https://doi.org/10.1007/s00165-019-00503-1
BCS © 2020
Formal Aspects of Computing (2020) 32: 71–111

Formal Aspects of Computing

Formal reliability and failure analysis of ethernet based communication networks in a smart grid substation

Waqar Ahmad[1], Osman Hasan[2] and Sofiène Tahar[1]

[1]Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada.
[2]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan.

Abstract. Secure and continuous operation of a smart grid substation mainly depends upon the reliable functioning of its communication network. The communication system of a smart substation is typically based on a high performance Ethernet communication network that connects various intelligent embedded devices, such as Intelligent Electronic Devices (IED) and Merging Units (MU), to ensure continuous monitoring, automation and efficient demand response of the smart substation. Traditionally, Reliability Block Diagram (RBD) and Fault Tree (FT) methods are used to develop reliability and failure models for these communication networks by considering the failure characteristics of their substation intelligent embedded devices and other components, like transformers and circuit breakers. These resulting reliability and failure models are then analyzed using paper-and-pencil methods or computer simulations, but they cannot assure accuracy in the analysis due to their inherent limitations. As an accurate alternative, we propose a methodology, based on higher-order logic theorem proving, for conducting the formal RBD and FT-based reliability and failure analysis of smart substation communication networks, respectively. This paper also describes a sound transformation of smart grid FT models to their equivalent RBDs - a well-known method to reduce the complexity of FT-based failure analysis. Some ML-based tactics have been developed to automatically compute the reliability and failure probability of smart grid substations for practical purposes.

Keywords: Smart grid; Reliability block diagrams; Fault tree; Higher-order logic; Theorem proving.

1. Introduction

Nowadays, the electrification of modern society is among the biggest challenges as people across the globe are increasingly relying on continuous supply of electricity both at home and work. As an outset to this increasing reliance on electricity, the power blackouts and outages, which are a common occurrence in the traditional power grids, not only drastically affect the performance but also lead to heavy financial losses. For instance, the 2003 power blackout of 29 h in New York city kept about 800 million US$ of economic activity from taking place and destroyed 250 million US$ of perishable goods [MSN03]. To mitigate the effect of power outages, it has been realized that a close attention must be paid for conducting rigorous reliability analysis of substation automation systems (SASs) since the failure or malfunctioning of a SAS may pose great threat to power system functioning [KS09].

Correspondence to: W. Ahmad, E-mail: waqar@ece.concordia.ca

In modern smart substations, SAS is achieved using multi-functional microprocessor based Intelligent Electronic Devices (IEDs) [BTF05] replacing traditional panels, which incorporate stand-alone relays, control switches, meters, and status indicators. A major challenge faced by SAS design engineers is to ensure interoperability among the protection, control, and monitoring devices from the various manufacturers. To resolve this challenge, IEC 61850 [Mac06], a generic functional safety standard, proposes Ethernet communication networks, such as cascaded, ring, star-ring, and redundant ring, at station and process levels aiming to provide highly flexible communication architectures for power systems. This also leads to the development of several variants of Ethernet-based communication networks. For instance, a smart area backup protection (SABP) [CZXH13] communication network, provides a high performance substation protection system consisting of Ethernet switches that are transmitting real-time data from different components of the smart substation, such as Intelligent Electronic Devices (IED), Merging Units (MU), multiple voltage bus levels and circuit breakers, to the failure monitoring and protection system. However, due to the safety-critical nature of smart substations, the rigorous reliability and failure analysis of these Ethernet communication networks is a dire need. For this purpose, Reliability Block Diagram (RBD) [GB06] and Fault Tree (FT) [IEC06] methods, which are graphical techniques for analyzing the effect of the system component failures upon overall system reliability, have been the most commonly used reliability and failure modeling techniques for smart substations communication networks [KS09].

Traditionally, paper-and-pencil based analytical methods and computer simulations have been used for RBD and FT-based reliability and failure analysis of Ethernet communication networks in smart substations. In the paper-and-pencil based analytical methods, the first step is the construction of RBD and FT models for the given Ethernet communication network, consisting of IEDs, MUs, breakers and transformers. The second step in the RBD-based reliability analysis is the assignment of an appropriate failure distribution, such as Exponential or Weibull, with corresponding failure rate λ_i, which can be alternatively expressed in terms of a reliability attribute, i.e., Mean-time-to-failure (MTTF) [HK81], as $\lambda_i = \frac{1}{MTTF_i}$ [HK81], to each substation component. Since we are not analyzing the availability properties of smart substations, therefore, our analysis is not incorporating the repair attribute, i.e., Mean-time-to-repair (MTTR) [HK81]. Lastly, these failure distributions, along with the RBD model, are then used to analytically derive mathematical expressions for the overall substation reliability.

For the FT-based failure analysis, the analytical method starts with the identification of the Minimal Cut Set (MCS) of failure events that are associated with the components of the given smart substation, such as MUs and IEDs, which is then followed by the association of failure random variables, i.e., Exponential or Weibull, to these MCS failure events. The Probabilistic Inclusion-Exclusion (PIE) principle [Tri02] is then used to evaluate the exact probability of failure of the overall substation communication network. However, these analytical analysis methods, for RBDs and FTs, are prone to human error due to the involvement of manual manipulation and simplification especially when analyzing large and complex smart substation communication networks, such as the T1-1 smart substation communication network [KS09]. On the other hand, simulation tools, such as ReliaSoft [Rel] and ASENT Reliability analysis tool [ASE], provide graphical editors to construct substation RBD and FT models and the analysis is carried out by generating samples from the Exponential and Weibull random variables that are associated with the substation components [BHJ14]. These samples are then processed to evaluate the reliability and failure probability of the complete substation communication network using computer arithmetic and numerical techniques at the given mission time, i.e., a time at which the reliability or failure probability is desired. Certainly, these tools provide a more scalable and practical alternative to the analytical methods, but cannot ascertain an error-free analysis, which is highly recommended in the safety-critical smart substations as per the IEC 61850 standard [Mac06], because of the involvement of pseudo-random numbers and numerical methods and the inherent sampling-based nature of simulation.

Formal methods [HT14], which are computer-based mathematical reasoning techniques, have been used to overcome the inaccuracy limitations of paper-and-pencil proof methods and simulation and for analyzing RBD and FT models of a variety of systems, such as oil and gas pipelines [AHTH18], aerospace systems [BCK+09] and satellite solar arrays [AH15]. The main idea behind the formal analysis of a system is to first construct a mathematical model of the given system using a state-machine or an appropriate logic and then use logical reasoning and deduction methods to formally verify that this system exhibits the desired characteristics, which are also specified mathematically using an appropriate logic.

We can categorize formal methods into two mainstream techniques: model checking [BK08] and theorem proving [Har09]. Model checking is a state-based technique in which system behavior, specified as a state-machine, is analyzed by verifying the temporal properties exhaustively over the entire state-space of the formal model of the given system within a computer. While, theorem proving allows using logical reasoning to verify relationships

between a system and its properties as theorems, specified in an appropriate logic, using a computer. Both model checking and theorem proving have been used for probabilistic analysis of systems (e.g. [HT08, KNP10, EHTA11]), which is the foremost requirement for conducting reliability and failure analysis. Due to the state-based nature of model checking, it suits the Markov chain based reliability and failure analysis quite well and allows us to analyze only state related properties, like deadlock, liveness and safety. On the other hand, theorem proving based on the expressive higher-order-logic (HOL) [BD83] allows working with a variety of data types, such as lists and real numbers, and can be used to verify generic mathematical expressions. In this paper, given the involvement of several elements of continuous and random nature, we propose to conduct the formal RBD and FT analysis of a substation communication network by using higher-order-logic theorem prover technique [Har09]. For this purpose, we plan to build upon the recently proposed formal dependability analysis framework [Ahm17], based on higher-order-logic formalizations of RBD and FT. These foundational formalizations of RBD and FT have been successfully used to conduct the formal reliability analysis of virtual data centers [AHT16] and failure analysis of an air traffic management system [AH16]. However, to the best of our knowledge, formal methods have never been used to conduct RBD and FT-based reliability and failure analysis of communication networks in a smart substation, where the accuracy of the analysis is a prime concern. In fact, the primary reason of this work is to ensure the compliance of IEC 61508 standard, which highly recommends the use of formal modeling and analysis of critical-systems during different phases of system development [IEC].

The main novel contributions of the paper are as follows:

- A unified framework of RBD and FT formalizations in order to conduct the formal reliability and failure analysis of Ethernet communication networks in smart grid substations (Sect. 4).
- Formal modeling of generic RBDs for commonly used Ethernet communication architectures, such as cascaded, ring, star-ring and redundant ring, and formal verification of their corresponding reliability expressions in HOL (Sect. 7.1).
- Definition of explicit steps that are essential to analyze the reliability of real-world smart substation communication networks. For illustration purposes, the paper presents the formal reliability analysis of the T1-1 smart substation intra-bay and inter-bay communication networks, which can be realized as cascaded and redundant ring architecture, using our proposed formal framework in HOL (Sect. 7.2).
- Formal reasoning support related to the natural correspondence of RBD and FT techniques by presenting some verified lemmas that allow us to transform a given FT model to its equivalent RBD (Sect. 6.3). This transformation considerably facilitates the formal failure analysis of the SABP communication network compared to traditional PIE based FT analysis approach (Sect. 8).
- To minimize the user intervention in the formal verification process, the paper also presents some automatic evaluating functions, written in the functional programming language Standard Meta-Language (SML) [Pau96] that can simplify and numerically compute the reliability and failure probabilities of smart grid substation communication networks within the HOL environment (Sect. 9).
- For comparison purposes, the paper also analyzes the T1-1 smart substation inter-bay communication network exhibiting redundant ring architecture using the Raptor tool [MCGM03], which is a well-known simulation tool for dependability analysis. The results obtained from our proposed approach are arguably more accurate than the Raptor tool as it utilizes the monte-carlo simulation, which is a sampling based approach. Whereas, our proposed HOL SML functions evaluate the exact formally verified reliability expressions consisting of exponential functions.

The organization of the paper is as follows: Sect. 2 provides a survey of the related work on the reliability and failure analysis of smart grids. Section 3 describes an overview of theorem proving and the HOL theorem prover to facilitate the understanding of the rest of the paper. Section 4 presents a detailed description of the proposed methodology that has been used to carry out the reliability and failure analysis of smart grid communication systems. Sections 5 and 6 describe the formalizations of RBDs and FTs and their mutual relationship in HOL. Section 7 provides a generic HOL formalization of Ethernet communication architectures in a smart substation and the formal reliability analysis of T1-smart substation communication architectures. Section 8 presents the formal failure analysis of a smart grid communication network in a 220 kV smart substation by utilizing formalized FT gates. In Sect. 9, we discuss the utilization of SML functions for automatic simplification and computation of smart substation reliability and failure probability. Finally, Sect. 10 concludes the paper.

2. Related work

Chen et al. [CZXH13] studied the structural design and performance of a smart grid substation communication network and conducted its failure and reliability analysis by utilizing the Fault Tree (FT) and Reliability Block Diagrams (RBD) techniques [BA92]. Similarly, Rongrong et al. [YCPV13] proposed a generic FT-based reliability analysis method for industrial grids by considering the failure effects of Variable Frequency Devices (VFD), which are the power electronic devices that are widely utilized to increase system efficiency and reduce losses. Walfer et al. [WH13] extended the capabilities of the analytical reliability analysis method by combining the pivotal decomposition method with RBD and Markov modeling techniques for the dependability analysis of smart grids. Niyato et al. [NWH12] carried out the reliability and redundant design analysis for a smart grid wireless communication system using RBDs. A recent study also highlights that smart grid metering networks are playing a pivotal role in improving the reliability, efficiency, and sustainability of the traditional energy infrastructures [KLB+19]. This study also presented a brief overview of cyber attack real incidents in traditional energy networks and also investigated the open research issues in smart grid metering networks.

The IEC 61850 standard, released in 2003, highly recommended the use of Ethernet based communication networks at station and process level of smart grid substations. Since then, there has been an extensive amount of literature available focusing on the reliability and availability analysis of Ethernet communication networks. For instance, Kanabar et al. [KS09] are among the first ones to present an RBD-based reliability and availability analysis of the typical Ethernet switch architectures, such as cascaded, ring, star ring and redundant ring. They also demonstrated that these Ethernet architectures RBDs can be used to effectively analyze the reliability and availability of T1-1 smart substation. Similarly, Hajian-Hoseinabadi [HH11a] studied nine types of Ethernet based substation automation architectures, including cascaded and different types of star-ring topologies. The RBDs of these architectures are developed and then analyzed for availability analysis focusing on the effect of redundancy and repair. Ali et al. [ATGH15] presented a new Ethernet based substation communication network (SCN) by modifying the existing star-ring topology for substations. In particular, they further introduced the redundancy in the protection IEDs and the Ethernet switches and thus significantly improved the transmission reliability and the real-time performance of the message exchanges in a smart substation.

Colored Petri nets (CPN) have been used for modeling dynamic RBDs (DRBD) and FTs (DFT) [BCK+09], which are used to describe the dynamic reliability behavior of systems. CPN verification tools, based on model checking principles, are then used to verify behavioral properties of the DRBD and DFT models to identify reliability constraints in power generation plants [RXXZ10]. Similarly, Signoret et al. [SDC+13] described a modular approach to model the behaviour of RBDs using Petri nets. For illustration purposes, the reliability analysis of a High Integrity Pressure Protection System (HIPPS), which protects an industrial installation against overpressures, is presented. Kleyner et al. [KV10] presented an application of Stochastic Petri nets (SPN) to analyze the availability of an automotive electronics airbag controller. The COMPASS tool-set [BCK+09] supports dynamic FT analysis specifically for aerospace systems using the NuSMV and MRMC model checkers. The Interval Temporal Logic (ITS), i.e., a temporal logic that supports first-order logic, has been used, along with the Karlsruhe Interactive Verifier (KIV), for formal FT analysis of a rail-road crossing [OS07]. Similarly, Zeng et al. [ZJLS12] carried out the SPN based reliability and availability analysis of control center networks, which provide the interconnectivity and continuous monitoring by using a Wide Area Network (WAN) for smart grid substations. The authors of [ZJLS12] also discussed, in detail, the methods to reduce the state-space explosion problem for the case of large smart grids.

Probabilistic model checking has also been used for the development of a distributed probabilistic-control hybrid automata for analyzing smart grid applications [MPL11], the performance and energy consumption evaluation of smart grids [YZN+12], and the reliability analysis of protective relays in power distribution systems [KATH13]. However, to the best of our knowledge, none of the above-mentioned formal methods has been used for the reliability or failure analysis of smart substation Ethernet communication architectures. Moreover, due to the state-based nature of the above-mentioned techniques, such as model checking and Petri nets, they suffer from the state-space explosion problem for large smart grids [ZJLS12]. Particularly, this limitation of Petri nets is also mentioned by the authors of [SDC+13] while discussing the applicability of their approach to practical systems. On the other hand, the proposed theorem proving based approach has the ability to concisely model and verify generic reliability relationships, which is beneficial for RBD and FT based analysis of substation communication architectures.

A number of formalizations of probability theory are available in higher-order logic (e.g. [Hur03, MHT11, HH11b]). Hurd's formalization of probability theory [Hur03] has been utilized to verify sampling algorithms of a

number of commonly used discrete [Hur03] and continuous random variables [HT07] based on their probabilistic and statistical properties. However, Hurd's formalization of probability theory only supports having the whole universe as the probability space. This feature limits its scope of this probability theory and hence it cannot be used to formalize more than a single continuous random variable. Whereas, in the case of reliability and failure analysis of smart grid communication systems, multiple continuous random variables are required. The recent formalizations of probability theory by Mhamdi et al. [MHT11] and Hölzl et al. [HH11b] are based on extended real numbers (including $\pm\infty$) and provide the formalization of Lebesgue integral for reasoning about advanced statistical properties. These theories also allow using any arbitrary probability space that is a subset of the universe and thus are more flexible than Hurd's formalization [Hur03] and support formalizing multiple continuous random variables. Particularly, Mhamdis' probability theory formalization [MHT11] on extended-real numbers (real numbers including $\pm\infty$), has been recently used to reason about the RBD-based analysis of a virtual data center [AHT16] and failure analysis of an air traffic management system [AH16], which involves multiple exponential random variables.

In this paper, we mainly build our formal reliability and failure analysis based on static RBDs and FTs since these have been extensively used in the past for the analysis of Ethernet based Communication Networks in a Smart Grid Substation (See, e.g., [ATGH15, KS09]). Moreover, static RBD and FT have been effectively employed to perform the initial reliability and failure analysis of many other safety and mission critical applications as well. For instance, Bistouni et al. [BJ18] utilized RBDs to determine the reliability importance of switching elements in multistage interconnection networks (MINs), which are also known as shuffle exchange networks. Similarly, Oliva et al. [OLOV18] proposed a new method for solving complex static FTs, without any possibility of combinatorial explosion. For illustration purposes, the authors also applied their proposed method to 40 different FTs from the Aralia FTs database [Gro]. Moreover, industrial standards, such as IEC 65108 [Mac06] and ISO 26262 [PWHR11] highly recommend the use of static RBD and FT for the reliability and failure analysis of systems at various levels of abstraction. These standards also encourage the use of formal methods for most critical systems [IEC, PWHR11]. We believe that our work is a step towards providing a formal framework to engineers for conducting the formal reliability and failure analysis of not only smart substations but also for other critical systems, such as automotive and avionics. In particular, we endeavor to combine these RBD and FT formalizations and also go one step further by proposing a unified framework to conduct the formal reliability and failure analysis of smart grid substations. Additionally, we present the generic formalization of Ethernet communication architectures, such as cascaded, ring, star-ring, redundant ring, that can be used to conduct the reliability analysis of a wide variety of real-world smart substation communication systems. For practical purposes, this work also provides some SML functions that can be effectively used to automatically simplify and compute the formally verified reliability and failure relationships of smart substation communication networks. To the best of our knowledge, this is the first time that higher-order-logic theorem proving has been used for the analysis of reliability and failure related aspects of smart grids.

3. Preliminaries

In this section, we give a brief introduction to theorem proving and the HOL theorem prover to facilitate the understanding of the rest of the paper. We also summarize Mhamdi's formalization of probability theory [MHT11] in this section.

3.1. Theorem proving

Theorem proving [Gor89] is a widely used formal verification technique. The system that needs to be analyzed is mathematically modeled in an appropriate logic and the properties of interest are verified using computer based formal tools. The use of formal logics as a modeling medium makes theorem proving a very flexible verification technique as it is possible to formally verify any system that can be described mathematically. The core of theorem provers usually consists of some well-known axioms and primitive inference rules. Soundness is assured as every new theorem must be created from these basic or already proved axioms and primitive inference rules.

The verification effort of a theorem in a theorem prover varies from trivial to complex depending on the underlying logic [Har96]. For instance, first-order logic [Fit96] utilizes the predicate calculus and terms (constants, function names and free variables) and is semi-decidable.

