt0_ninfinity [X i]`) ascertains that each element in the group of random variables of `X` that have their indices in `L1 ∪ L2 ∪ L3 ∪ L4 ∪ {2 * i | i ∈ L} ∪ {2 * i + 1 | i ∈ L}` are greater than or equal to 0 but not equal $+\infty$. This condition is required to be able to use the CDF of the random variables.

In a similar manner to the SEN network, we provide a generic model where any number of spares can be used for the input switches. The modified DFT is shown in Fig. 22. We express the top event using the DFT operators as:

```
Q_dSEN_Network2 =
 n_OR
  (MAP_SET_LIST
    (λi. if i = 0 then WSP (Y 0) (Ys_a 0) (Ys_d 0)
          else if i = 1 then
           (n_OR (MAP_SET_LIST X L1))
         else if i = 3 then (n_OR (MAP_SET_LIST X L2)) ·
                            (n_OR (MAP_SET_LIST X L3))
         else if i = 4 then n_OR (MAP_SET_LIST X L4)
         else (X (2 * i)) · (X (2 * i + 1)))
    ({0; 1; 3; 4} UNION L))
```

(10)

In the previous equation, `Y`, $\mathtt{Ys}_a$ and $\mathtt{Ys}_d$ are indexed random variables that represent the main and spare parts for each spare gate. We choose to use the same hierarchy of Fig. 21, where we assign index 0 for the first spare and the rest of the spares have their indices in set `L1`. In addition, the model of these additional spares is embedded within `X` as will be explained shortly.

We verify the probability of failure of the top event as:

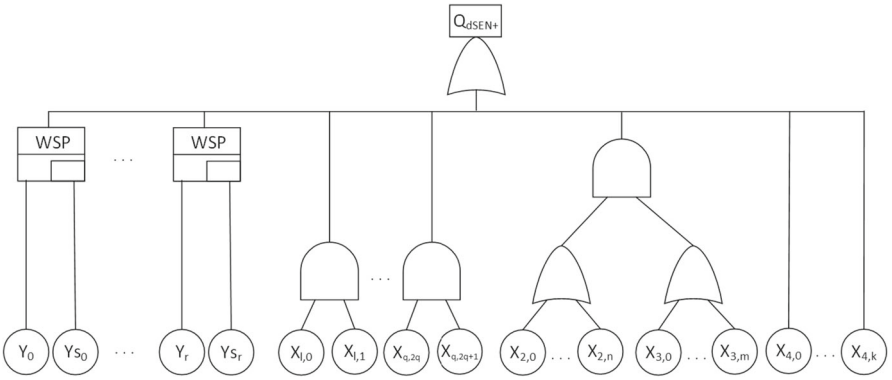


Fig. 22 DFT of SEN+ with multiple spares.

Theorem 22

```
⊢ ∀ p L1 L2 L3 L4 L X Y Ys_a Ys_d t.
 let
  s0= {2 * i | i ∈ L};
  s1 = {2 * i + 1 | i ∈ L};
  s2 = (λi. if i ∈ s0 ∪ s1 then {i}
            else ind_set [{0}; L1; L2; L3; L4] i);
  s3 = (λj. if j ∈ L then {2 * j; 2 * j + 1}
            else ind_set [{0}; {1}; {}; {2; 3}; {4}] j);
  events_WSP0= event_set [(DFT_event p (WSP (Y 0) (Ys_a 0)
  (Ys_d 0)) t,0)];
  probl = prob p (DFT_event p (Q_dSEN_Network2) t) ;
  prob0= ∏_{l∈({0}∪L1}
              (1 - prob p (DFT_event p (WSP (Y l) (Ys_a l) (Ys_d l))t);
  prob1 = 1 - ∏_{l∈L2} (1 - F_{X_l}(t));
  prob2 = 1 - ∏_{l∈L3} (1 - F_{X_l}(t)));
  prob3 =  ∏_{l∈L4} (1 - F_{X_l}(t));
  prob4 = ∏_{j∈L} (1 - F_{X_{2*j}}(t) * F_{X_{2*j+1}}(t));
  prob5 = 1 - prob1 * prob2;
  probr = 1 - prob0* prob5 * prob3 * prob4;
 in
  SEN_network_set_req p L1 L2 L3 L4 L
    s2 s3 ({0; 1; 3; 4} ∪ L)
    (λi. events_WSP0  (rv_to_devent p X t) i) ∧
  (∀ i. i ∈ L1 ∪ L2 ∪ L3 ∪ L4 ∪ s0∪ s1 ⇒ rv_gt0_ninfinity
  [X i]) ∧
  (∀ i. i ∈ L1 ⇒ (X i = WSP (Y i) (Ys_a i) (Ys_d i)) ⇒
  (probl = probr)
```

In Theorem 22, the conditions are similar to Theorem 21. However, we add the condition that (∀ i. i ∈ L1 ⇒ (X i = WSP (Y i) (Ys$_a$ i)(Ys$_d$ i)), which adds the additional spare gates. This way, we can use Theorem 21 to verify Theorem 22. Set {0} ∪ L1 is used to provide the indices of the spares, including the first one with index 0.

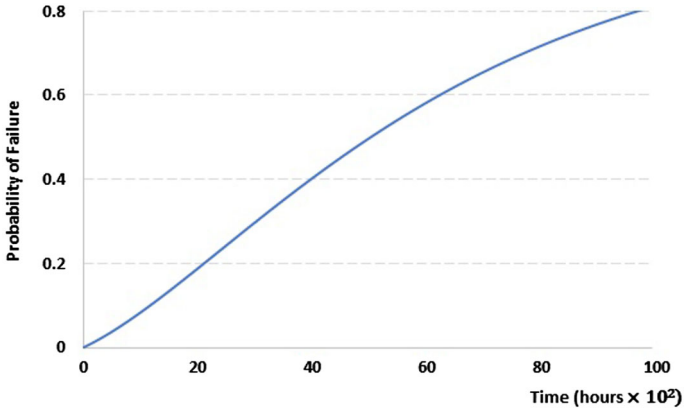


Fig. 23 The probability of failure of the network of a 128 × 128 SEN+.

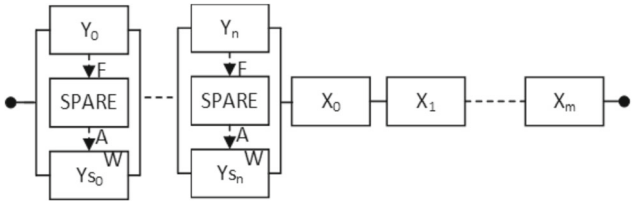


Fig. 24 DRBD of SEN network.

We evaluate the probability of failure of the network DFT, shown in Fig. 22, for a 128×128 SEN+. The DFT of this SEN has 32 AND gates in the first level. Each OR gate in the first level has 160 inputs. Furthermore, we assume that all 64 input switches have spares. Figure 23 shows the evaluated result of the probability of failure, where the failure rates of each switching element is 1×10^{-5} with a dormancy factor of 0.1.

6.2 DRBD analysis of SEN and SEN+

Similar to the DFT models, we start first with the network reliability model of the SEN. Since it is a single path, it can be modeled using the series DRBD of Fig. 13. Thus, we can use Theorem 11 to provide a generic expression for its reliability. We provide a generic model in Fig. 24, where additional spares are used. This provides a general case where we can choose how many switches can be replaced with spares.

We express the structure function of this DRBD using DRBD operators as:

$$\begin{aligned} &Q_{\texttt{SEN\_Network}} = \\ &\texttt{nR\_AND}\ (\lambda \texttt{i}.\ \textbf{if}\ \texttt{i} \in \texttt{L1}\ \textbf{then}\ \texttt{R\_WSP}\ (\texttt{Y i})\ (\texttt{Ys}_{\texttt{a}}\ \texttt{i})\ (\texttt{Ys}_{\texttt{d}}\ \texttt{i}) \\ &\qquad\qquad \textbf{else}\ \texttt{X i})\ (\texttt{L1} \cup \texttt{L2}) \end{aligned} \tag{11}$$

In the previous equation, L1 and L2 provide the indices of the blocks in the series structure for the spare constructs and the remaining blocks, respectively.

Similar to the proof steps of Theorem 13, we verify the reliability of the SEN network as:

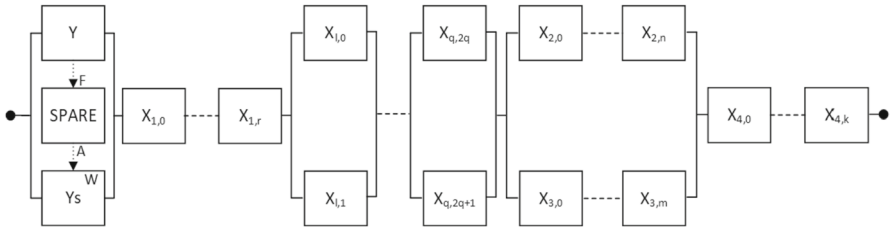


Fig. 25 DRBD of SEN+ newtork.