Table 1. HOL symbols and functions

HOL4 symbol	Standard symbol	Meaning
∧	*and*	Logical *and*
∨	*or*	Logical *or*
¬	*not*	Logical *negation*
::	*cons*	Adds a new element to a list
++	*append*	Joins two lists together
HD L	*head*	Head element of list L
TL L	*tail*	Tail of list L
EL n L	*element*	nth element of list L
MEM a L	*member*	True if a is a member of list L
\x.t	$\lambda\, x.t$	Function that maps x to $t(x)$
SUC n	$n + 1$	Successor of a *num*
exp x	e^x	Exponential function
PREIMAGE f s	$f^{-1}(s)$	Function that takes a range set and returns a domain set

A number of sound and complete first-order logic automated reasoners are available that can enable automated proofs for a large set of problems. More expressive logics, such as higher-order logic [Bro07], can be used to model a wider range of problems than first-order logic, but theorem proving for these logics cannot be fully automated and thus involves user interaction to guide the proof tools. For reliability and failure analysis of smart grids, we need to formalize (mathematically model) random variables as functions and their distribution properties are verified by quantifying over random variable functions. Henceforth, first-order logic does not support such formalization and we need to use higher-order logic to formalize the foundations of reliability analysis of smart grids, which we have built upon to formalize RBDs and FTs in this paper.

3.2. HOL4 Theorem prover

HOL4 is an interactive theorem prover developed at the University of Cambridge, UK, for conducting proofs in higher-order logic. It utilizes the simple type theory of Church [Chu40] along with Hindley–Milner polymorphism [Mil77] to implement higher-order logic. HOL4 has been successfully used as a verification framework for both software and hardware as well as a platform for the formalization of pure mathematics.

HOL4 core consists of only 4 basic axioms and 8 primitive inference rules, which are implemented as ML functions. Soundness is assured as every new theorem must be verified by applying these basic axioms and primitive inference rules or any other previously verified theorems/inference rules.

We have utilized many HOL4 theories, including Booleans, lists, sets, positive integers, *real* numbers, measure and probability, in our work. In fact, one of the primary motivations of selecting HOL4 for our work was to benefit from these built-in mathematical theories. Table 1 provides the mathematical interpretations of some frequently used HOL4 symbols and functions, which are inherited from existing HOL4 theories, in this paper.

3.3. Probability theory and random variables in HOL4

Mathematically, a measure space is defined as a triple (Ω, Σ, μ), where Ω is a set, called the sample space, Σ represents a σ-algebra of subsets of Ω, where the subsets are usually referred to as measurable sets, and μ is a measure with domain Σ. A probability space is a measure space $(\Omega, \Sigma, \Pr)$, such that the measure, referred to as the probability and denoted by Pr, of the sample space is 1. In the HOL4 formalization of probability theory [MHT11], given a probability space p, the functions `p_space p`, `events p` and `prob p` return the corresponding Ω, Σ and Pr, respectively. This formalization also includes the formal verification of some of the most widely used probability axioms, which play a pivotal role in formal reasoning about reliability properties.

A random variable is a measurable function between a probability space and a measurable space. The measurable functions belong to a special class of functions, which preserves the property that the inverse image of each measurable set is also measurable. A measurable space refers to a pair $(S, \mathscr{A})$, where S denotes a set and $\mathscr{A}$

represents a nonempty collection of sub-sets of S. Now, if S is a set with finite elements, then the corresponding random variable is termed as a discrete random variable otherwise it is called a continuous one.

The probability that a random variable X is less than or equal to some value t, $\Pr(X \leq t)$ is called the cumulative distribution function (CDF) and it characterizes the distribution of both discrete and continuous random variables. The CDF has been formalized in HOL as follows [AHTH14]:

```
⊢ ∀ p X t. CDF p X t = distribution p X {y | y ≤ Normal t}
```

where the variables p, X and t represent a probability space, a random variable and a *real* number, respectively. The function `Normal` is a typecasting operator that takes a *real* number as its inputs and converts it to its corresponding value in the *extended-real* data-type, i.e, it is the *real* data-type with the inclusion of positive and negative infinity. The function `distribution` takes three parameters: a probability space p, a random variable X and a set of *extended-real* numbers and outputs the probability of a random variable X that acquires all the values of the given set in probability space p.

Now, reliability $R(t)$ is stated as the probability of a system or component performing its desired task over certain interval of time t.

$$\begin{aligned} R(t) = \Pr(X > t) &= 1 - \Pr(X \leq t) \\ &= 1 - F_X(t) \end{aligned} \tag{1}$$

where $F_X(t)$ is the CDF. The random variable X, in the above definition, models the time to failure of the system and is usually modeled by the exponential random variable with parameter λ, which corresponds to the failure rate of the system. Based on the HOL formalization of probability theory [MHT11], Eq. (1) has been formalized as follows [AHTH14]:

```
⊢ ∀ p X t. Reliability p X t = 1 - CDF p X t
```

The RBDs, described in this paper, are based on the notion of mutual independence of random variables, which is one of the most essential prerequisites for reasoning about the mathematical expressions for all RBDs. If N reliability events are mutually independent then

$$Pr\left(\bigcap_{i=1}^{N} E_i\right) = \prod_{i=1}^{N} Pr(E_i) \tag{2}$$

This concept has been formalized as follows [AHTH14]:

```
⊢ ∀ p L.
mutual_indep p L = ∀ L1 n. PERM L L1 ∧
1 ≤ n ∧ n ≤ LENGTH L ⇒
  prob p (inter_list p (TAKE n L1)) =
  list_prod (list_prob p (TAKE n L1))
```

The function `mutual_indep` accepts a list of events L and probability space p and returns $True$ if the events in the given list are mutually independent in the probability space p. The predicate `PERM` ensures that its two lists as its arguments form a permutation of one another. The function `LENGTH` returns the length of the given list. The function `TAKE` returns the first n elements of its argument list as a list. The function `inter_list` performs the intersection of all the sets in its argument list of sets and returns the probability space if the given list of sets is empty. The function `list_prob` takes a list of events and returns a list of probabilities associated with the events in the given list of events in the given probability space. Finally, the function `list_prod` recursively multiplies all the elements in the given list of real numbers. Using these functions, the function `mutual_indep` models the mutual independence condition such that for any 1 or more events n taken from any permutation of the given list L, the property $Pr(\bigcap_{i=1}^{N} E_i) = \prod_{i=1}^{N} Pr(E_i)$ holds. The HOL formalization of some of the functions that are used in this paper are described in Table 2.

4. Proposed methodology

The proposed methodology for the formal reliability and failure analysis of smart grids, depicted in Fig. 1, allows us to formally verify reliability and failure probability expressions corresponding to the given Ethernet communication network description and thus formally check that the given smart substation satisfies its requirements.

Table 2. HOL Functions used throughout Formalization

Function Name	HOL Definition
PERM	⊢ ∀ L1 L2. PERM L1 L2 = ∀ x. FILTER ($= x) L1 = FILTER ($= x)L2
LENGTH	⊢ (LENGTH [] = 0) ∧ ∀ h t. LENGTH (h::t) = SUC (LENGTH t)
TAKE	⊢ (∀ n. TAKE n [] = []) ∧ ∀ n x xs. TAKE n (x::xs) = if n = 0 then [] else x::TAKE (n − 1) xs
inter_list	⊢ (∀ p. inter_list p [] = p_space p) ∧ ∀ p h t. inter_list p (h::t) = h ∩ inter_list p t
list_prod	⊢ (∀ list_prod [] = 1) ∧ ∀ h t. list_prod (h::t) = h * list_prod t
list_prob	⊢ (∀ p. list_prob p [] = []) ∧ ∀ p h t. list_prob p (h::t) = prob p h * list_prob p t
one_minus_list	⊢ (one_minus_list [] = []) ∧ ∀ h t. one_minus_list (h::t) = 1 − h::one_minus_list t)
FLAT	⊢ (FLAT [] = []) ∧ ∀ h t. FLAT (h::t) = h ++ FLAT t
list_sum	⊢ (list_sum [] = 0) ∧ ∀ h t. list_sum (h::t) = h + list_sum t

The core components of this methodology are the higher-order logic formalizations of probability theory [MHT11], Reliability Block Diagrams (RBDs) [AHT16] and Fault Trees (FTs) [AH16], as depicted in a container, enclosed by a dashed-box, in Fig. 1. These core HOL formalizations are then used to formalize the Ethernet network topologies, such as cascaded, ring, star-ring, and redundant ring.

The first step in the proposed methodology is to identify the Ethernet communication networks in a given smart grid substation. The second step is to construct the RBD or FT models of these Ethernet communication networks. These models can then be formalized in higher-order logic using the above-mentioned core formalizations, particularly using the formalization of RBD or FT gates. The next step is to assign the failure distributions, like Exponential and Weibull, to each network components, such as IEDs and MUs. A *proof goal* or a theorem describes the reliability/failure property of an Ethernet communication network operating in a smart substation. It can be constructed by specifying an appropriate failure distribution and the formal RBD or FT model of an Ethernet communication network. It is important to include all necessary conditions with a proof goal, so that a user can reason about its correctness in a HOL theorem prover. We developed a rich library of formally verified generic lemmas and theorems, which are part of the above-mentioned HOL core theories shown in a container in Fig. 1, enabling a user with some basic knowhow about theorem proving to verify the reliability or failure probability expressions of Ethernet communication networks in a smart substation. Lastly, if all proof goals are discharged then we get formally verified expressions. Otherwise, we can use the failing proof goals to debug our formalizations of the model and requirements or the originally specified model.

5. Formalization of the reliability block diagrams

Commonly used RBD configurations for the reliability analysis of real-world systems, include series, parallel and a combination of both, and are depicted in Fig. 2. In this section, we briefly present their formalization, which in turn can then be used to formally model the structures of a smart grid substation communication network in HOL and reason about their reliability characteristics.

The formalization process of RBDs [AHT16] starts by type abbreviating the notion of event, which is essentially a set of observations with type $'a \rightarrow bool$ as follows:

```
type_abbrev ("event" , ``:'a ->bool'')
```

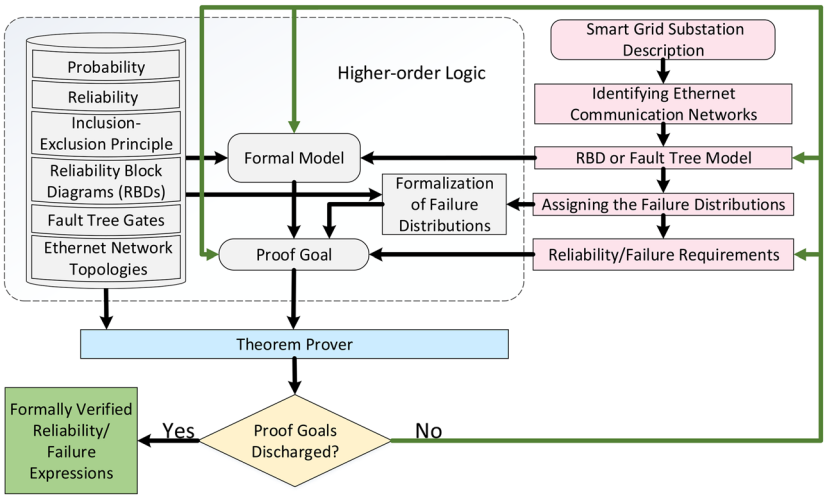


Fig. 1. Methodology for formal reliability and failure analysis of smart grid substation

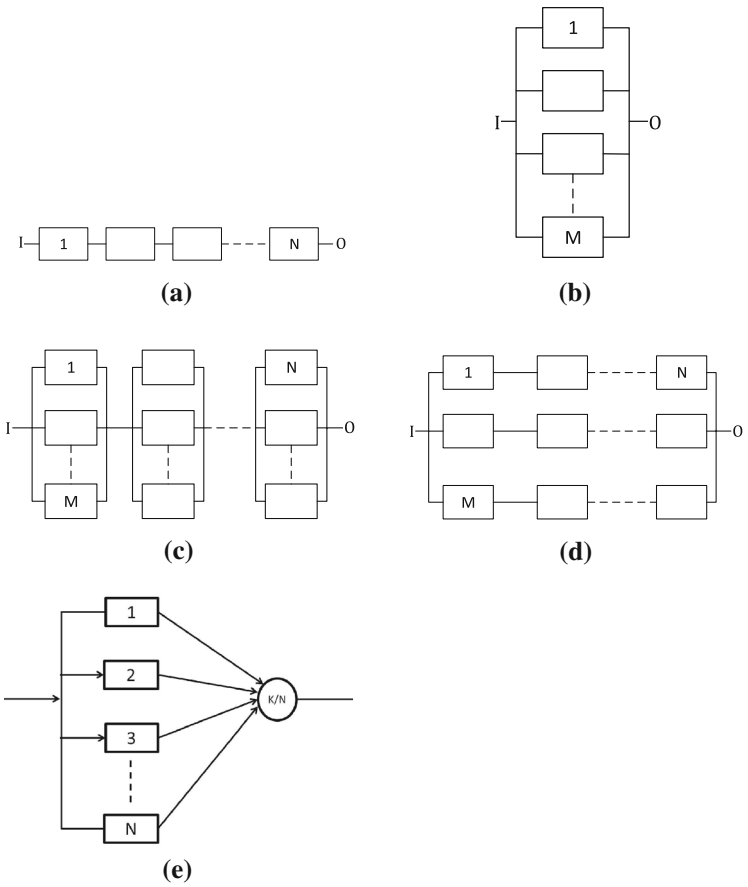


Fig. 2. Reliability block diagrams **a** series **b** parallel **c** series–parallel **d** parallel–series **e** K-out-N

A recursive datatype *rbd* is then defined in the HOL system as follows:

```
Hol_datatype ‘rbd = series of rbd list |
                    parallel of rbd list |
                    atomic of ’a event‘
```

An RBD can either be a series configuration, a parallel configuration or an atomic event. The type constructors `series` and `parallel` recursively function on *rbd*-typed lists and thus enable us to deal with complex RBD configurations. The type constructor `atomic` is basically a typecasting operator between *event* and *rbd*-typed variables.

A semantic function `rbd_struct` is defined over *rbd* datatype in order to extracts the corresponding event from the given RBD configuration as follows:

Definition 5.1 ⊢
```
(∀ p. rbd_struct p (series []) = p_space p) ∧
(∀ xs x p.
   rbd_struct p (series (x::xs)) =
   rbd_struct p x ∩ rbd_struct p (series xs)) ∧
(∀ p. rbd_struct p (parallel []) = {}) ∧
(∀ xs x p.
  rbd_struct p (parallel (x::xs)) =
  rbd_struct p x ∪ rbd_struct p (parallel xs)) ∧
(∀ p a. rbd_struct p (atomic a) = a)
```

The above function decodes the semantic embedding of an arbitrary RBD configuration by extracting the corresponding reliability event, which can then be used to determine the reliability of a given RBD configuration. The function `rbd_struct` takes an *rbd*-typed list identified by a type constructor `series` and returns a complete probability space (`p_space p`) if the given list is empty and otherwise returns the intersection of the events in order to model the series RBD configuration behaviour. Similarly, to model the behaviour of a parallel RBD configuration, the function `rbd_struct` operates on an *rbd*-typed list encoded by a type constructor `parallel`. It then returns the union of the events after applying the function `rbd_struct` on each element of the given list or an empty set if the given list is empty. The function `rbd_struct` returns the reliability event using the type constructor `atomic`.

In the subsequent sections, we present the HOL formalization of RBDs on any reliability event list of arbitrary length. The notion of reliability event is then incorporated in the formalization while carrying out the reliability analysis of a real smart grid substation, as it will be described in Sect. 7.

5.1. Formalization of series reliability block diagram

The reliability of a system with components connected in series is considered to be reliable at time t only if all of its components are functioning reliably at time t, as depicted in Fig. 2a. If $A_i(t)$ is a mutually independent event that represents the reliable functioning of the ith component of a serially connected system with N components at time t, then the overall reliability of the complete system can be expressed as [GB06]:

$$R_{series}(t) = Pr\left(\bigcap_{i=1}^{N} A_i(t)\right) = \prod_{i=1}^{N} R_i(t) \tag{3}$$

Now using Definition 5.1, we can formally verify the reliability expression, given in Eq. 3, for a series RBD configuration in HOL as follows [Ahm17]:

Theorem 5.1 ⊢ ∀ p L.
```
¬NULL L ∧
rbd_conds p L ⇒
  (prob p (rbd_struct p (series (rbd_list L))) =
   list_prod (list_prob p L))
```

The first assumption, in Theorem 5.1, ensures the list of events L, representing the reliability of individual components, must have at least one event. The second assumption represented by the predicate function `rbd_conds` contains the conditions, in addition to the first assumption, that are necessary for the verification of Eq. (3).

The function `rbd_conds` contains following predicates: (i) `prob_space p` which ensures that p is a valid probability space; (ii) `in_events p L` ensures that all the corresponding events in the given list L are drawn from the events space p; and (iv) `mutual_indep p L` guarantees that events in the given list L are mutually independent [AHTH14]. The conclusion of the theorem represents Eq. (3). The function `rbd_list` generates a list of type *rbd* by mapping the function `atomic` to each element of the given event list L to make it consistent with the assumptions of above theorem.

The proof of Theorem 5.1 is primarily based on mutual independence properties and some fundamental axioms of probability theory.

5.2. Formalization of parallel reliability block diagram

The reliability of a system with parallel connected sub-modules, depicted in Fig. 2b, mainly depends on the component with the maximum reliability. In other words, the system will continue functioning as long as at least one of its components remains functional. If the event $A_i(t)$ represents the reliable functioning of the ith component of a system with N parallel components at time t, then the overall reliability of the system can be mathematically expressed as [Bir04]:

$$R_{parallel}(t) = Pr\left(\bigcup_{i=1}^{N} A_i(t)\right) = 1 - \prod_{i=1}^{N}(1 - R_i(t)) \tag{4}$$

Similar to the formalization of series RBD, the reliability expression for the parallel RBD configuration, given in Eq. (4), is formalized in HOL as follows:

Theorem 5.2 ⊢ ∀ p L.
¬NULL L ∧
rbd_conds p L ⇒
(prob p (rbd_struct p (parallel (rbd_list L))) =
1 - list_prod (one_minus_list (list_prob p L)))

The above theorem is verified under the same assumptions as Theorem 5.1. The conclusion of the theorem represents Eq. (4) where, the function `one_minus_list`, described in Table 2, accepts a list of *real* numbers $[x1, x2, x3, \ldots, xn]$ and returns the list of *real* numbers such that each element of this list is 1 minus the corresponding element of the given list, i.e., $[1 - x1, 1 - x2, 1 - x3, \ldots, 1 - xn]$.

The above formalization, described for series and parallel RBD configurations, is then used to formalize the combination of series and parallel RBD configurations. The type constructors `series` and `parallel` can take a list containing other *rbd* type constructors, such as `series`, `parallel` or `atomic`, allowing the function `rbd_struct` to yield the corresponding event for an RBD configuration that is composed of a combination of series and parallel RBD configurations.

5.3. Formalization of series–parallel reliability block diagram

If in each serial stage the components are connected in parallel, as shown in Fig. 2c, then the configuration is termed as a series–parallel structure. If $A_{ij}(t)$ is the event corresponding to the proper functioning of the jth component connected in an ith subsystem at time index t, then the reliability of the complete system can be expressed mathematically as follows [GB06]:

$$R_{series-parallel}(t) = Pr\left(\bigcap_{i=1}^{N}\bigcup_{j=1}^{M} A_{ij}(t)\right) = \prod_{i=1}^{N}\left(1 - \prod_{j=1}^{M}(1 - R_{ij}(t))\right) \tag{5}$$

Using Theorems 5.1 and 5.2, the generic reliability expression for series–parallel RBD configuration, given in Eq. (6), is formalized in HOL as follows:

Theorem 5.3 ⊢ ∀ p L.
```
not_null L ∧
rbd_conds p (FLAT L) ⇒
  (prob p (rbd_struct p
    ((series of (λa. parallel (rbd_list a))) L)) =
  (list_prod of
    (λa. 1 - list_prod (one_minus_list (list_prob p a)))) L)
```

The above theorem is verified under the same assumptions as Theorem 5.1. The conclusion of the Theorem 5.3 represents Eq. (4) where, the HOL function `FLAT` is used to flatten the two-dimensional list, i.e., to transform a list of lists, into a single list. The function `not_null` ensures that each member of the two-dimensional list L must not be an empty list. The function `of` is defined as an infix operator [GM93] in order to connect the two rbd-typed constructors by using the HOL `MAP` function and thus facilitates the natural readability of complex RBD configurations.

5.4. Formalization of parallel–series reliability block diagram

If $A_{ij}(t)$ is the event corresponding to the reliability of the jth component connected in a ith subsystem at time t, then the reliability of the complete system can be expressed as follows:

$$R_{parallel-series}(t) = Pr\left(\bigcup_{i=1}^{M}\bigcap_{j=1}^{N} A_{ij}(t)\right) = 1 - \prod_{i=1}^{M}\left(1 - \prod_{j=1}^{N}(R_{ij}(t))\right) \tag{6}$$

Similarly, the generic expression of the parallel–series RBD configuration, given in Eq. (6), is formalized in HOL as follows:

Theorem 5.4 ⊢ ∀ p L. `not_null L ∧`
```
rbd_conds p (FLAT L) ⇒
 (prob p
   (rbd_struct p ((parallel of (λa. series (rbd_list a))) L)) =
  1 - (list_prod o (one_minus_list) of
    (λa. list_prod (list_prob p a))) L)
```

The assumptions of Theorem 5.4 are similar to those used in Theorem 5.3. The conclusion models the right-hand side of Eq. (6). Further, details about the proof of Theorems 5.3 and 5.4 can be found in [AHT16].

5.5. k-out-of-n Reliability block diagram

A n-component system is said to be in the k-out-of-n configuration if we need at least k components out of the total n components to be functional for the overall functionality of the system [Nar05]. The RBD for a k-out-of-n configuration is depicted in Fig. 2c. This behaviour can be modeled by utilizing the concept of binomial trials, which are used to find the chances of at least k success in n trials. Now, if R is the reliability of the k-components that are functioning correctly among n-components then the reliability of the overall system can be expressed mathematically as follows [Nar05]:

$$R_{k|n}(t) = Pr\left(\bigcup_{i=k}^{n}\{\textit{exactly i components functioning properly}\}\right) = \sum_{i=k}^{n}\left(\binom{n}{k}R^i(1-R)^{n-i}\right) \tag{7}$$

The HOL formalization of k-out-of-n RBD is as follows [AHTH18]:

Definition 5.2 ⊢ ∀ p X k n.
```
k_out_n_struct p X k n =
BIGUNION (IMAGE (λx. PREIMAGE X {Normal (&x)} ∩ p_space p)
            {x | k ≤ x ∧ x < SUC n})
```

The function k_out_n_struct accepts a probability space p, a binomial random variable X and two variables, k and n, which represent the number of successes and total number of trials, respectively. It then returns the union of the corresponding events that are associated with the binomial random variable X which takes values from the set $\{x \mid k \leq x \wedge x < SUC\ n\}$. The function IMAGE takes a function f and an arbitrary domain set and returns a range set by applying the function f to all the elements of the given domain set. The function BIGUNION returns the union of all the element of given set of sets.

Based on the Definition 5.2, the mathematical expression of k-out-of-n configuration, given in Eq. (7), is verified in HOL as follows:

Theorem 5.5 ⊢ ∀ p n k X R.
prob_space p ∧ k ≤ SUC n ∧
in_events_k|n p X n ∧
bino_dist_rand p X R n ⇒
(prob p (k_out_n_struct p X k n) =
sum (k, SUC n - k)
(λx. (&binomial n x)*(R pow x)*(1 - R) pow (n-x)))

The first and second assumptions ensure that p is a valid probability space and the number of successes of trails k must be less than or equal the total number of trials n. In the third assumption, the function in_events_k|n takes a probability space p, a time-to-failure random variable X and a natural number n and makes sure that all the n corresponding events that are associated with the random variable X are drawn from the events space p. The function bino_dist_rand, in the last assumption, takes the probability space p, the time-to-failure random variable X, the success probability R and the natural number n and ensures that the random variable X is exhibiting a binomial distribution with success probability R, which in our case is the reliability of each of the n-identical components connected in a k-out-of-n structure. The conclusion of the theorem represents Eq. (7).

6. Formalization of fault trees

Fault Tree (FT) analysis is one of the widely used technique to conduct the failure analysis of real-world systems, like communication networks at smart substations. It mainly provides a graphical model for analyzing the conditions and factors causing an undesired top event, i.e., a critical event, which can cause the complete system failure upon its occurrence. The preceding nodes of the FT are represented by gates, like OR, AND and XOR, which are used to link two or more cause events of a fault in a prescribed manner.

6.1. Formalization of fault tree gates

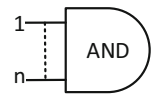
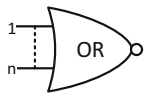
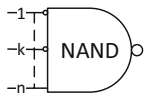
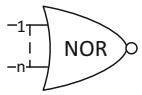
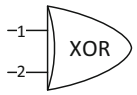
The FT gates are formally modeled by using a new recursive datatype *gate* in HOL as follows [AH16]:

```
Hol_datatype ‘gate = AND of gate list |
                    OR of gate list |
                    NOT of gate |
                    atomic of ’a event‘
```

The type constructors AND and OR recursively function on *gate*-typed lists and the type constructor NOT operates on *gate*-type variables. The type constructor atomic is basically a typecasting operator between *event* and *gate*-typed variables.

A semantic function is then defined over the *gate* datatype that can yield the corresponding event from the given FT gate as follows:

Table 3. HOL4 Formalization of Fault Tree Gates

FT Gates	HOL Formalization
	⊢ ∀ p L1 L2. AND_FT_gate p L1 L2 = FTree p (AND (gate_list L)
	⊢ ∀ p L. OR_FT_gate p L = FTree p (OR (gate_list L))
	⊢ ∀ p L1 L2. NAND_FT_gate p L1 L2 = FTree p (AND (gate_list (compl_list p L1 ++ L2)))
	⊢ ∀ p L. NOR_FT_gate p L = FTree p (NOT (OR (gate_list L)))
	⊢ ∀ p A B. XOR_FT_gate p A B = FTree p (OR [AND [NOT A; B]; AND [A; NOT B]])

Definition 6.1 ⊢

```
(∀ p. FTree p (AND []) = p_space p) ∧
(∀ xs x p.
    FTree p (AND (x::xs)) =
    FTree p x ∩ FTree p (AND xs)) ∧
(∀ p. FTree p (OR []) = {}) ∧
(∀ xs x p.
     FTree p (OR (x::xs)) =
     FTree p x ∪ FTree p (OR xs)) ∧
(∀ p a.
     FTree p (NOT a) =
     p_space p DIFF FTree p a) ∧
(∀ p a. FTree p (atomic a) = a)
```

The function `FTree` takes a list of type *gate*, identified by the type constructor `AND`, and returns a complete probability space `p_space p` if a given list is empty and otherwise returns the intersection of events in a given list. Similarly, to model the behavior of the OR FT gate, the function `FTree` returns the union of all the events after applying the function `FTree` on each element of a given list or an empty set if a given list is empty. The function `FTree` takes the type constructor `NOT` and returns the complement of a failure event obtained from the function `FTree`. The function `FTree` returns the failure event using the type constructor `atomic`.

If the occurrence of a failure event at the output is caused by the occurrence of all input failure events then this kind of behavior can be modeled by using the AND FT gate. Similarly, in the OR FT gate, the occurrence of an output failure event depends upon the occurrence of any one of its input failure event. The NOT FT gate can be used in conjunction with the AND and OR FT gates to formalize other FT gates. The formalization of these gates [AH16] is given in Table 3. The NAND FT gate, represented by the function `NAND_FT_gate` in Table 3, models the behavior of the occurrence of an output failure event when at least one of the failure events at its input does not occur.

Table 4. Probability of Failures of Fault Tree Gates

Mathematical expressions	Theorem's conclusion
$F_{AND}(t) = Pr(\bigcap_{i=2}^{N} A_i(t)) = \prod_{i=2}^{N} F_i(t)$	⊢ ∀ p L1 L2. (prob p (AND_FT_gate L) = list_prod (list_prob p L))
$F_{OR}(t) = Pr(\bigcup_{i=2}^{N} A_i(t)) = 1 - \prod_{i=2}^{N}(1 - F_i(t))$	⊢ ∀ p L1 L2. (prob p (OR_FT_gate p L) = 1 - list_prod (one_minus_list (list_prob p L)))
$F_{NAND}(t) = Pr(\bigcap_{i=2}^{k} \overline{A}_i(t) \cap \bigcap_{j=k}^{N} A_i(t)) = \prod_{i=2}^{k}(1 - F_i(t)) * \prod_{j=k}^{N}(F_j(t))$	⊢ ∀ p L1 L2. (prob p (NAND_FT_gate p L1 L2) = list_prod ((list_prob p (compl_list p L1))) * list_prod (list_prob p L2))
$F_{NOR}(t) = 1 - F_{OR}(t) = \prod_{i=2}^{N}(1 - F_i(t))$	⊢ ∀ p L. (prob p (NOR_FT_gate p L) = list_prod (one_minus_list (list_prob p L)))
$F_{XOR}(t) = Pr(\overline{A}(t)B(t) \cup A(t)\overline{B}(t)) = (1 - F_A(t))F_B(t) + F_A(t)(1 - F_B(t))$	⊢ ∀ p A B. (prob p (XOR_FT_gate p (A) (B) = (1 - prob p A) * prob p B + prob p A * 1 - prob p B)

This type of gate is used in FTs when the non-occurrence of a failure event in conjunction with the other failure events causes the top failure event to occur. This behavior can be expressed as the intersection of complementary and normal events, where the complementary events model the non-occurring failure events and the normal events model the occurring failure events. The output failure event occurs in the 2-input XOR FT gate if only one, and not both, of its input failure events occur.

The verification of the corresponding failure probability expressions, of the above-mentioned FT gates, is presented in Table 4. These expressions are verified under the following assumptions: (i) `prob_space p` ensures that p is a valid probability space; (ii) `2 ≤ LENGTH L` makes sure that the given list must have at least two elements; (iii) `in_events p L` ensures that all the corresponding events in the given list L are drawn from the events space p; and (iv) `mutual_indep p L` guarantees that events in the given list L are mutually independent [AHTH14].

6.2. Formalization of probabilistic inclusion–exclusion principle

In FT analysis, firstly all the basic failure events are identified that can cause the occurrence of the system top failure event. These failure events are then combined to model the overall fault behavior of the given system by using the fault gates. These combinations of basic failure events, called cut sets, are then reduced to minimal cut sets (MCS) by using some set-theory rules, such as idempotent, associative and commutative. Then, the Probabilistic Inclusion Exclusion (PIE) principle is used to evaluate the overall failure probability of the given system based on the MCS events. According to the PIE principle, if A_i represents the ith basic failure event or a combination of failure events then the overall failure probability of the given system can be expressed as follows:

$$\mathbb{P}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{t \neq \{\}, t \subseteq \{1,2,\ldots,n\}} (-1)^{|t|+1} \mathbb{P}\left(\bigcap_{j \in t} A_j\right) \tag{8}$$

The above equation has been formally verified in HOL as follows [AH15]:

Theorem 6.1 ⊢ ∀ p L.
prob_space p ∧ in_events p L ⇒
(prob p (union_list L) =
 sum_set {t | t ⊆ set L ∧ t ≠ {}}
         (λt. -1 pow (CARD t + 1) * prob p (BIGINTER t)))

The assumptions of the above theorem are the same as the ones used in Theorem 6.1. The function sum_set takes an arbitrary set s with element of type α and a real-valued function f and recursively sums the return values of the function f, when applied on each element of the given set s. In the above theorem, the set s is represented by the term $\{x \mid C(x)\}$ that contains all the values of x, which satisfy condition C. Whereas, the λ abstraction function (λt. -1 pow (CARD t + 1) * prob p (BIGINTER t)) models $(-1)^{|t|+1}\ \mathbb{P}(\bigcap_{j\in t} A_j)$, such that the functions CARD and BIGINTER return the number of elements and the intersection of all the elements of the given set, respectively.

6.3. Transformation of FT to RBD

Typically, the FT-based failure analysis is carried out using the PIE principle, which takes the union of events and returns the sum of ℙ(all individual events) + ℙ(the intersection of every compound event) + ℙ(the intersection of all events). The number of output terms in the PIE principle is equivalent to the number of terms resulting from the power set expansion. However, the computation complexity of the PIE principle is exactly *O(2n)*, and no matter what the events are, the complexity order cannot be decreased [Che14]. Several methods have been proposed over the years to reduce the complexity issues of FT analysis. A tractable solution is to transform the given system FT to its equivalent Reliability Block Diagram (RBD) [BA92], described in Sect. 5. This transformation considerably reduces the analysis complexity due to the fact that an RBD offers closed form expressions compared to a FT, which requires unfolding of all the PIE terms.

Interestingly, there is a natural correspondence between FT gates and the commonly used RBD configurations, which can be seen in Fig. 3. A sound transformation of FT gates to their corresponding RBDs can be done by formally verifying the equivalence relationships in HOL. For instance, an AND FT gate can be translated to series RBD, depicted in Fig. 3a, by verifying the following lemma in HOL:

Lemma 1 ⊢ ∀ p L.
FTree p (AND (gate_list L)) =
rbd_struct p (series (rbd_list L))

Due to the fact that we have adopted the same approach for the formalization of RBDs and FTs, the correspondence between them can be clearly observed in Lemma 1. Similarly, we can also formally verify the equivalence of OR FT gate to its corresponding parallel RBD, shown in Fig. 3b. Table 5 presents the formally verified relationship of the OR FT gate to parallel RBD and other FT gates, such as NAND, NOR, XOR, inhibit and comparator, to their corresponding RBDs. Beside these FT gates, we can also transform generic FT structures to their equivalent RBDs. For instance, the generic AND-OR FT can be transformed to series–parallel RBD and similarly OR-AND FT to parallel–series RBD, as depicted in Fig. 3c and d. These FT transformations greatly helped us in analyzing the FTs of smart grid substations, described in Sect. 8. However, it is important to note that the structural transformation of the FT gates to their corresponding RBD is one-to-one. The difference in the structure occurs when one is translating a failure tree to a success RBD model. For instance, consider the OR FT gate for which all failure events must occur for the output failure to occur. If we are considering the failure after the transformation then it is transformed into the parallel RBD. On the other hand, if we are analyzing the successful working after the transformation, then it has to be transformed into the series RBD [Rel15] (Fig. 4).

7. Formal reliability analysis of smart grid substation communication architectures

A typical smart grid substation consists of Intelligent Electronic Devices (IEDs), Merging Units (MUs), transformers, multiple voltage bus levels and various line or voltage bays as depicted in Fig. 4a. The bays are used to connect the input power line to the busbar assembly and typically consist of circuit breakers and disconnectors. The IEDs are microprocessor-based controllers for power system equipment, such as circuit breakers, transformers and capacitor banks.

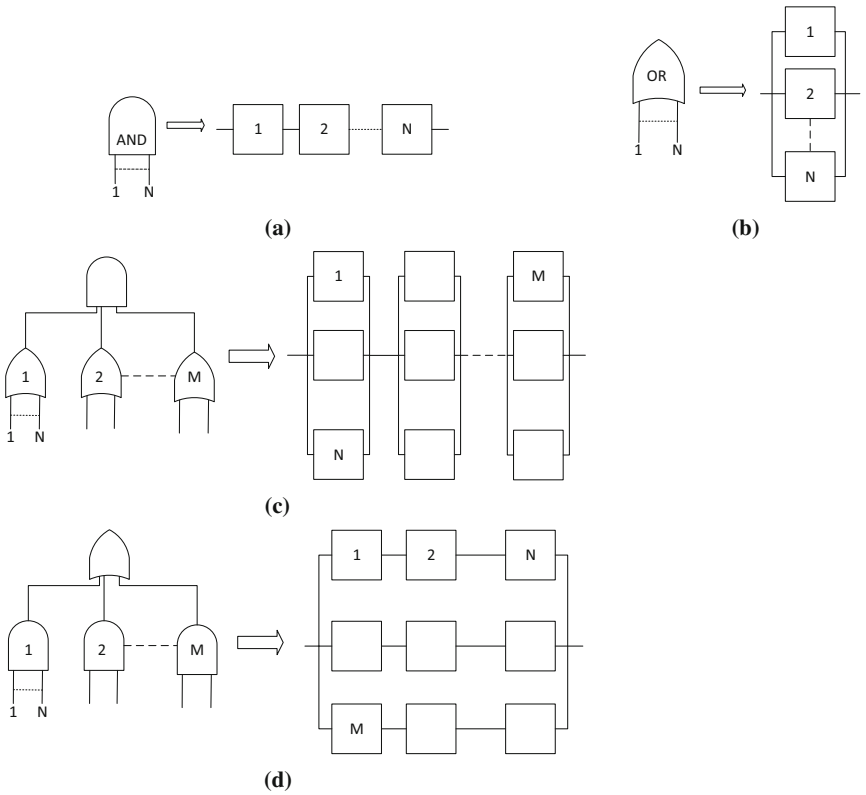


Fig. 3. a Series RBD to AND FT Gate. **b** Parallel RBD to OR FT Gate. **c** Series–Parallel RBD to AND-OR FT. **d** Parallel–Series RBD to OR-AND FT

They are the most integral part of the substation automation system (SAS) and serve various functions including electrical protection, advanced local control intelligence and monitor processes and then communicate directly to a control system. The protection IEDs (PIEDs) serve many functions for the substation, including protection from rate of change of voltage frequency, fuse failure and breaker failure. Similarly, the control IEDs (CIEDs) provide various functions and specialized interfaces to implement IEC 6180 communication standards for monitoring the substation process. MUs are also a type of IEDs, which synchronously collect digital signals from electronic current/voltage transformers (CTs/VTs), and transmit these signals to protective devices and measure-control devices. The Time Synchronization (TS) devices provide precise time synchronization for protection and control functions in substations. The IEC 6180 standard [ABM10] provides a mechanism to achieve interoperability among multi-vendor substation automation devices, such as IEDs and MUs, for communication networks within the smart grid substations. It also provides the ability to synchronize internal clocks, using the TS devices, in the IEDs, the MUs, the protection/control units, and the Ethernet switches, and thus helps to achieve accurate control and precise global analysis of communication network responses in a substation against failures.