Theorem 23

```
⊢ ∀ p X Y Ys_a Ys_d t L1 L2.
 let
  probl = prob p (DRBD_event p Q_SEN_Network t);
  prob0 = ∏_{i∈L1} (Rel p (R_WSP (Y i) (Ys_a i) (Ys_d i)) t) ;
  prob1 = ∏_{i∈L2} (Rel p (X i) t);
  probr = prob0* prob1;
 in
  DISJOINT L1 L2 ∧ FIN_NonEmpty L1 ∧ FIN_NonEmpty L2 ∧
  indep_sets p
   (λi.
    {if i ∈ L1 then DRBD_event p (R_WSP (Y i) (Ys_a i) (Ys_d i)) t
     else (rv_to_event p X t) i}) (L1 ∪ L2) ⇒
  (probl = probr)
```

The DRBD of the SEN+ network is modeled in Fig. 25, where only one of the switches of the input stage can be replaced by a spare. This DRBD is composed of a series–parallel-series structure. The indices of each level can be treated in a similar manner to the DFT.

We express the structure function using the operators with the same sets of indices of the DFT as:

```
Q_SEN+_Network =
 nR_AND
    (λi.
      if i= 0 then R_WSP Y Ys_a Ys_d
     else if i = 1 then nR_AND X L1                          (12)
     else if i = 3 then (nR_AND X L2) + (nR_AND X L3)
     else if i = 4 then nR_AND X L4
     else (X (2 * i)) + (X (2 * i + 1)))
    ({0; 1; 3; 4} ∪ L))
```

Then, we verify that the `DRBD_event` of this structure can be expressed as a series–parallel-series structure as:

Theorem 24

```
⊢ ∀ p L1 L2 L3 L4 L X Y Ys_a Ys_d t.
 let
  s0= {2 * i | i ∈ L};
```

```
  s1 = {2 * i + 1 | i ∈ L};
  R_WSPY = R_WSP Y Ys_a Ys_d;
  events_Y = (λi. event_set [(DRBD_event p R_WSPY t,0)]
                (rv_to_event p X t) i);
  F0= (λj.
            if j ∈ L then {2 * j; 2 * j + 1}
           else ind_set [{0}; {1}; {}; {2; 3}; {4}] j);
  DRBD_s0= (λa. DRBD_series events_Y
              ((λi. if i ∈ s0 ∪ s1 then {i}
                    else ind_set [{0}; L1; L2; L3; L4] i) a));
  DRBD_p0= (λj. DRBD_parallel DRBD_s0(F0  j));
  DRBD_s1 = DRBD_series DRBD_p0({0; 1; 3; 4} ∪ L);
 in
  FIN_NonEmpty L1 ∧ FIN_NonEmpty L2 ∧ FIN_NonEmpty L3 ∧
  FIN_NonEmpty L4 ∧ FINITE L ∧
  DISJOINT {0; 1; 3; 4} L ∧
  (∀ i. i ∈ L ⇒ DISJOINT {2 * i; 2 * i + 1} {0; 1; 2; 3; 4}) ∧
  disjoint_family_on
   (ind_set [{0}; L1; L2; L3; L4; s0 ∪ s1]) {0; 1; 2; 3; 4; 5} ⇒
  (DRBD_event p (Q_SEN_Network) t = DRBD_s1)
```

Details about the steps of the proof can be found at [8].

Finally, we verify the reliability of the DRBD using the previous Theorem and the probability of parallel and series structures as:

Theorem 25

```
⊢ ∀  p L1 L2 L3 L4 L X Y Ys_a Ys_d t.
 let
  s0= {2 * i | i ∈ L};
  s1 = {2 * i + 1 | i ∈ L};
  s2 = (λi. if i ∈ s0∪ s1 then {i}
            else ind_set [{0}; L1; L2; L3; L4] i);
  s3 = (λj. if j ∈ L then {2 * j; 2 * j + 1}
            else ind_set [{0}; {1}; {}; {2; 3}; {4}] j);
  R_WSPY = R_WSP Y Ys_a Ys_d;
  probl = prob p (DRBD_event p (Q_SEN_Network)  t);
  prob0= Rel p R_WSPY t;
  prob1 = ∏_{l∈L1} (Rel p (X l) t);
  prob2 = 1 - ∏_{l∈L2} (Rel p (X l) t);
  prob3 = 1 - ∏_{l∈L3} (Rel p (X l) t);
  prob4 = 1 - prob2 * prob3;
  prob5 = ∏_{l∈L4}(Rel p (X l) t);
  prob6 = 1 - Rel p (X (2 * j)) t;
  prob7 = 1 - Rel p (X (2 * j + 1)) t;
  prob8 = ∏_{j∈L} (1 - prob6 * prob7);
  probr = prob0* prob1 * prob4 * prob5 * prob8;
 in
  SEN_network_set_req p L1 L2 L3 L4 L s2 s3 ({0; 1; 3; 4} ∪ L)
    (event_set [(DRBD_event p (R_WSP Y Ys_a Ys_d) t, 0)]
```