IEC 61850 based substation automation systems (SAS) can be typically classified into three distinct levels, as shown in Fig. 4b: (1) Station level; (2) Bay level; and (3) Process level. The station level mainly facilitates the exchange of data among different bays, such as line and transformer. In particular, those functions that require data from bays are implemented at this level. Similarly, the Bay level consists of IEDs, which are responsible for the protection and control of data exchanges between the bay and the station levels. Lastly, the process level provides an interface between the bay level and the substation components, such as CTs/ VTs, remote I/O, and actuators. At this level, the instant exchange of status and control data messages between the process and the bay level is highly desired.

Different kinds of Ethernet communication architectures, such as cascaded, ring and star-ring and redundant star-ring, have been used at the bay level to realize intra-bay and inter-bay communications. In the intra-bay communication, the data is exchanged between the local IEDs within the respective bay, such as line, transformer and bus bays. Whereas in the inter-bay communication, the data messages are exchanged among different bays, for instance, between line and transformer bays.

Table 5. Transformation of FT Gates to RBDs

FT Gate to RBD	HOL Formalization
OR FT Gate to Parallel RBD	∀p L. FTree p (OR (gate_list L)) = rbd_struct p (parallel (rbd_list L))
NAND FT Gate Transform	∀L1 L2 n p. (FTree p (AND (gate_list (compl_list p L1 ++ L2)))) = (rbd_struct p (series (rbd_list (compl_list p L1 ++ L2))))
NOR Gate Transform	∀p L. p_space p DIFF FTree p (OR (gate_list L)) = p_space p DIFF rbd_struct p (parallel (rbd_list L))
Inhibit Gate Transform	∀p A B C. prob_space p ∧ C ∈ events p ⇒ (FTree p (AND [OR [A; B]; NOT C]) = (rbd_struct p (parallel [A; B]) ∩ (p_space p DIFF rbd_struct p C)))
Comparator Gate Transform	∀p A B. prob_space p ∧ A ∈ events p ∧ B ∈ events p ⇒ (FTree p (OR [AND [A; B]; NOT (OR [A; B])]) = rbd_struct p (series [A; B]) ∪ (p_space p DIFF rbd_struct p (parallel [A; B])))
AND-OR to series–parallel	∀p L. (FTree p ((AND of (λa. OR (gate_list a))) L) = rbd_struct p ((series of (λa. parallel (rbd_list a))) L))
OR-AND to parallel–series	∀p L. (FTree p ((OR of (λa. AND (gate_list a))) L) = rbd_struct p ((parallel of (λa. series (rbd_list a))) L))

In this work, we assume that all the equipment taking part in establishing the Ethernet communication network in a smart grid substation are smart electronic devices, such as IEDs, MUs and TS. Therefore, it is quite practical to assume that the failures of these smart devices are mutually independent since the devices are independent physically and also in their functionality [KS09, CZXH13]. In the following sections, we describe, in detail, the commonly used Ethernet switch architectures and the construction of their corresponding RBD models for smart substations (Sect. 7.1). We also present the HOL formalization of these RBDs and their utilization to conduct the formal reliability analysis of the T1-1 smart substation communication network in HOL (Sect. 7.2).

7.1. Commonly used ethernet communication architectures

In this section, we first present generic models of commonly used Ethernet architectures to realize communication in smart substations. Then, we describe the construction of their corresponding RBD models, which can be used to formally analyze the reliability of a wide variety of smart substations utilizing these Ethernet architectures.

7.1.1. Cascaded architecture

The cascaded architecture can be considered as a serial connection of n-Ethernet Switches (ESWs) allowing, without loop communication between the devices in control station, such as servers and Human Machine Interfaces (HMI), and the IEDs, to connect with the substation components, as shown in Fig. 5a. It is the simplest architecture and also least expensive, but leads to a high latency during the data exchange. Kanaber et al. [KS09] modeled the cascaded architecture RBD for T1-1 substation communication network by serially connecting the RBD blocks for the PIEDs on one side and the CIEDs on the other side of the Ethernet switches.

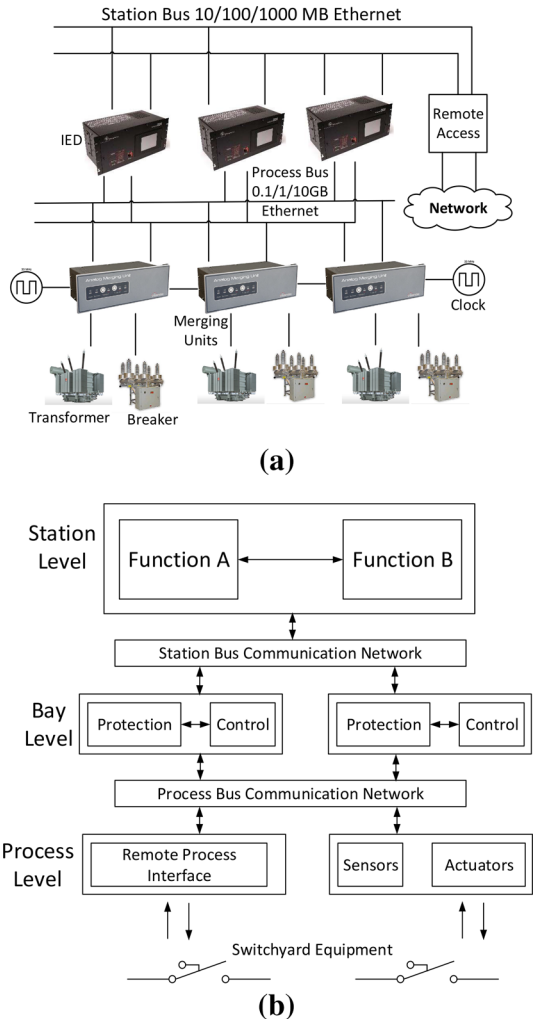


Fig. 4. a A typical smart substation ethernet architecture. **b** Functional hierarchy of IEC 61850 based substation automation system

Whereas, Ali et al. [ATGH15] included additional RBD-blocks of MUs and time synchronization (TS) devices in series for modeling the cascaded architecture's RBD. In order to construct a generic RBD model for the cascaded architecture, we introduce the RBD blocks for protection and control devices (PCDs), such as PIED, CIED, TS, and MU and connect them on either side of the Ethernet switches RBD blocks, as depicted in Fig. 5b. A prime benefit of these PCDs RBD blocks is that we can now easily model a generic cascaded architecture RBD having an arbitrary number of RBD blocks.

Since all equipment in the smart substation communication network are electronic devices, their failure rate remains constant during the useful period of life and hence, their failure distribution is considered as Exponential with a failure rate λ, and a time index t [KS09]. The reliability of the substation cascaded architecture can be expressed mathematically as follows:

$$R_{series\_casc}(t) = e^{-(\Sigma_i^n \lambda_{PCDs\_l} + \Sigma_j^n \lambda_{ESW(i)} + \Sigma_k^n \lambda_{PCDs\_r})(t)} \tag{9}$$

where the parameter λ is typically derived from the reliability metric Mean-time-to-failure (MTTF) using the relation $\lambda = \frac{1}{MTTF}$. The HOL formalization of the cascaded substation architecture RBD configuration, as shown in Fig. 5b, is as follows:

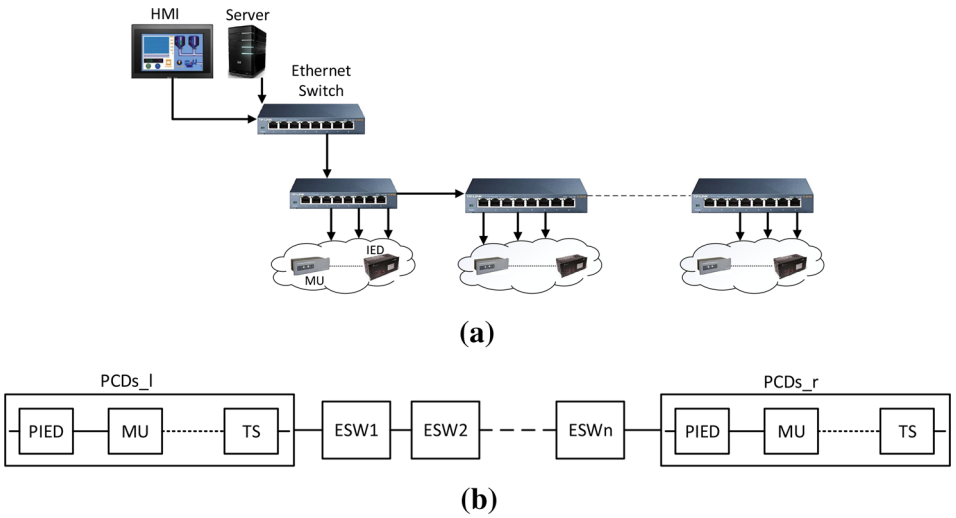


Fig. 5. **a** Cascaded architecture. **b** Cascaded architecture RBD

Definition 7.1 ⊢ ∀ p t PCDs_l PCDs_r ESW_list.
casc_arch_RBD p PCDs_l ESW_list PCDs_r t =
rbd_struct p (series
(rbd_list (rel_event_list p (PCDs_r ++ ESW_list ++ PCDs_r) t)))

where $PCDs\_l$, and $PCDs\_r$ are the lists of random variables associated with the protection and control devices, such as PIED, CIED and TS, and ESW_list represents the Ethernet switch random variable list containing n-Ethernet switches. The function `rel_event_list` returns the list of events by mapping the function `rel_event` (Definition 7.2) on each element of the given list of random variables. The symbol ++ in HOL is used for appending the lists.

Definition 7.2 ⊢ ∀ p X t.
rel_event p X t = PREIMAGE X {y | t < y} ∩ p_space p

The above definition models the conventional notion of reliability event $X > t$, described in Eq. (1), using the HOL function `PREIMAGE` (Table 1). The failure event can simply be modeled by taking the complement of reliability event with respect to a probability space p. It can be formalized in HOL as:

Definition 7.3 ⊢ ∀ p X t.
fail_event p X t = p_space p DIFF rel_event p X t

Using Definition 7.1, we formally verify the reliability expression, given in Eq. (9), of the generic substation cascaded architecture RBD model (Fig. 5b) in HOL as follows:

Theorem 7.1 ⊢ ∀ PCDs_l PCDs_r ESW_list C_PCDs_l C_PCDs_l C_ESW_list p t.
time_positive t ∧
rbd_conds (rel_event_list p (PCDs_l++ESW_list++PCDs_r) t) ∧
¬NULL (PCDs_l++ESW_list++PCDs_r) ∧
exp_dist_list p
(PCDs_l++ESW_list++PCDs_r)
(C_PCDs_l++C_ESW_list++C_PCDs_r) ⇒
(prob p (casc_arch_RBD p PCDs_l ESW_list PCDs_r t) =
exp (-list_sum (C_PCDs_l++C_ESW_list++C_PCDs_r) * t))

The first assumption in the above theorem ensures that the time index t must be positive. The next assumption contains the necessary predicates, as described in Theorem 5.1, associated with the series RBD configuration. The third assumption makes sure that the random variable list ESW_list, associated with the Ethernet switches, must not be empty.

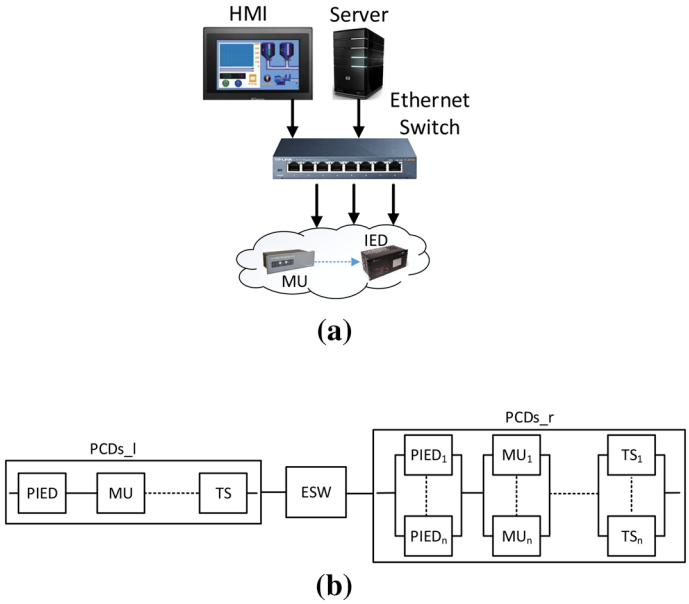


Fig. 6. a Star architecture for substation. **b** RBD of star architecture

In the last assumption, the function `exp_dist_list` provides the relationship between the Exponential random variables, i.e., *PIED*, *CIED* and the list *ESW_list*, associated with the substation components connected in a cascaded architecture, and their respective failure rates *C_PIED*, *C_CIED* and *C_ESW_list*, i.e., `exp_dist_list [x1;x2] [c1;c2] = (∀t. 0 ≤ t ⇒ ((Pr(ω p x1 t) = 1 − e^{−λc1 ∗t}) ∧ (Pr(ω p x2 t) = 1 − e^{−λc2 ∗t}))`. Here, the symbol ω represents the function `fail_event` described in Definition 7.3. The function `list_sum` recursively sums all the elements in a given list (Table 2). The conclusion of Theorem 7.1 models the series RBD configuration of an arbitrary substation cascaded architecture. The proof of Theorem 7.1 is primarily based on Theorem 5.1, exponential function properties and probability theory axioms.

7.1.2. Star architecture

In the star architecture, all components are connected to a central Ethernet switch with an individual communication cable, as shown in Fig. 6a. Unlike the cascaded architecture, there is only one main Ethernet switch in a star architecture. The RBD model of the star architecture, as shown in Fig. 6b, is quite similar to the cascaded architecture RBD except it contains a single Ethernet switch. To develop a generic RBD for a star architecture, we connect a series RBD block and a series–parallel RBD block of PCDs on either side of the Ethernet switch to model the non-redundant and redundant components connected to the Ethernet switch, respectively. The reliability expression for the star architecture RBD can be expressed mathematically as:

$$R_{series\_casc}(t) = e^{-(\Sigma_i^n \lambda_{PCDs\_l} + \lambda_{ESW})} * \prod_{j=1}^{n} \left(1 - \prod_{k=1}^{n} (1 - e^{-\lambda_{PCDs\_r(jk)}\, t})\right) \tag{10}$$

Using Definition 7.1, we can formally model the RBD of the star architecture in HOL as:

Definition 7.4 `⊢ ∀ p t PCDs_l PCDs_r ESW.`
```
star_arch_RBD p PCDs_l ESW PCDs_r t =
casc_arch_RBD p PCDs_l [ESW] [] t ∩
 rbd_struct p (series (MAP (λa. parallel (rbd_list a))
    (two_dim_rel_event_list p PCDs_r t))))
```

where *PCDs_r* is a two-dimensional list of random variables that are associated with each protection and control device connected in the series–parallel structure. The function `two_dim_rel_event_list` takes a two-dimensional list, say L, and maps the functions `rel_event_list`, already explained in Definition 7.1 on each element of list L.

Now, we can formally verify Eq. 10 using the above definition in HOL as:

Theorem 7.2 ⊢ ∀ PCDs_l PCDs_r ESW C_PCDs_l C_PCDs_l ESW p t.
```
time_positive t ∧
rbd_conds (FLAT
  (two_dim_rel_event_list p
    ([PCDs_l]++[[ESW]]++PCDs_r) t)) ∧
not_null [PCDs_l] ++ PCDs_r ∧
(LENGTH C_PCDs_l = LENGTH PCDs_l) ∧
(LENGTH C_PCDs_r = LENGTH PCDs_r) ∧
(∀n.
  n < LENGTH PCDs_r ⇒
  (LENGTH (EL n PCDs_r) = LENGTH (EL n C_PCDs_r))) ∧
two_dim_exp_dist_list p
   ([PCDs_l]++[[ESW]]++PCDs_r)
   ([C_PCDs_l]++[[C_ESW]]++C_PCDs_r) ⇒
 (prob p (star_arch_RBD p PCDs_l ESW PCDs_r t) =
  exp (-list_sum (C_PCDs_l++[C_ESW]) * t) *
  (list_prod of
   (λa. 1 - list_prod
    (one_minus_list (exp_func_list a t)))) C_PCDs_r))
```

where the function `not_null` ensures that every list of the given two-dimensional list must not be empty. The function `two_dim_exp_dist_list` accepts two lists, i.e., a two-dimensional list of random variables and failure rates, corresponding to the components at each stage of a series–parallel RBD. It recursively calls the function `exp_dist_list` to ensure that all elements of a given list are Exponentially distributed with the corresponding failure rates, given in the list within the probability space p.

7.1.3. Ring architecture

In the ring architecture, all Ethernet switches are connected in a single loop, as shown in Fig. 7a. In the ring communication network, unlike the cascaded architecture, only k-out-n Ethernet switches are required for inter-bay communication. This redundancy in the Ethernet switches is achieved based on the Rapid Spanning Tree Protocol (RSTP) [RI09], which can reconfigure the communication network in case of failure. Similar to the star architecture, we also connect the series–parallel RBD block in the ring architecture RBD for modeling the redundant components, as shown in Fig. 7b. The reliability expression of the ring architecture for smart substations can be expressed mathematically as:

$$\begin{aligned} R_{ring-arc}(t) = e^{-(\Sigma_i^n \lambda_{PCDs\_l} + \Sigma_i^n \lambda_{ESW(i)})(t)} * \\ \sum_{i=k}^{n} \left(\binom{n}{k} (e^{-\lambda_{ESWs}(t)})^i (1 - e^{-\lambda_{ESWs}(t)})^{n-i} \right) * \\ \prod_{j=1}^{n} \left(1 - \prod_{k=1}^{n} (1 - e^{-\lambda_{PCDs\_r(jk)}\, t}) \right) \end{aligned} \tag{11}$$

By using our formalization of cascaded architecture RBD (Definition 7.1) and k-out-n RBDs (Definition 5.2), we can formally model the ring architecture RBD in HOL as follows:

Definition 7.5 ⊢ ∀ p PCDs_l ESW_list PCDs_r k n t.
```
ring_arch_RBD p PCDs_l ESW_list X_bino PCDs_r n k t =
(casc_arch_RBD p PCDs_l ESW_list [] t) ∩
 K_out_N_struct p X_bino k n ∩
 rbd_struct p (series (MAP (λa. parallel (rbd_list a))
    (two_dim_rel_event_list p PCDs_r t))))
```

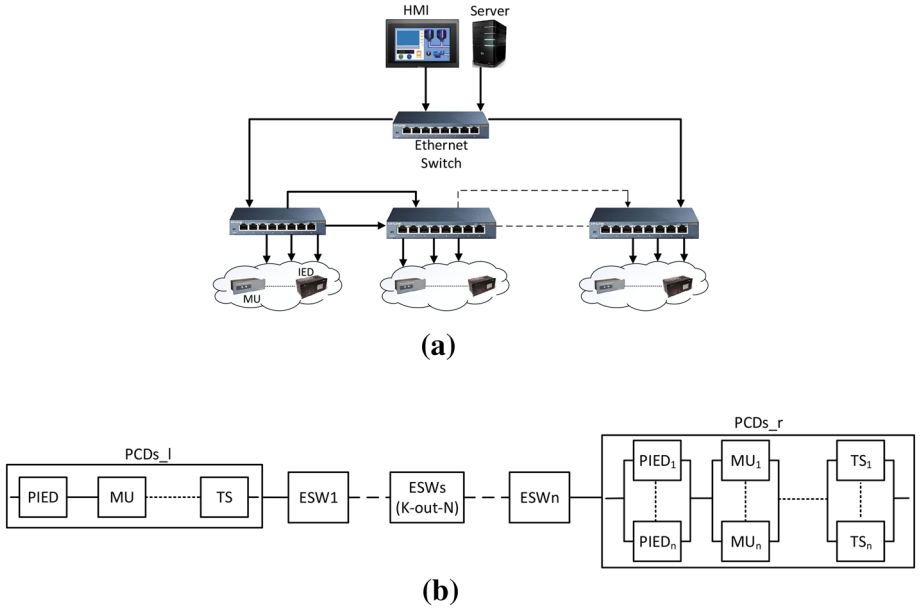


Fig. 7. a Ring architecture for substation. **b** RBD of ring architecture

Based on the above definition, we can formally verify Eq. 11 in HOL as follows:

Theorem 7.3 ⊢ ∀ PCDs_l PCDs_r ESW_list C_PCDs_l C_PCDs_r C_ESW_list p t X_bino k n ESWs.
binomial_conds p X_bino ESWs k n [X_bino] t ⇒
(prob p
   (ring_arch_RBD p PCDs_l ESW_list X_bino PCDs_r n k t) =
 exp (-list_sum (C_PCDs_l++C_ESW_list) * t) *
 sum (k,SUC n - k)
     (λi. &binomial n x * exp (-C_ESWs * t) pow i *
     (1 - exp (-C_ESWs * t)) pow (n - i))) *
  (list_prod of
   (λa. 1 - list_prod
    (one_minus_list (exp_func_list a t)))) C_PCDs_r))

where the predicate binomial_conds contains the conditions that are associated with the k-out-n RBD, as described in Theorem 5.5. The assumptions of the above theorem are similar to the ones used in Theorems 5.5 and 7.1 and the conclusion of the theorem models in Eq. 11. For simplicity, we only mention those assumptions with Theorem 7.3 that are additional to the ones used in Theorem 7.2. The proof of the above theorem is based on Theorems 5.1 and 5.5, exponential function properties and some probability theory axioms.

7.1.4. Star-ring architecture

To implement the star-ring architecture, two additional redundant Ethernet switches are required in a ring. Both these Ethernet switches are connected to the rest of the switches forming a star configuration, as shown in Fig. 8a. In the RBD of the star-ring architecture, the redundant Ethernet switches are connected to all bays Ethernet switches. Hence, both redundant Ethernet switches are connected in parallel in the RBD, as depicted in Fig. 8b. Mathematically, the reliability of the star-ring architecture, with components exhibiting exponential failure distribution, can be expressed as follows:

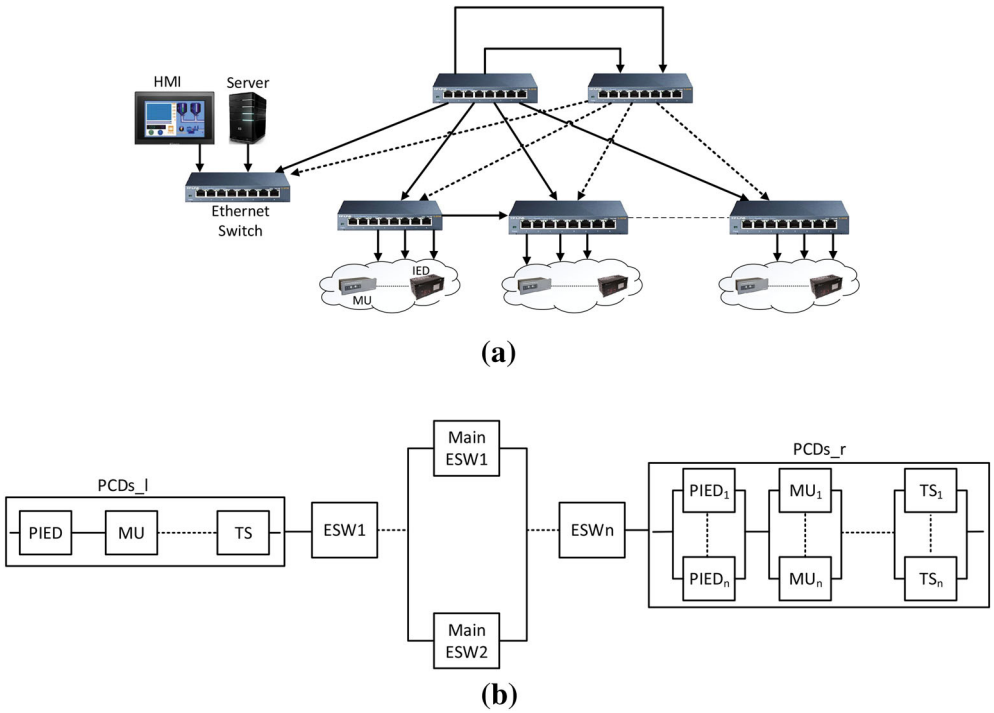


Fig. 8. a Star-ring architecture for substation. **b** Star-ring RBD

$$
\begin{aligned}
R_{star-ring-arc}(t) = e^{-(\Sigma_i^n \lambda_{PCDs\_l} + \Sigma_i^n \lambda_{ESW(i)})(t)} * \\
(1 - (1 - e^{-\lambda_{MainESWs1}(t)})(1 - e^{-\lambda_{MainESWs2}(t)})) * \\
\prod_{j=1}^{n} \left(1 - \prod_{k=1}^{n} (1 - e^{-\lambda_{PCDs\_r(jk)}\, t})\right)
\end{aligned}
\tag{12}
$$

We can formally model the star ring architecture RBD, shown in Fig. 8b, by using the formalizations of cascaded architecture RBD (Definition 7.1), parallel (Sect. 5.2) and series–parallel RBDs (Sect. 5.3) in HOL as follows:

Definition 7.6 ⊢ ∀ p PCDs_l ESW_list PCDs_r Main_ESW1 Main_ESW2 t.
star_ring_arch_RBD p PCDs_l ESW_list PCDs_r Main_ESW1 Main_ESW2 t =
(casc_arch_RBD p PCDs_l ESW_list [] t) ∩
rbd_struct p (parallel (rbd_list
(rel_event_list [Main_ESW1;Main_ESW2] t))) ∩
rbd_struct p (series (MAP (λa. parallel (rbd_list a))
(two_dim_rel_event_list p PCDs_r t)))

We can formally verify Eq. 12 using the above definition in HOL as follows:

Theorem 7.4 ⊢ ∀ PCDs_l ESW_list PCDs_r Main_ESW1 Main_ESW2 C_PCDs_l
C_PCDs_r C_ESW_list C_Main_ESW1 C_Main_ESW2 p t.
(prob p
(star_ring_arch_RBD p PCDs_l ESW_list PCDs_r Main_ESW1 Main_ESW2 t =
exp (-list_sum (C_PCDs_l++C_ESW_list) * t) *
(1 - (1 - exp(-C_Main_ESW1 * t)) * (1 - exp(-C_Main_ESW2 * t))) *
(list_prod of
(λa. 1 - list_prod
(one_minus_list (exp_func_list a t)))) C_PCDs_r))

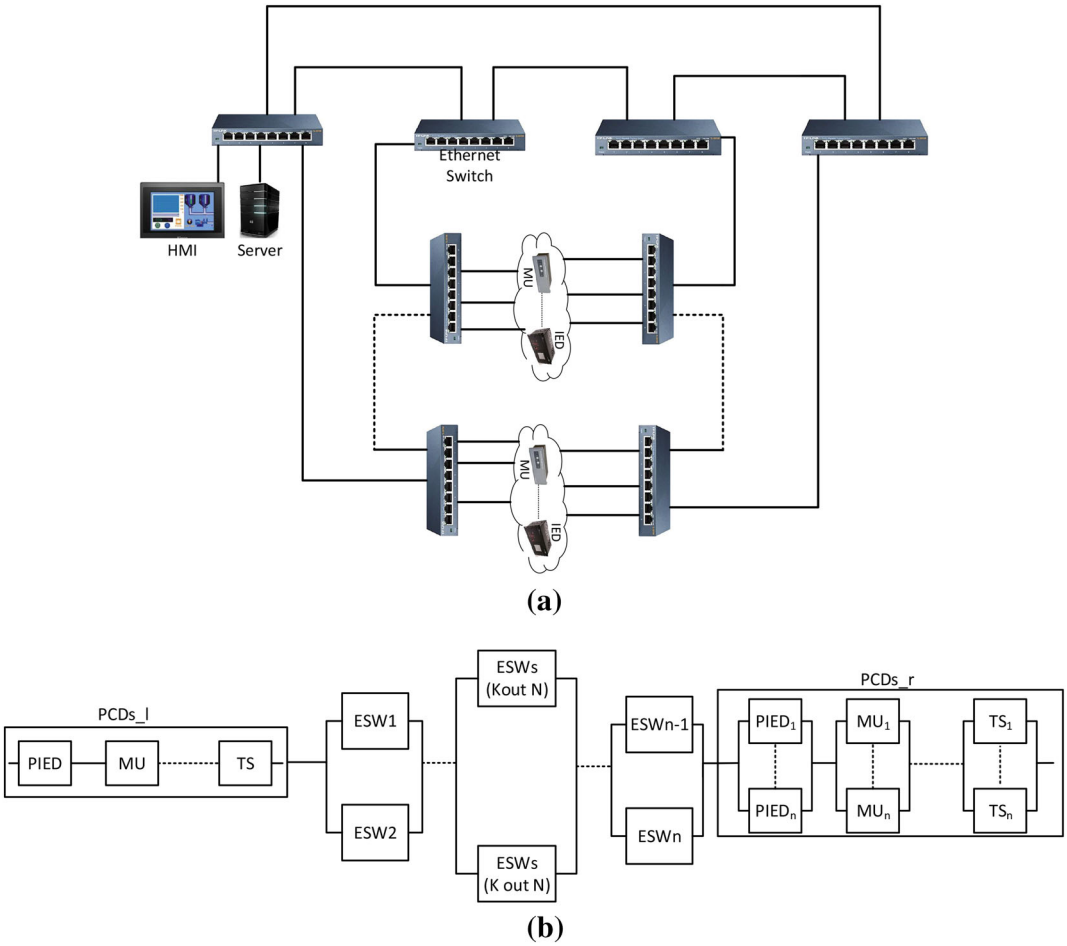


Fig. 9. a Redundant ring architecture for substation. **b** Redundant ring RBD

The assumptions of the above theorem are omitted since they are similar to the ones used in Theorem 7.2 and the conclusion of the theorem models in Eq. 12. The proof of Theorem 7.4 is based on Theorems 5.1 and 5.2, exponential function properties and some probability theory axioms.

7.1.5. Redundant ring architecture

The redundant ring architecture provides maximum redundancy and protects the communication network in the case of failure. In Fig. 9a, it can be seen that all Ethernet switches are connected in a two redundant rings configuration. These redundant rings are then connected again in a ring architecture through the Ethernet switches. In the RBD of the redundant ring architecture, as shown in Fig. 9b, the IEDs are connected to two redundant Ethernet switches, hence, ESW components are connected in parallel for both PIEDs. Since each IED is also connected to two redundant ring configurations, therefore, both the ring configuration should be connected in parallel, represented by a k-out-n RBD. This redundant ring architecture can be modeled by using the series–parallel RBD configuration and its reliability can be expressed mathematically as follows:

$$R_{RR}(t) = \left(1 - \left(1 - \sum_{i=k}^{n} \left(\binom{n}{k} (e^{-\lambda_{ESWs}(t)})^i (1 - e^{-\lambda_{ESWs}(t)})^{n-i}\right)\right)^2\right) \\ e^{\sum_i^n \lambda_{PCDs\_l}\, t} * \left\{\prod_{i=1}^{n} (1 - (1 - e^{\lambda_{ESWi1}\, t})(1 - e^{\lambda_{ESWi2}\, t}))\right\} * \\ \prod_{j=1}^{n} \left(1 - \prod_{k=1}^{n} (1 - e^{-\lambda_{PCDs\_r(jk)}\, t})\right) \tag{13}$$

We formally model the redundant star-ring architecture RBD in HOL as follows:

Definition 7.7
```
⊢ ∀ p t PCDs_l PCDs_r list_ESW_list X_bino k n.
redund_ring_RBD p PCDs_l PCDs_r list_ESW_list X_bino k n t =
(casc_arch_RBD p PCDs_l [] [] t) ∩
rbd_struct p (parallel
  (rbd_list (K_out_N_struct_list p [X_bino; X_bino] k n))) ∩
 rbd_struct p (series (MAP (λa. parallel (rbd_list a))
  (two_dim_rel_event_list p (list_ESW_list ++ PCDs_r) t))))
```

The above definitions are generic enough to model the reliability of any smart substation exhibiting redundant ring communication architecture by specializing the two-dimensional list `list_ESW_list` to a sub-list containing random variables associated with the ESWs in the parallel stage, as given in Fig. 9b.

Theorem 7.5
```
⊢ ∀PCDs_l ESWs PCDs_r list_ESW_list C_PCDs_l C_PCDs_r
C_list_ESW_list C_ESWs p k n t.
time_positive t ∧
k < SUC n ∧
rbd_conds (FLAT
  (K_out_N_struct_list p [X_bino; X_bino] k n::
   two_dim_rel_event_list p
    ([PCDs_l]++list_ESW_list++PCDs_r) t)) ∧
not_null list_ESW_list ∧
binomial_conds p X_bino ESWs k n [X_bino;X_bino] t
exp_dist p ESWs C_ESWs ∧
two_dim_exp_dist_list p
   ([PCDs_l]++list_ESW_list++PCDs_r)
   ([C_PCDs_l]++C_list_ESW_list++C_PCDs_r) ⇒
(prob p
  (redund_ring_RBD p PCDs_l PCDs_r list_ESW_list X_bino k n t) =
exp (-list_sum (PCDs_l) * t) *
(1 - (1 - sum (k,SUC n - k)
           (λx. &binomial n x * exp (-C_ESWs * t) pow x *
           (1 - exp (-C_ESWs * t)) pow (n - x))) pow 2) *
(list_prod of
  (λa. 1 - list_prod
   (one_minus_list (exp_func_list a t)))) (C_list_ESW_list++C_PCDs_l)
```

The function `K_out_N_struct_list` maps the function `K_out_N_struct`, described in Definition 5.2, on each element of the given list.

The theorems, presented in Sects. 7.1.1–7.1.5, are of generic nature, i.e., all variables are quantified for all values. They can be easily utilized to conduct a complete and sound reliability analysis of Ethernet communication architectures for many real-world smart grid substations. A key challenge in these theorems is to verify the independence relationship between different types of RBDs. For instance, in Theorem 7.1.5, we formally verify the independence relationship between the series–parallel RBD and parallel K-out-N RBDs. For illustration purposes, in the next section, we present the formal reliability analysis of Ethernet communication architectures deployed in the T1-1 smart grid substation.

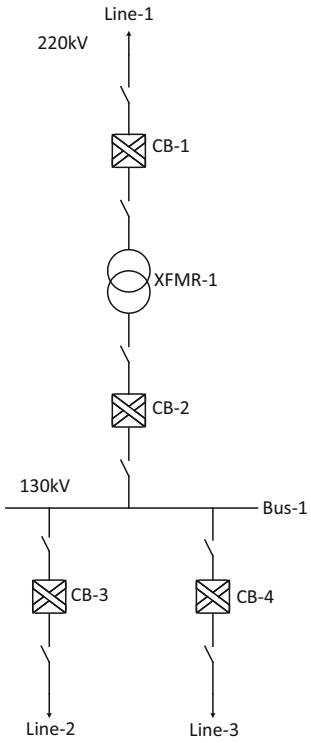


Fig. 10. T1-1 substation communication architectures

7.2. Formal reliability analysis of a T1-1 smart grid substation architecture

We consider a 220/130KV T1-1 typical substation with a single bus consisting of 5 line bays, i.e., 3 for the line bays, 1 for the transformer and 1 for the bus [KS09], and its layout is shown in Fig. 10. Each bay has redundant PIEDs with 2 MUs for the bus and transformer bays, as both bays require AC signals from 2 to 3 sets of current/voltage transformers (CTs/VTs). Moreover, the feeder bay requires only 1 MU.

The steps described below provide a guideline for the formal reliability assessment of smart grid substation communication networks, which can be modeled by using commonly used Ethernet architectures (Sect. 7.1).

Step 1: Analyze the substation communication network and identify the Ethernet communication architecture. For instance, if the protection and control devices, such as IEDs, MUs and TS, are connected directly with a single Ethernet switch then it is a star configuration (Sect. 7.1.2).

Step 2: Construct an RBD diagram by serially connecting the non-redundant RBD blocks at the left-hand-side of the Ethernet switch(s) and the redundant RBD blocks at the right-hand-side.

Step 3: Assign the failure rates to each substation components, which are used in the construction of the RBD.

Step 4: Formally verify the reliability of the smart substation Ethernet communication network by utilizing the generic theorems presented in Sect. 7.1.

We now conduct the RBD-based formal reliability analysis of IEC 6180 classified T1-1 substation's [KS09] intra-bay and inter-bay communication network exhibiting the above-mentioned features and having the cascaded, star and redundant ring architectures (Fig. 10).