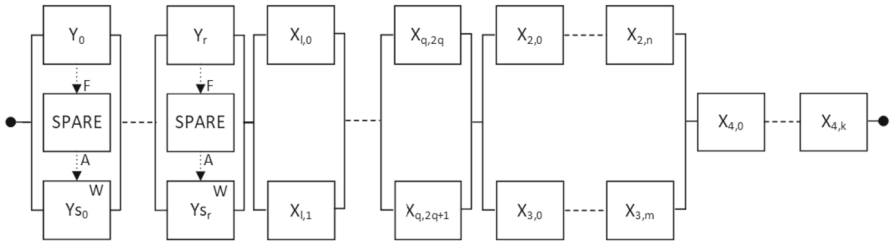


Fig. 26 DRBD of SEN+ network with multiple spares.

```
      (rv_to_event p X t)) ⇒
(probl = probr)
```

It is worth mentioning that the conditions of the sets are similar to those in Theorem 21 of the DFT.

Finally, we provide a generic model to have any number of spares that can replace the input switches, as shown in Fig. 26. We choose to use the same indices of Fig. 25 in order to reutilize the verified theorems.

We express the structure of the DRBD of Fig. 26 as:

```
Q_SEN_Network2 =
 nR_AND
   (λi.
     if  i = 0 then  R_WSP (Y 0) (Ys_a 0) (Ys_d 0)
     else if i = 1 then nR_AND X L1                          (13)
     else if i = 3 then (nR_AND X L2) + (nR_AND X L3)
     else if i = 4 then nR_AND X L4
     else (X (2 * i)) + (X (2 * i + 1)))
   ({0; 1; 3; 4} ∪ L))
```

In the previous equation, `Y`, $\texttt{Ys}_\texttt{a}$ and $\texttt{Ys}_\texttt{d}$ are indexed groups of random variables that represent the main parts and their spares.

Finally, we use Theorem 25 to verify the reliability of this DRBD as:

Theorem 26

```
⊢ ∀ p L1 L2 L3 L4 L X Y Ys_a Ys_d t.
 let
  s0= {2 * i | i ∈ L};
  s1 = {2 * i + 1 | i ∈ L};
  s2 = (λi. if i ∈ s0 ∪ then {i}
            else ind_set [{0}; L1; L2; L3; L4] i);
  s3 = (λj. if j ∈ L then {2 * j; 2 * j + 1}
            else ind_set [{0}; {1}; {}; {2; 3}; {4}] j);
  events_R_WSPY0= event_set
                [(DRBD_event p (R_WSP (Y 0) (Ys_a 0) (Ys_d 0)) t, 0)];
  probl = prob p (DRBD_event p (Q_SEN_Network2) t);
  prob0=∏_{l∈({0}∪L1} (Rel p (R_WSP (Y l) (Ys_a l) (Ys_d l)) t);
```