7.2.1. Intra-bay communication network

The reliability analysis of the intra-bay communication allows us to analyze the availability of the communication components for a particular bay communication. In the T1-1 smart substation, there are three kinds of bays, i.e., line, transformer and bus. It can be seen that the line bay has redundant PIEDs and also all other components, such as TS, MUs, CIED, and Ethernet switch must be in working state to enable the intra-bay communication,

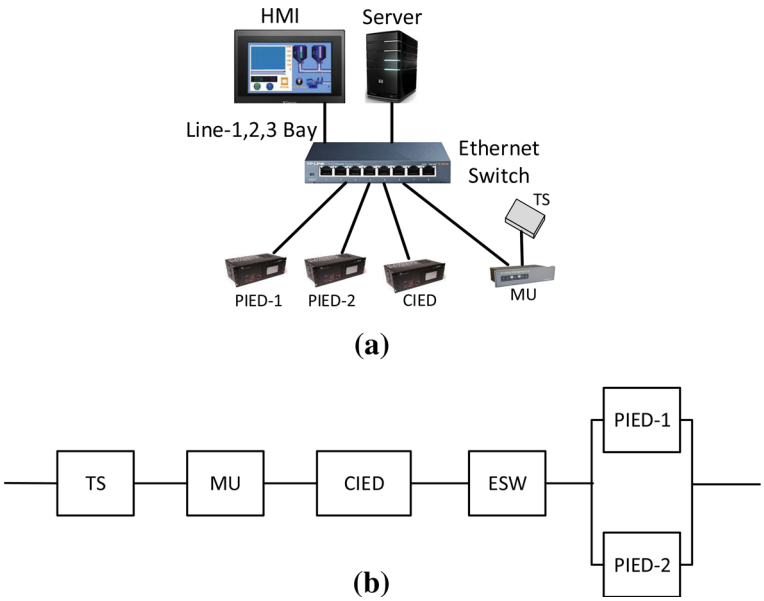


Fig. 11. a Star architecture for the T1-1 line bay intra communication **b** Line bay RBD

as shown in Fig. 11a. The line bay and also the transformer and bus bays have a single Ethernet switch and all IEDs are connected to its corresponding Ethernet switch and thus can be modeled by using the star architecture. The corresponding RBD model of the intra-bay communication for T1-1 line bay is depicted in Fig. 11b and its reliability can be formally analyzed in HOL as follows:

Definition 7.8 ⊢ ∀ p t TS MU CIED ESW PIED1 PIED2.
T1-1_line_bay_RBD p TS MU CIED ESW PIED1 PIED2 t =
star_arch_RBD p [TS;MU;CIED] ESW [[PIED1;PIED2]] t

We can easily formally verify the reliability of the intra-bay communication network in a line bay using Theorem 7.2 in HOL as follows:

Theorem 7.6 ⊢ ∀ TS MU CIED ESW PIED PIED2 C_TS C_MU C_ESW
C_PIED1 C_PIED2 p t.
time_positive t ∧
rbd_conds (FLAT
(two_dim_rel_event_list p
([[TS;MU;CIED]]++[[ESW]]++[[PIED1;PIED2]]) t)) ∧
two_dim_exp_dist_list p
([[TS;MU;CIED]]++[[ESW]]++[[PIED1;PIED2]])
([[C_TS;C_MU;C_CIED]]++[[C_ESW]]++[[C_PIED1;C_PIED2]]) ⇒
(prob p (T1-1_line_bay_RBD p TS MU CIED ESW PIED t) =
exp (-(C_TS + C_MU + C_CIED + C_ESW) * t) *
(1 - (1 - exp (-C_PIED1 * t))*(1 - exp (-C_PIED2 * t))))

Similarly, we can also formally analyze the reliability of the intra-bay communication in the T1-1 transformer and bus bays. The transformer bay has two MUs and two separate TS devices connected to each of them (Fig. 12a. The RBDs of the transformer bay, as shown in Fig. 12c, is similar to the line bay RBD except that there are two TS sources and MUs connected in series. Similarly, the bus bay configuration is similar to the transformer bay except that the TU is shared among MUs and also the RBD of the T1-1 bus bay contains one TS RBD block instead of two connected in series. Therefore, we only present the formal RBD model of the transformer bay and the formal verification of its corresponding reliability expression in HOL as follows:

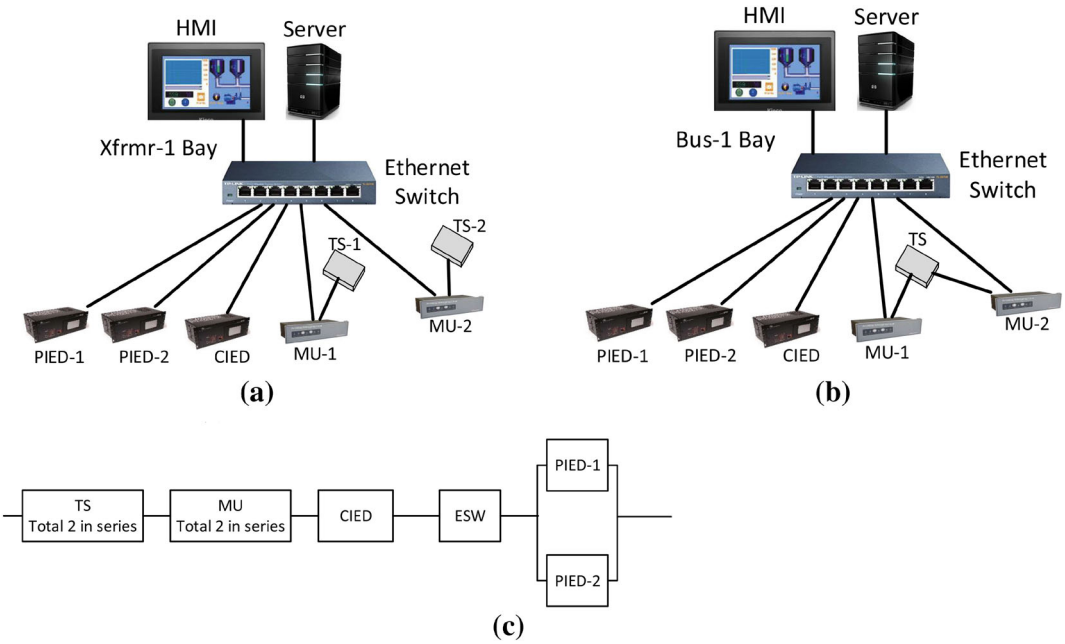


Fig. 12. a Star architecture for T1-1 transformer bay intra communication. b Star architecture for T1-1 bus bay. **c** Transformer bay RBD

Definition 7.9 ⊢ ∀ p t TS1 TS2 MU1 MU2 CIED ESW PIED1 PIED2.
T1-1_Xfrmr_bay_RBD p TS1 TS2 MU1 MU2 CIED ESW PIED1 PIED2 t =
star_arch_RBD p [TS1;TS2;MU1;MU2;CIED] ESW [[PIED1;PIED2]] t

Theorem 7.7 ⊢ ∀ TS1 TS2 MU1 MU2 CIED ESW PIED1 PIED2 C_TS C_MU C_ESW C_PIED1 C_PIED2 p t.
time_positive t ∧
rbd_conds (FLAT
  (two_dim_rel_event_list p
    ([[TS1;TS2;MU1;MU2;CIED]]++[[ESW]]++[[PIED1;PIED2]]) t)) ∧
two_dim_exp_dist_list p
  ([[TS1;TS2;MU1;MU2;CIED]]++[[ESW]]++[[PIED1;PIED2]])
  ([[C_TS1;C_TS2;C_MU1;C_MU2;C_CIED]]++[[C_ESW]]++[[C_PIED1;C_PIED2]]) ⇒
 (prob p
   (T1-1_Xfrmr_bay_RBD p TS1 TS2 MU1 MU2 CIED ESW PIED1 PIED2 t) =
  exp (-(C_TS1 + CS_TS2 + C_MU1 + C_MU2 + C_CIED + C_ESW) * t) *
  (1 - (1 - exp (-C_PIED1 * t))*(1 - exp (-C_PIED2 * t))))

It is quite evident that the analysis of the T1-1 smart substation intra-bay communication network is quite straight forward due to our generic formalization of Ethernet communication architectures described in Sect. 7.1.

7.2.2. Inter-bay communication network

In the inter-bay communication, we want to analyze the reliability of the T1-1 smart substation communication network among its bays, such as line, transformer and bus. For instance, if the T1-1 substation inter-bay communication is realized by cascaded Ethernet architecture then we can model the network as a serial connection of Ethernet switches, which are connected to T1-1 line, transformer and bus bays, as shown in Fig. 13a. To construct the RBD of the T1-1 inter-bay communication network forming a cascaded architecture, we consider one PIED, one CIED and the Ethernet switches that must be functional in order to establish the communication network. The reliability expression of the inter-bay communication architecture exhibiting cascaded architecture, shown in Fig. 13b, can be formally verified by specializing the $PCDs\_r$, $PCDs\_l$ and ESW_list random variable lists, in Theorem 7.1:

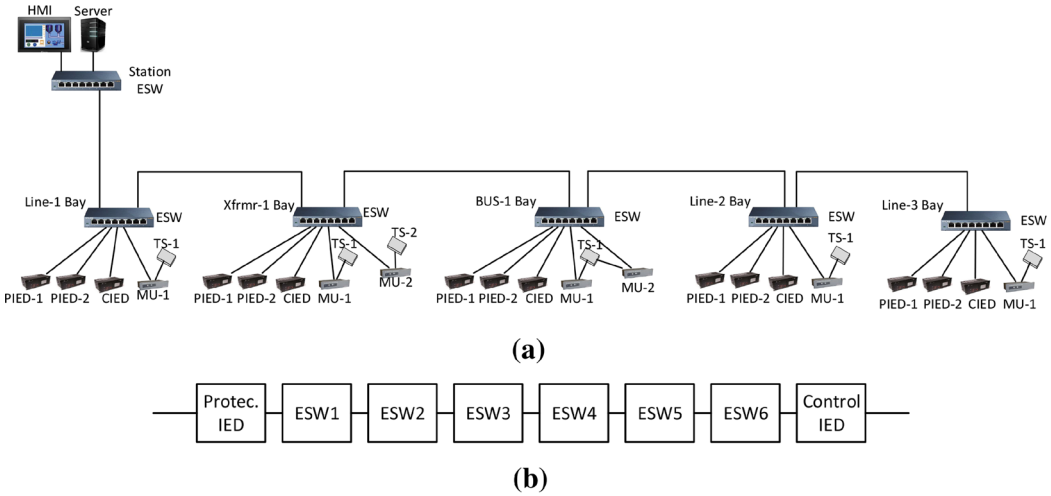


Fig. 13. a T1-1 inter-bay cascaded architecture. **b** RBD Model

Theorem 7.8 ⊢ ∀ PIED CIED ESW1 ··· ESW6
C_PIED C_CIED C_ESW1 ··· C_ESW6 p t.
time_positive t ∧
rbd_conds (rel_event_list p [PIED;ESW1;...;ESW6;CIED] t) ∧
exp_dist_list p
([PIED;ESW1;...;ESW6;CIED])
([C_PIED;C_ESW1;...;C_ESW6;C_CIED]) ⇒
(prob p (casc_series_RBD p [PIED] [ESW1; ...; ESW6] [CIED] t) =
exp (-list_sum ([C_PIED;C_ESW1;...;C_ESW6;C_CIED]) * t))

Similarly, if the T1-1 inter-bay communication architecture is exhibiting the redundant ring model, then its network configuration can be modeled by connecting the redundant Ethernet switches with the line, transformer and bus bays, as shown in Fig. 14a. To construct its RBD model, we can observe that all the IEDs are connected to two redundant Ethernet switches and also ESWs are in parallel for both PIEDs. Moreover, each IED is connected with two redundant ring configurations, and they should be connected in parallel, as shown in Fig. 14a. We can easily analyze its reliability by specializing the $PCDs\_r$, $PCDs\_l$ and ESW_list variables in the redundant ring architecture Theorem 7.5 as follows:

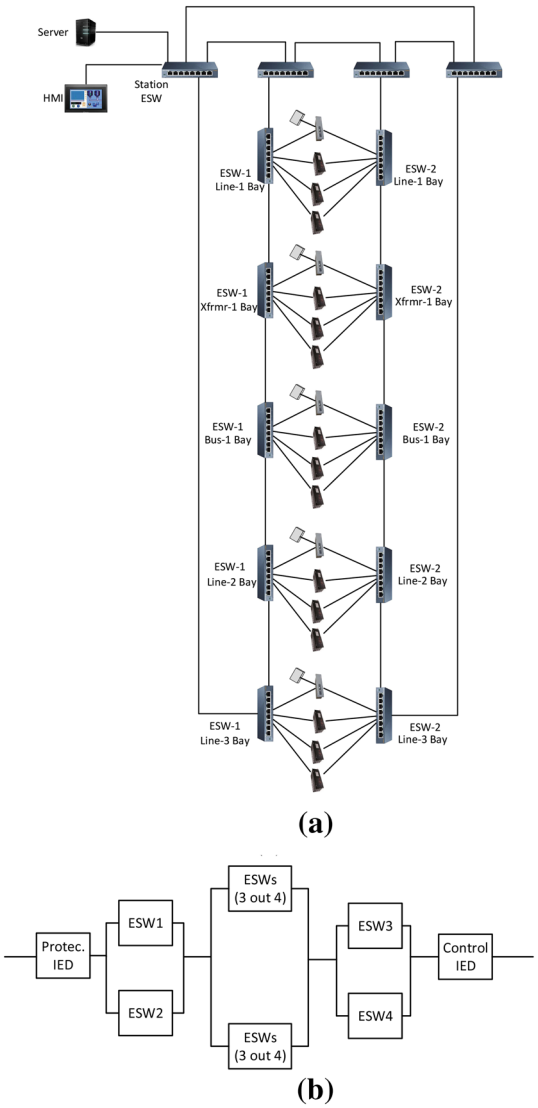


Fig. 14. a T1-1 inter-bay redundant ring architecture. **b** RBD Model

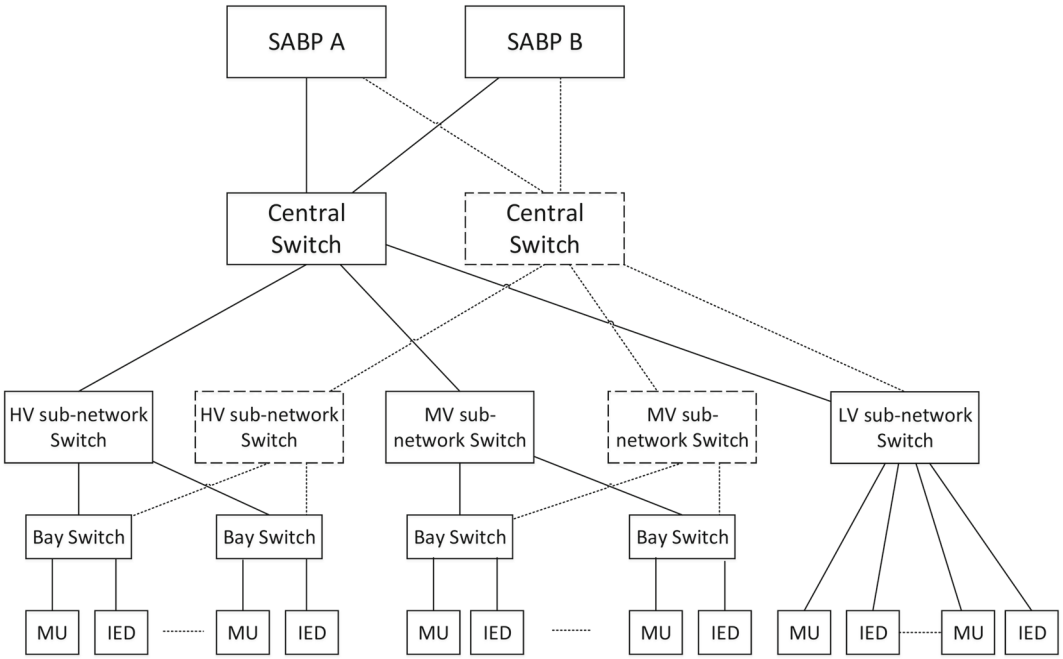


Fig. 15. General design of the SABP communication network

Theorem 7.9 ⊢ ∀PIED CIED ESW1 ⋯ ESW4 C_PIED C_CIED C_ESW1 ⋯ C_ESW4 C_Main_ESW1 C_Main_ESW2 p p' t.

```
time_positive t ∧
rbd_conds
   (FLAT (K_out_N_struct_list p [X_bino; X_bino] 3 4::
    two_dim_rel_event_list p
      ([[PIED]; [ESW1; ESW2]; [ESW3; ESW4]; [CIED]]) t)) ∧
binomial_conds p' X_bino ESWs 3 4 [X_bino;X_bino] t
exp_dist p ESWs C_ESWs ∧
two_dim_exp_dist_list p
   ([[PIED]; [ESW1; ESW2]; [ESW3; ESW4]; [CIED]])
   ([[C_PIED]; [C_ESW1;C_ESW2]; [C_ESW3;C_ESW4];[C_CIED]]) ⇒
(prob p
  (redund_star-ring_RBD p [PIED] [[CIED]] [[ESW1;ESW2];[ESW3;ESW4]]
    X_bino 3 4 t) =
(1 -
    (1 - sum (3,SUC 4 - 3)
      (λi. &binomial 4 x * exp (-C_ESWs * t) pow i *
      (1 - exp (-C_ESWs * t)) pow (4 - i)))) pow 2 *
(list_prod of
   (λa. 1 - list_prod
     (one_minus_list (exp_func_list a t))))
      ([[C_PIED];[C_ESW1;C_ESW2];[C_ESW3;C_ESW4];[C_CIED]])
```

where PIED, X_ESW1,X_ESW2,X_ESW3,X_ESW4, and CIED are the random variables associated with the substation components and C_CIED,C_ESW1,C_ESW2,C_ESW3, C_ESW4, and C_PIED are their corresponding failure rates.

The distinguishing features of the formally verified Theorems 7.6–7.9, compared to the reliability analysis of the smart grid substation communication architectures in [KS09], include their generic nature, i.e., all variables are universally quantified and thus can be specialized to obtain the reliability of the T1-1 smart substation intra-bay and inter-bay communication network for any given failure rate (Sect. 9). The correctness of the results are guaranteed due to the involvement of a sound theorem prover in their verification, which ensures that all required assumptions for the validity of the result are accompanying the theorems. The proofs of Theorems 7.6–7.9 took only a few lines of HOL4 code [Ahm19], and were based on real-analysis based manual reasoning only. This is due to the generically verified results presented in Theorems 7.1–7.5, which took a total of 1200 lines of HOL4 code. This clearly indicates the usefulness of our core formalizations [Ahm19].

In addition to accurate reliability analysis of the communication networks in smart substations, it is essentially important to conduct their accurate failure analysis in order to understand the impact of substation components failure upon the communication networks. For this purpose, we present the FT-based formal failure analysis of a substation area backup protection communication network in the next section.

8. Formal failure analysis of substation area backup protection communication network

A smart area backup protection (SABP) communication network [CZXH13] is a substation protection system based on a high performance communication network, which can transmit the substation area information, such as the currents and voltages of local electric elements and each circuit breaker's switching status. The general design of a SABP communication network, as shown in Fig. 15, illustrates the interconnection of high-voltage (HV), medium-voltage (MV) and low-voltage (LV) sub-networks with the bay-level Ethernet switches, which are in turn establishing Ethernet communication network among intelligent devices, such as IEDs and MUs. All Ethernet switches in the entire SABP communication network are redundant. In other words, if any Ethernet switch in the SABP communication network fails, then the network is still in the functional state due to the presence of a redundant Ethernet switch. It can also be observed that the SABP communication network is closely related to the redundant star architecture since IEDs and MUs are connected directly with their respective Ethernet switches.

To conduct the FT-based formal failure analysis of the SABP communication network, we consider a 220kV substation [CZXH13] consisting of four kinds of bays: (i) Line bays, composed of four 220kV transmission lines ($L1\_220$-$L4\_220$), five 110kV transmission lines ($L1\_110$-$L5\_110$), and fourteen 10kV feeder lines ($L1\_10$-$L14\_10$); (ii) Bus-bar bays, including 220kV, 110kV and 10kV buses, named as Bus_220, Bus_110 and Bus_10; (iii) Transformer bays, consisting of T1 and T2, while each transformer bay includes three merging units (MUs) and three IEDs coming from its high voltage, medium voltage and low voltage sides; and (iv) Bus-couple bay and section switch bay, named as BC_110, SS_220 and SS_10. A schematic design of the SABP communication network incorporating intelligent sensor devices, such as SWs, for the 220kV smart substation is shown in Fig. 16. The switches *SW1*-SW2, *SW3*-SW4, and *SW5*-*SW9* are used for exchange of data in the 220kV, 110kV, and 10kV sub-networks, respectively. The switch *SW10* is designated as the central switch managing all messages within the smart substation. In Fig. 16, we can observe that the Ethernet switches *SW1* and *SW2* are connected to 220kV bays, *SW3* and *SW4* establish an Ethernet communication network for 110kV bays, *SW5*-SW10 are connected with the 10kV line bays, and *SW9* is connected to the 10kV transformer and Bus bays. All these Ethernet switches in turn establish a communication network with the central Ethernet switch *SW10*. In the construction of the SABP communication network FT, we mainly consider that each Ethernet switch is redundant, for instance, two Ethernet switches *SW1* and *SW1'* are connected with the 220kV line bays.

We now present the formal failure analysis of the SABP communication network using our proposed FT formalization (Sect. 6).

8.1. Formal specification

In order to formalize the SABP communication network in a smart substation, we first present the formal modeling of the failure events [AH16] that are associated with the failures of Ethernet switches connected with the substation bays.

Definition 8.1 ⊢ ∀ p L t.
`fail_event_list p L t = MAP (fail_event p a t) L`

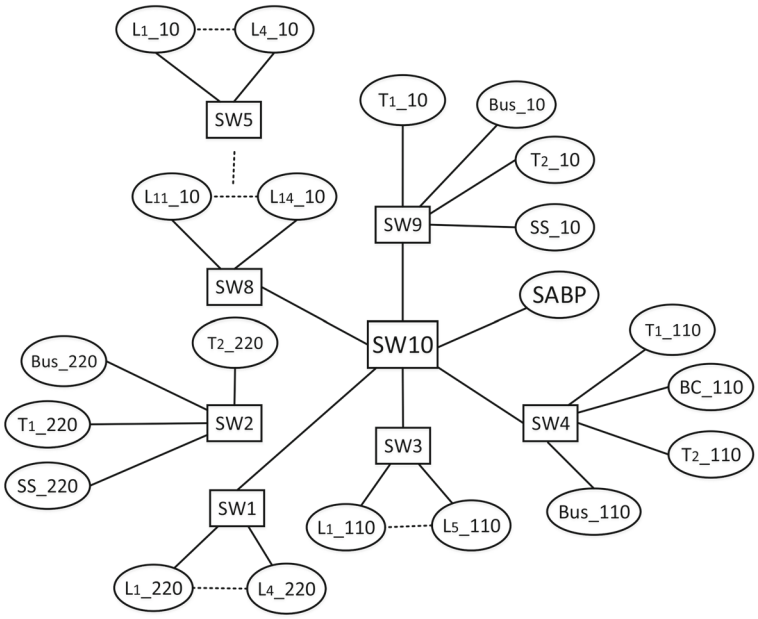


Fig. 16. Schematic architecture design of the SABP communication network for a 220 kV smart substation

The function `fail_event_list` accepts a probability space p, a list of random variables, representing the failure time of individual components, and a real number t, which represents the time index at which the failure of the component occurs. It returns a list of failure events by mapping the function `fail_event` (Definition 7.3) on each element of the given list of random variables representing the failure of all individual components at time t.

The formal definitions of the FT gates, given in Table 3, along with the above definitions can be utilized to formally represent the FT of the SABP communication network in terms of its failure events. In this work, we present the formal failure analysis of the SABP communication network subsystem, as shown in Fig. 17, protecting the 220kV level sub-network. The SABP sub-networks of other levels, like 110kV and 10kV, can also be formally analyzed by following a similar procedure. The FT model of the SABP communication network, as depicted in Fig. 17, is constructed by starting from the top event F representing the failure of the overall network failure. As shown in Fig. 16, the Ethernet switches *SW1* and *SW2* are connected with the SABP sub-network at 220kV level. Thus, the failure of both these switches and the bays connected to them would cause the overall SABP sub-network at 220kV level to fail. Therefore, we connect the AND FT gate with the top event F. We connect two OR FT gates at the input of AND FT gate since each Ethernet switch is redundant, i.e., *SW1, SW1', SW2* and *SW2'*. Lastly, the AND FT gates represented by the symbols *G1*, *G2*, *G3* and *G4* are used to model the behavior that all bays must be in failure to propagate the failure from the basic events. We start formalizing the SABP sub-network FT shown in Fig. 17 from the *G1* sub-FT in HOL as follows:

Definition 8.2 ⊢ ∀ p t X_SW1 X_L1_220 X_L2_220 X_L3_220 X_L4_220.
```
G1_FT_gate p t X_SW1 X_L1_220 X_L2_220 X_L3_220 X_L4_220 =
AND (gate_list (fail_event_list p
    [X_SW1;X_L1_220;X_L2_220;X_L3_220;X_L4_220] t))
```

Similarly, we can also formally define the functions `G2_FT_gate`, `G3_FT_gate` and `G4_FT_gate` representing the AND gates *G2*, *G3*, and *G4* of Fig. 17, respectively, in HOL. It is important to mention that although the Ethernet switches in *G1* and *G2* are identical, they are assigned different random variables. For instance, *SW1* in *G1* is assigned with X_SW1 whereas in *G2* with X'_SW1. Based on these definitions, the SABP communication sub-network FT at 220kV level can be formalized in HOL as follows:

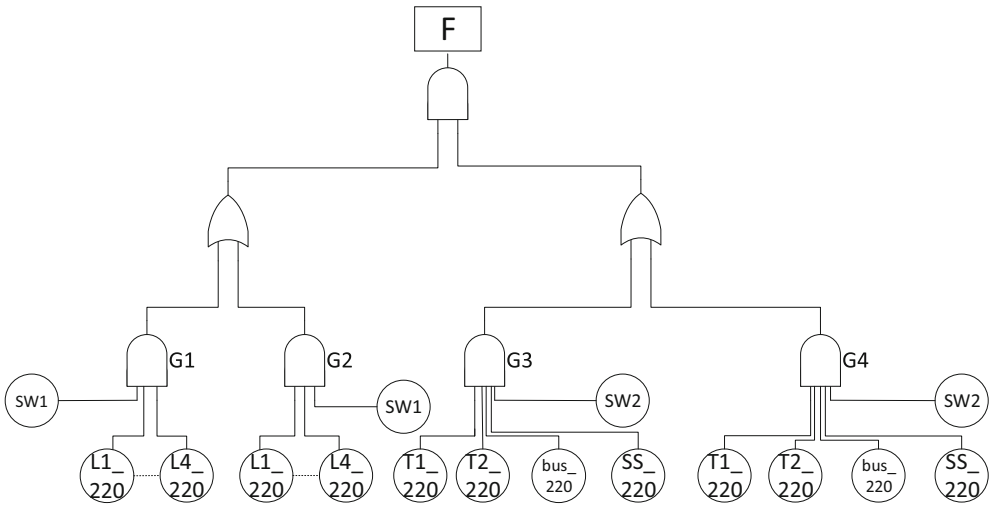


Fig. 17. SABP fault tree of 220 kV level sub-network

Definition 8.3 ⊢ ∀ p t X_SW1 X_L1_220 ··· X_L4_220 X'_SW1 X'_L1_220 ··· X'_L4_220 X_SW2 X_T1_220 X_T2_220 X_BUS_220 X_SS_220 ···.
SABP_FT_220 p t X_SW1 X_L1_220 ··· X'_SS_220 =
FTree p
(AND [OR [G1_FT_gate p t X_SW1 X_L1_220 ··· X_L4_220;
G2_FT_gate p t X'_SW1 X_L1_220 ··· X_L4_220];
OR [G3_FT_gate p t X_SW2 X_T1_220 X_T2_220 X_BUS_220 X_SS_220;
G4_FT_gate p t X'_SW2 X_T1_220 X_T2_220 X_BUS_220
X_SS_220]])

We can obtain the minimal cut sets (MCS) of the above FT model by utilizing some set-theoretic properties, like the distribution of the intersection over the union and the idempotent law of intersection [IEC06].

$C_1 = \{SW1, SW2, L1\_220, \ldots, L4\_220, T1\_220, T2\_220, BUS\_220, SS\_220\}$

$C_2 = \{SW1, SW2', L1\_220, \ldots, L4\_220, T1\_220, T2\_220, BUS\_220, SS\_220\}$

$C_3 = \{SW1', SW2, L1\_220, \ldots, L4\_220, T1\_220, T2\_220, BUS\_220, SS\_220\}$

$C_4 = \{SW1', SW2', L1\_220, \ldots, L4\_220, T1\_220, T2\_220, BUS\_220, SS\_220\}$

There are four MCSs obtained for the SABP FT at 220kV level and the occurrence of each of these MCSs can cause the overall SABP communication sub-network to fail. For instance, if C_1 occurs, due the failures of the Ethernet switches SW1 and SW2, the line bays $L1\_220 - L4\_220$, the transformer bays $T1\_220$ and $T2\_220$, the bus bay bus_220 and the section switch bay SS_220, then the overall SABP sub-network at 220kV will be in failure.

8.2. Formal verification

As already discussed in Sect. 6.3, the PIE principle suffers from complexity issues and may not be an appropriate choice to reason about even sizable systems, like smart grids. Hence, in order to determine the failure probability of the SABP sub-network at 220kV level, we propose to transform the corresponding FT model to its equivalent series–parallel RBD, as shown in Fig. 18, by verifying the following lemma in HOL:

Lemma 2 ⊢ prob_space p ⇒
(SABP_FT_220 p t X_SW1 X_SW1' X_SW2 X_L1_220 ··· X_SS_220 =
rbd_struct p
((series of (λa. parallel (rbd_list a)))
(list_fail_event_list p
[[X_L1_220];[X_L2_220];[X_L3_220];[X_L4_220]; [X_SW1;X_SW1'];
[X_T1_220];[X_T2_220];[X_Bus_220];[X_SS_220];[X_SW2;X_SW2']] t)))

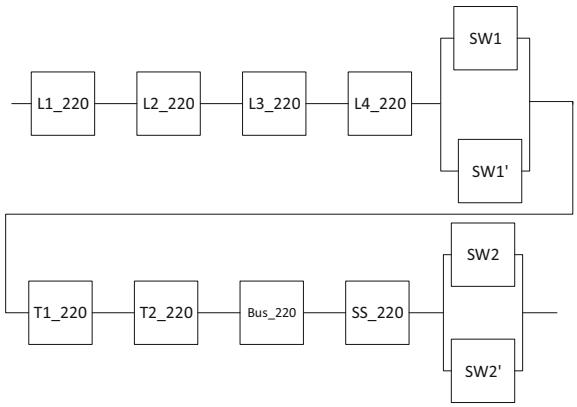


Fig. 18. SABP RBD of 220 kV level sub-network

The function `list_fail_event_list` takes a two-dimensional list of random variables and returns a two-dimensional list of failure events by mapping the function `fail_event_list` (Definition 8.1) on each element of the given list.

By utilizing Lemma 2, we can be formally verify the failure probability of SABP sub-network FT at 220kV level in HOL as follows:

Theorem 8.1 ⊢ ∀ p t X_SW1 X_L1_220 ··· X_SS_220 C_SW1 C_L1_220 ··· C_SS_220.

```
time_positive t ∧
two_dim_exp_dist_list p
  [[X_L1_220];[X_L2_220];[X_L3_220];[X_L4_220];[X_SW1;X_SW1'];
   [X_T1_220];[X_T2_220];[X_BUS_220];[X_SS_220];[X_SW2;X_SW2']]
  [[C_L1_220];[C_L2_220];[C_L3_220];[C_L4_220];[C_SW1;C_SW1'];
   [C_T1_220];[C_T2_220];[C_BUS_220];[C_SS_220];[C_SW2;C_SW2']] ∧
rbd_conds p
  (FLAT (list_fail_event_list p
   [[X_L1_220];[X_L2_220];[X_L3_220];[X_L4_220];
    [X_SW1;X_SW1'];[X_T1_220];[X_T2_220];[X_BUS_220];
    [X_SS_220];[X_SW2;X_SW2']] t)) ⇒
(prob p
   (SABP_FT_220 p t X_SW1 X_SW1' X_SW2 X_L1_220 ··· X_SS_220) =
(1 - (1 - (1 - exp (-C_L1_220 * t)))) *
(1 - (1 - (1 - exp (-C_L2_220 * t)))) *
(1 - (1 - (1 - exp (-C_L3_220 * t)))) *
(1 - (1 - (1 - exp (-C_L4_220 * t)))) *
(1 - (1 - (1 - exp (-C_SW1 * t))) * (1 - (1 - exp (-C_SW1'*t)))) *
(1 - (1 - (1 - exp (-C_T1_220 * t)))) *
(1 - (1 - (1 - exp (-C_T2_220 * t)))) *
(1 - (1 - (1 - exp (-C_BUS_220 * t)))) *
(1 - (1 - (1 - exp (-C_SS_220 * t)))) *
(1 - (1 - (1 - exp (-C_SW2 * t))) * (1 - (1 - exp (-C_SW2' * t)))))
```

The assumptions are similar to the ones described in Theorem 7.8. The conclusion of Theorem 8.1 models the failure probability expression of the SABP sub-network FT at 220kV level. The function `one_minus_exp` accepts a list of failure rates and returns a one minus list of exponentials and the function `one_minus_exp_prod` accepts a two dimensional list of failure rates and returns a list with one minus product of one minus exponentials of every sub-list. For example, `one_minus_exp_prod` $[[c1;\ c2;\ c3];\ [c4;\ c5];\ [c6;\ c7;\ c8]]\ t = [1 - ((1 - e^{-(c1)t}) * (1 - e^{-(c2)t}) * (1 - e^{-(c3)t}));\ (1 - (1 - e^{-(c4)t}) * (1 - e^{-(c5)t}));\ (1 - (1 - e^{-(c6)t}) * (1 - e^{-(c7)t}) * (1 - e^{-(c8)t}))]$.

The proof of Theorem 8.1 utilizes Lemma 2, Theorem 5.4 and some axioms of probability, such as the probability of the empty set Φ is 0 and the complete space Ω is 1.

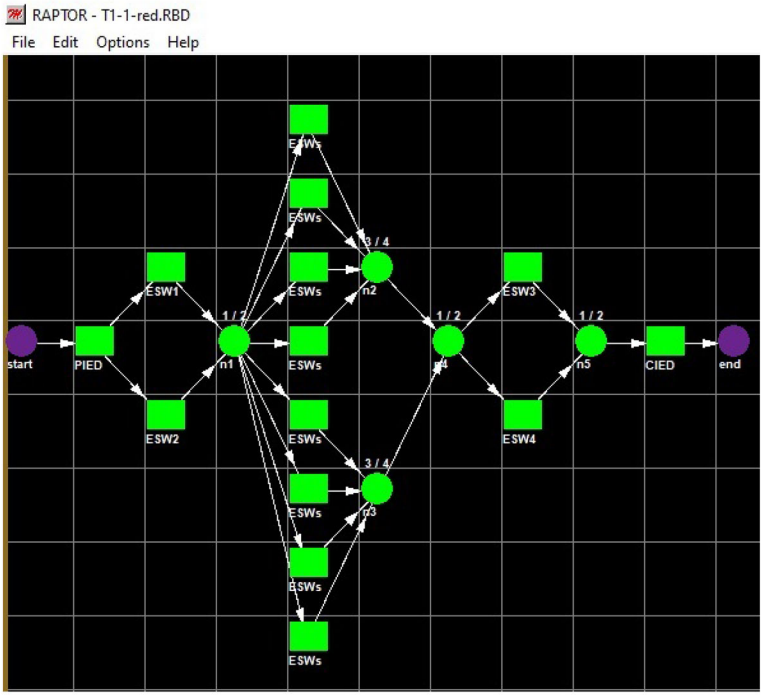
In summary, the major benefits of the above-mentioned formal failure analysis of the SABP communication sub-network operating at the 220kV level smart substation are as follows: (i) It provides formally verified failure probability results; (ii) The substation design engineers may use our formally verified results while accurately designing and analyzing the SABP supported communication networks for any smart substation; and (iii) We can accurately achieve an acceptable level of safety as per the recommendations of safety standards, such as IEC 61850.