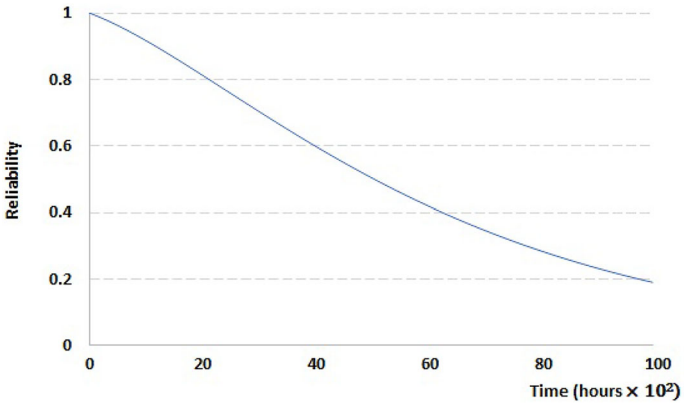


Fig. 27 The network reliability of a 128 × 128 SEN+.

```
  prob1 = 1 - ∏_{l∈L2} (Rel p (X l) t);
  prob2 = 1 - ∏_{l∈L3} (Rel p (X l) t);
  prob3 = 1 - prob1 * prob2;
  prob4 = ∏_{l∈L4}(Rel p (X l) t);
  prob5 = ∏_{j∈L} (1 -
                  (1 - Rel p (X (2 * j)) t) *
                  (1 - Rel p (X (2 * j + 1)) t));
  probr = prob0* prob3 * prob4 * prob5;
in
  SEN_network_set_req p L1 L2 L3 L4 L s2 s3
     ({0; 1; 3; 4} ∪ L)
    (events_R_WSPY0(rv_to_event p X t)) ∧
  (∀ i. i ∈ L1 ⇒ (X i = R_WSP (Y i)(Ysa i) (Ysd i))) ⇒
  (probl = probr)
```

As an application, we evaluate the network reliability of a 128 × 128 SEN+, as shown in Fig. 27, where there are 32 parallel structures that are connected in series. The DRBD has 64 spare constructs, while there are 160 blocks in the inner series structures. Finally, the series structure on the right hand side of Fig. 26 has 64 blocks. We assume that the failure rates of each switching element is 1×10^{-5} with a dormancy factor of 0.1. It is worth mentioning that the equivalence of the DFT and DRBD models of the network reliability can be verified in a similar manner to the terminal and broadcast reliability. The proof script of the verification of SEN and SEN+, which is available at [8], is around 9600 lines of code and it took around 80 h to be developed.

7 Conclusions

In this paper, we presented the formal dynamic dependability analysis of SEN and SEN+ MINs that form a critical part in the routing process of multiprocessor systems. Based on our proposed framework for formal dynamic dependability analysis using DFTs and DRBDs, we provided generic expressions of reliability and probability of failure of SEN/SEN+ that are independent of the failure distributions. Furthermore, we verified these expressions for an

arbitrary number of system blocks that can be instantiated later to a certain number without the need to repeat the verification process. For instance, we evaluated the reliability and probability of failure using MATLAB for a specific number of system components based on these generic expressions. It is worth mentioning that such sound generic results cannot be obtained using simulation or model checking as the state space should be defined in advance. In order to facilitate the dynamic dependability analysis in HOL4, a future work direction would consider automating this process using machine learning techniques. For instance, the TacticToe approach implemented in [15] can be used to automate the selection of the proper tactics to prove a goal in HOL4. This will allow end-users that are unfamiliar with theorem proving to benefit from our DFT and DRBD formalization to provide sound analysis of complex engineering systems.

Data availability The theories developed during the current study are available at http://hvg.ece.concordia.ca/code/hol/SEN/index.php.

References

1. Aggarwal R, Kaur L (2008) On reliability analysis of fault-tolerant multistage interconnection networks. Int J Comput Sci Secur 2(4):01–08
2. Ahmed W, Hasan O (2015) Towards formal fault tree analysis using theorem proving. In: Intelligent computer mathematics, LNCS 9150. Springer, pp 39–54
3. Avizienis A, Laprie J, Randell B, Landwehr C (2004) Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Dependable Secure Comput 1(1):11–33
4. Bistouni F, Jahanshahi M (2014) Analyzing the reliability of shuffle-exchange networks using reliability block diagrams. Reliab Eng Syst Saf 132:97–106
5. Bistouni F, Jahanshahi M (2019) Determining the reliability importance of switching elements in the shuffle-exchange networks. Int J Parallel Emergent Distrib Syst 34(4):448–476
6. Distefano S, Puliafito A (2007) Dynamic reliability block diagrams vs dynamic fault trees. In: reliability and maintainability symposium. IEEE, pp 71–76
7. Distefano S, Xing L (2006) A new approach to modeling the system reliability: dynamic reliability block diagrams. In: Reliability and maintainability symposium. IEEE, pp 189–195
8. Elderhalli Y (2019) Shuffle-exchange network formal dependability analysis: HOL4 script. Concordia University, Canada. http://hvg.ece.concordia.ca/code/hol/SEN/index.php
9. Elderhalli Y, Ahmad W, Hasan O, Tahar S (2018) Formal probabilistic analysis of dynamic fault trees in HOL4. Tech. rep., Concordia University, Canada. https://arxiv.org/abs/1807.11576
10. Elderhalli Y, Ahmad W, Hasan O, Tahar S (2019) Probabilistic analysis of dynamic fault trees using HOL theorem proving. J Appl Log 2631(3):469
11. Elderhalli Y, Hasan O, Tahar S A (2019) Formally verified algebraic approach for dynamic reliability block diagrams. In: Formal engineering methods, LNCS 11852. Springer, pp 253–269
12. Elderhalli Y, Hasan O, Tahar S (2019) A formally verified HOL algebra for dynamic reliability block diagrams. Technical report, Concordia University, Canada. http://arxiv.org/abs/1908.01930
13. Elderhalli Y, Hasan O, Tahar S (2019) A methodology for the formal verification of dynamic fault trees using HOL theorem proving. IEEE Access 7:136176–136192
14. Elderhalli Y, Volks M, Hasan O, Katoen J, Tahar S (2019) Formal verification of rewriting rules for dynamic fault trees. In: Software engineering and formal methods, LNCS 11724. Springer, pp 513–531
15. Gauthier T, Kaliszyk C, Urban J (2017) TacticToe: learning to reason with HOL4 tactics. In: Logic for programming, artificial intelligence and reasoning, vol 46, pp 125–143
16. Gunawan I (2008) Redundant paths and reliability bounds in gamma networks. Appl Math Model 32(4):588–594
17. Gunawan I (2013) Reliability prediction of distributed systems using Monte Carlo method. Int J Reliab Saf 7(3):235–248
18. Hasan O, Ahmed W, Tahar S, Hamdi MS (2015) Reliability block diagrams based analysis: a survey. In: International conference of numerical analysis and applied maths, vol 1648, p 850129. AIP
19. Hennessy J, Patterson D (2011) Computer architecture: a quantitative approach. Elsevier, Amsterdam

20. Jeng M, Siegel H (1986) A fault-tolerant multistage interconnection network for multiprocessor systems using dynamic redundancy. In: International conference on distributed computing systems. IEEE, pp 70–77
21. Kumar V, Reddy S (1988) Fault-tolerant multistage interconnection networks for multiprocessor systems. In: Concurrent computations. Springer, pp 495–523
22. MATLAB (2017) 2017a, The MathWorks, Natick
23. Merle G (2010) Algebraic modelling of dynamic fault trees, contribution to qualitative and quantitative analysis. Ph.D. thesis, ENS, France
24. Mhamdi T (2012) Information-theoretic analysis using theorem proving. Ph.D. thesis, Concordia University, Montreal, QC, Canada
25. Mhamdi T, Hasan O, Tahar S (2010) On the formalization of the Lebesgue integration theory in HOL. In: Interactive theorem proving, LNCS 6172. Springer, pp 387–402
26. Mhamdi T, Hasan O, Tahar S (2011) Formalization of entropy measures in HOL. In: Interactive theorem proving, LNCS 6898. Springer, pp 233–248
27. Nipkow T, Wenzel M, Paulson LC (2002) Isabelle/HOL: a proof assistant for higher-order logic. Springer, Berlin
28. Panda D, Dash R, Mishra A, Mohapatra S (2018) Reliability evaluation and analysis of multistage interconnection networks. Int J Pure Appl Math 119(14):1729–1737
29. Qasim M, Hasan O, Elleuch M, Tahar S (2016) Formalization of normal random variables in HOL. In: Intelligent computer mathematics, LNCS 9791. Springer, pp 44–59
30. Rajkumar S, Goyal N (2016) Review of multistage interconnection networks reliability and fault-tolerance. IETE Tech Rev 33(3):223–230
31. Ruijters E, Stoelinga M (2015) Fault tree analysis: a survey of the state-of-the-art in modeling, analysis and tools. Comput Sci Rev 15–16:29–62
32. Stamatelatos M, Vesely W, Dugan J, Fragola J, Minarick J, Railsback J (2002) Fault tree handbook with aerospace applications. NASA Office of Safety and Mission Assurance
33. Yunus N, Othman M (2015) Reliability evaluation for shuffle exchange interconnection network. Procedia Comput Sci 59:162–170
34. Yunus N, Othman M, Hanapi Z, Kweh Y (2019) Evaluation of replication method in shuffle-exchange network reliability performance. In: Advances in data and information sciences. Springer, pp 271–281
35. Yunus N, Othman M, Hanapi Z, Lun K (2016) Reliability review of interconnection networks. IETE Tech Rev 33(6):596–606