9. Evaluation

In order to facilitate the utilization of our formally verified results for industrial design engineers for their reliability and failure probabilities assessment, we have also developed a set of HOL SML functions to automate the simplification step of Theorems 7.1-8.1. Also, these functions have the ability to numerically compute the reliability and the failure probability by evaluating the exponential functions for a given list of failure rates corresponding to the smart substation network components. For instance, consider the SML function `smart_grid_RBD_EVAL` that takes a list of failure rates, time index and a specific theorem, say Theorem 7.1, to obtain the simplified and numerically computed reliability as follows:

```
Under the following assumptions:
prob_space p ∧
mutual_indep p (FLAT (K_out_N_struct_list p [X_bino;X_bino] 3 4::
 two_dim_rel_event_list p [[PIED];[ESW1;ESW2];[ESW3;ESW4];[CIED]] 10.0)) ∧
in_events p (FLAT (two_dim_rel_event_list p
   [[PIED];[ESW1;ESW2];[ESW3;ESW4];[CIED]] 10.0)) ∧
(∀x. x < SUC 4 ⇒
  in_events p (binomial_event_list [X_bino; X_bino] x)) ∧
(∀x.
  distribution p X_bino Normal (&x) =
  &binomial 4 x * Reliability p ESWs 10.0 pow x *
  (1 - Reliability p ESWs 10.0) pow (4 - x)) ∧
exp_dist p ESWs (1.0 / 50.0) ∧
two_dim_exp_dist_list p
  [[PIED];[ESW1;ESW2];[ESW3;ESW4];[CIED]]
  [[1.0/150.0];[1.0/50.0;1.0/50.0];[1.0/50.0;1.0/50.0];
   [1.0/150.0]]
Simplified Reliability Expression of Redundant Star Ring:
prob p
(rbd_struct p (parallel
   (rbd_list (K_out_N_struct_list p [X_bino; X_bino] 3 4))) ∩
rbd_struct p (series
   (MAP (λa. parallel (rbd_list a)) (two_dim_rel_event_list p
    [[PIED]; [ESW1; ESW2]; [ESW3; ESW4]; [CIED]] 10)))) =
(1 -
  (1 - (4 * (exp (-1 / 5) * (exp (-1 / 5) * exp (-1 / 5))) *
  (1 - exp (-1 / 5)) +
exp (-1 / 5) * (exp (-1 / 5) * (exp (-1 / 5) * exp (-1/5))))) *
(1 -
 (4 * (exp (-1 / 5) * (exp (-1 / 5) * exp (-1/5))) *
 (1 - exp (-1 / 5)) +
exp (-1 / 5) * (exp (-1 / 5) * (exp (-1 / 5) * exp (-1/5)))))) *
(exp (-1 / 15) * ((1 - (1 - exp (-1 / 5)) * (1 - exp (-1/5))) *
((1 - (1 - exp (-1 / 5))*(1 - exp (-1/5)))*exp (-1/15))))

Numerically Computed Reliability of Redundant Star Ring:
0.799506670088
```



(a)

Final Results

Results from 100 run(s):

PARAMETER	MEAN	MIN	MAX	ST DEV
Ao	0.908581767	0.077050603	1.0000000000	0.232959349
MTBDE	>9.085818	0.770506	>10.000000	n/a
MDT (20 runs)	4.570912	0.046033	9.229494	3.228071
MTBM	>5.144012	0.770506	>10.000000	n/a
MRT (89 runs)	0.000000	0.000000	0.000000	0.000000
% Green Time	35.946470	0.061199	100.000000	32.612009
% Yellow Time	54.911707	0.000000	99.938801	35.772408
% Red Time	9.141823	0.000000	92.294940	23.295935
System Failures	0.200000	0	1	0.400000

R(t=10.000000) =0.800000

Average sparing data over 100 run(s):

COMPONENT	START	END	MIN	MAX	# DELAYS
NOT USED					

(b)

Fig. 19. a Redundant star ring architecture model in raptor. b Simulation results

The MTTF of the IEDs are generally quite high compared to the Ethernet switches. In the literature, the MTTF of the IEDs is generally valued as 150 years and Ethernet switch as 50 years [ATGH15, KS09]. In other words, if we statistically test 150 IEDs for a year than only one will fail compared to observing a single failure when testing 50 Ethernet switches for a year. To determine the accuracy of our proposed analysis approach, we also analyze the redundant ring architecture RBD of the T1-1 smart substation inter-bay communication network using the Raptor reliability analysis tool [MCGM03], which is a well-known RBD software for analyzing reliability, availability and maintainability of real-world systems. The RBD model and the reliability result obtained from the Raptor at time index $t = 10$ years is shown in Fig. 19. It can be seen that both values obtained from our HOL SML function and the Raptor tool is approximately equivalent. We also made a comparison among both these approaches for a range of time values from 1 upto 100 years and the results are represented by a histogram plot in Fig. 20. It can be observed that there is a slight difference in the values that may be due to the fact that the analysis in the Raptor tool is based on Monte Carlo simulation, which is a sampling based technique.

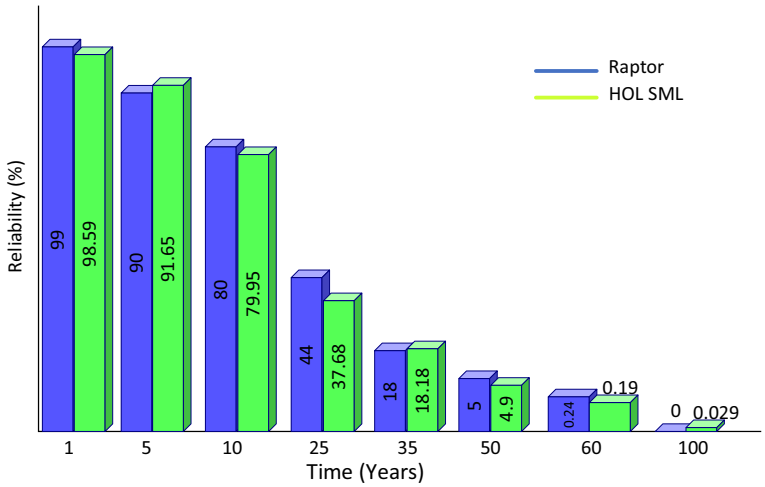


Fig. 20. Reliability of T1-1 redundant star-ring architecture for inter-bay communication obtained from raptor and HOL SML

On the other hand, our proposed approach evaluates the exact reliability expression consisting of negative exponential functions. Similarly, the HOL SML function `smart_grid_RBD_EVAL` can be used for any values of the failure rates and works equally well for Theorems 7.1–8.1.

We believe that the formally verified reliability theorems, presented in Sects. 7–8, provide useful insights that can help reliability design engineers to compare and correct their estimated reliability results, which are either obtained through manual manipulation or computer simulation. For instance, the reliability of these smart substations increases with the increase in the number of redundant Ethernet switches but it also raises the cost and complexity especially for large substations. In this context, our formalization would enable reliability design engineers to accurately determine the least number of Ethernet switches, which are required to achieve a desired level of reliability in a smart grid substation.

10. Conclusions

The accuracy of reliability and failure analysis of Ethernet communication networks in a smart substation has become a dire need to ensure safe and secure operation of smart grids, where an incorrect failure estimate may lead to blackouts and power outages and thus huge financial losses and even the loss of human life in the worst-case scenario. In this paper, we presented a generic framework for formal RBD and FT-based reliability and failure analysis of Ethernet communication networks in a smart substation within the sound core of the HOL4 theorem prover. We plan to build a GUI to capture any RBD and FT model, like smart grid communication RBD and FT, from the user and return the formally verified failure probability expression of the given system, by using the HOL4 theorem prover that is running seamlessly underlying this GUI. This would bring great benefits to non-HOL users, like industrial design engineers, in many respects. For instance, it can be used to certify the results estimated by the design engineers and then provide an opportunity at the design stage to correct this estimated result, which are traditionally either obtained through manual manipulation or computer simulations.

In this paper, we mainly focus on building a sound methodology for conducting the accurate reliability and failure assessment of Ethernet communication network in a smart substation using HOL theorem proving. However, our methodology can be further extended by formalizing the *Reliability Importance* (RI) [AB03] concept, which requires RBD and FT-based analysis as a prerequisite. RI is an important concept in reliability engineering typically used for identifying critical system components obtained by ranking all components based on their probability of failure. As future work, we plan to formalize the notion of reliability importance in HOL in order to identify the critical components in a smart grid substation communication network.

References

[AB03] Andrews JD, Beeson S (2003) Birnbaum's measure of component importance for noncoherent systems. IEEE Trans Reliab, 52(2):213–219

[ABM10] Adamiak M, Baigent D, Mackiewicz R (2010) IEC 61850 communication networks and systems in substations: an overview for users. In: Technical report, SISCO Systems

[AH15] Ahmad W, Hasan O (2015) Towards formal fault tree analysis using theorem proving. In: Kerber M, Carette J, Kaliszyk C, Rabe F, Sorge V (eds) Conferences on intelligent computer mathematics, volume 9150 of lecture notes in computer science. Springer, Berlin, pp 39–54

[AH16] Ahmad W, Hasan O (2016) Formalization of fault trees in higher-order logic: a deep embedding approach. In: Fränzle M, Kapur D, Zhan N, (eds) Dependable software engineering: theories, tools, and applications, volume 9984 of lecture notes in computer science. Springer, Berlin, pp 264–279

[Ahm17] Ahmad W (2017) Formal dependability analysis using higher-order-logic theorem proving. Ph.D. Thesis, National University of Sciences and Technology, Islamabad, Pakistan

[Ahm19] Ahmad W (2019) Formal dependability analysis of ethernet based communication networks in a smart substation, proof script. http://save.seecs.nust.edu.pk/smartgrids/

[AHT16] Ahmed W, Hasan O, Tahar S (2016) Formalization of reliability block diagrams in higher-order logic. J Appl Logic 18:19–41

[AHTH14] Ahmad W, Hasan O, Tahar S, Hamdi MS (2014) Towards the formal reliability analysis of oil and gas pipelines. In Watt SM, Davenport JH, Sexton AP, Sojka P, Urban J (eds) Conferences on intelligent computer mathematics, volume 8543 of lecture notes in computer science. Springer, Berlin, pp 30–44

[AHTH18] Ahmad W, Hasan O, Tahar S, Hamdi MS (2018) Formal reliability analysis of oil and gas pipelines. Proc Inst Mech Eng Part O J Risk Reliab 232(3):320–334

[ASE] ASENT, RBD Analysis Tool. https://www.raytheoneagle.com/asent/rbd.htm". Accessed on 05 Sept 2019.

[ATGH15] Ali I, Thomas MS, Gupta S, Hussain SMS (2015) IEC 61850 Substation communication network architecture for efficient energy system automation. Energy Technol Pol, 2(1):82–91

[BA92] Bilintion R, Allan RN (1992) Reliability evaluation of engineering system. Springer, Berlin

[BCK+09] Bozzano M, Cimatti A, Katoen J, Nguyen V, Noll T, Roveri M (2009) The compass approach: correctness, modelling and performability of aerospace systems. In: Buth B, Rabe G, Seyfarth T (eds) Computer safety, reliability, and security, volume 5775 of lecture notes in computer science. Springer, Berlin, pp 173–186

[BD83] Benthem JV, Doets K (1983) Higher-order logic. In: Gabbay D, Guenthner F (eds) Handbook of philosophical logic, IGI Global, Pennsylvania, pp 275–329

[BHJ14] Babu S, Hilber P, Jürgensen JH (2014) On the status of reliability studies involving primary and secondary equipment applied to power system. In: Probabilistic methods applied to power systems, July 7–10. IEEE, Durham, pp 1–6

[Bir04] Birolini A (2004) Reliability engineering: theory and practice. Springer, Berlin

[BJ18] Bistouni F, Jahanshahi M (2018) Determining the reliability importance of switching elements in the shuffle-exchange networks. Int J Parallel Emerg Distrib Syst 4:1–29

[BK08] Baier C, Katoen JP (2008) Principles of model checking. MIT Press, New York

[Bro07] Brown CE (2007) Automated reasoning in higher-order logic: set comprehension and extensionality in Church's type theory. College Publications, Charleston

[BTF05] Blackett AW, Teachman ME, Forth BJ (2005) Communications architecture for intelligent electronic devices. US Patent 6,944,555

[Che14] Chen SG (2014) Reduced recursive inclusion–exclusion principle for the probability of union events. In: International conference on industrial engineering and engineering management, Dec 9–12. IEEE, Bandar Sunway, pp 11–13

[Chu40] Church A (1940) A formulation of the simple theory of types. J Symb Logic 5:56–68

[CZXH13] Chen L, Zhang K, Xia Y, Hu G (2013) Structure design and performance analysis of substation area backup protection communication network in smart substation. Przeglkad Elektrotechniczny 89(5):182–186

[EHTA11] Elleuch M, Hasan O, Tahar S, Abid M (2011) Formal analysis of a scheduling algorithm for wireless sensor networks. In: Qin S, Qiu Z (eds) Formal engineering methods, volume 6991 of lecture notes in computer science. Springer, Berlin, pp 388–403

[Fit96] Fitting F (1996) First-order logic and automated theorem proving. Springer, New York

[GB06] Gallasch GE, Billington J (2006) A parametric state space for the analysis of the infinite class of stop-and-wait protocols. In: Valmari A (ed) Model checking software, volume 3925 of lecture notes in computer science. Springer, Berlin, pp 201–218

[GM93] Gordon MJC, Melham TF (1993) Introduction to HOL a theorem proving environment for higher-order logic. Cambridge Press, Cambridge

[Gor89] Gordon MJC (1989) Mechanizing programming logics in higher-order logic. In: Birtwistle G, Subrahmanyam PA (eds) Current trends in hardware verification and automated theorem proving, Springer, Berlin, pp 387–439

[Gro] CLA Group. Configuration benchmarks library (CLib). Aralia Fault Trees, IT - University of Copenhagen. https://www.itu.dk/research/cla/externals/clib/. Accessed on 5 Sept 2019

[Har96] Harrison J (1996) Formalized mathematics. Technical report 36, Turku Centre for Computer Science, Finland

[Har09] Harrison J (2009) Handbook of practical logic and automated reasoning. Cambridge University Press, Cambridge

[HH11a] Hajian-Hoseinabadi H (2011) Impacts of automated control systems on substation reliability. IEEE Trans Power Deliv 26(3):1681–1691

[HH11b] Holzl J, Heller A (2011) Three chapters of measure theory in Isabelle/HOL. In: van Eekelen M, Geuvers H, Schmaltz J, Wiedijk F (eds) Interactive theorem proving, volume 6172 of lecture notes in computer science. Springer, Berlin, pp 135–151

[HK81] Henley EJ, Kumamoto H (1981) Reliability engineering and risk assessment, volume 568. Prentice-Hall Englewood Cliffs, New Jersey

[HT07] Hasan O, Tahar S (2007) Formalization of continuous probability distributions. In: Pfenning F (ed) Automated deduction, volume 4603 of lecture notes in artificial intelligence. Springer, Berlin, pp 3–18

[HT08] Hasan O, Tahar S (2008) Performance analysis of ARQ protocols using a theorem prover. In: Performance analysis of systems and software, April 20–22, IEEE, Austin, pp 85–94
[HT14] Hasan O, Tahar S (2014) Encyclopedia of information science and technology, chapter formal verification methods, IGI Global, Pennsylvania, pp 7162–7170
[Hur03] Hurd J (2003) Formal verification of probabilistic algorithms. Ph.D. Thesis, University of Cambridge, Cambridge
[IEC] IEC 61850. Embedded software development and verification to IEC 61508 with formal methods. https://www.eschertech.com/standards/iec61508.php. Accessed 5 Sept 2019
[IEC06] International Electrotechnical Commission (IEC) (2006), 61025 Fault tree analysis, https://webstore.iec.ch/publication/4311. Accessed 5 Sept 2019
[KATH13] Khurram A, Ali H, Tariq A, Hasan O (2013) Formal reliability analysis of protective relays in power distribution systems. In: Pecheur C, Dierkes M (eds) Formal methods for industrial critical systems, volume 8187 of lecture notes in computer science. Springer, Berlin, pp 169–183
[KLB+19] Kumar P, Lin Y, Bai G, Paverd A, Dong JS, Martin A (2019) Smart grid metering networks: a survey on security, privacy and open research issues. IEEE Commun Surv Tutor 21(3):2886–2927
[KNP10] Kwiatkowska M, Norman G, Parker D (2010) Probabilistic model checking for systems biology. In: Iyengar MS (ed) Symbolic systems biology: theory and methods, Jones and Bartlett, Burlington, pp 31–59
[KS09] Kanabar MG, Sidhu TS (2009) Reliability and availability analysis of IEC 61850 based substation communication architectures. In: Power and energy society general meeting, July 26–30. IEEE, Calgary, pp 1–7
[KV10] Kleyner A, Volovoi V (2010) Application of petri nets to reliability prediction of occupant safety systems with partial detection and repair. Reliab Eng Syst Saf 95(6):606–613
[Mac06] Mackiewicz RE (2006) Overview of IEC 61850 and benefits. In: Power systems conference and exposition. IEEE, pp 623–630
[MCGM03] Murphy K, Carter C, Grimes E, Malerich A (2019) RAPTOR 7.0 Tutorial Workbook, http://raptorddr.com/pb/wp-content/uploads/2017/01/Raptor-User-Manual-20130607.pdf Accessed 5 Sept 2019
[MHT11] Mhamdi T, Hasan O, Tahar S (2011) On the formalization of the lebesgue integration theory in HOL. In: Kaufmann M, Paulson LC (eds) Interactive theorem proving, volume 6172 of lecture notes in computer science. Springer, Berlin, pp 387–402
[Mil77] Milner R (1977) A theory of type polymorphism in programming. J Comput Syst Sci 17:348–375
[MPL11] Martins J, Platzer A, Leite J (2011) Statistical model checking for distributed probabilistic-control hybrid automata with smart grid applications. In: Qin S, Qiu Z (eds) Formal methods and software engineering, volume 6991 of lecture notes in computer science. Springer, Berlin, pp 131–146
[MSN03] MSNBC. Blackout Costs N.Y. City $1 Billion, (2003) http://www.andersoneconomicgroup.com/LinkClick.aspx?link=upload/Doc298.pdf. Accessed 5 Sept 2019
[Nar05] Narasimhan K (2005) Reliability engineering: theory and practice. TQM Mag 17(2):209–210
[NWH12] Niyato D, Wang P, Hossain E (2012) Reliability analysis and redundancy design of smart grid wireless communications system for demand side management. IEEE Wireless Commun 19(3):38–46
[OLOV18] Oliva J, Llanes J, Ojeda M, Valle A (2018) Advanced combinatorial method for solving complex fault trees. Anna Nucl Energy 120:666–681
[OS07] Ortmeier F, Schellhorn G (2007) Formal fault tree analysis—practical experiences. Electron Notes Theor Comput Sci 185:139–151
[Pau96] Paulson LC (1996) ML for the working programmer. Cambridge University Press, Cambridge
[PWHR11] Palin R, Ward D, Habli I, Rivett R (2011) ISO 26262 safety cases: compliance and assurance. In: International conference on system safety, Sep 20–22. IEEE, Birmingham, pp 1–6
[Rel] ReliaSoft. http://www.reliasoft.com/. Accessed 5 Sept 2019
[Rel15] ReliaSoft. System analysis reference: reliability, availability and optimization. Technical report technical report, ReliaSoft Corporation, USA, (2015) http://www.synthesisplatform.net/references/System_Analysis_Reference.pdf. Accessed 5 Sept 2019
[RI09] Rose L, Ivaldi G (2009) Ring rapid spanning tree protocol, US Patent 7,564,779
[RXXZ10] Robidoux R, Xu H, Xing L, Zhou M (2010) Automated modeling of dynamic reliability block diagrams using colored petri nets. IEEE Trans Syst Man Cybern Part A Syst Hum 40(2):337–351
[SDC+13] Signoret JP, Dutuit Y, Cacheux PJ, Folleau C, Collas S, Thomas P (2013) Make your petri nets understandable: reliability block diagrams driven petri nets. Reliab Eng Syst Saf 113:61–75
[Tri02] Trivedi KS (2002) Probability and statistics with reliability, queuing and computer science applications. Wiley, Hoboken
[WH13] Wäfler J, Heegaard PE (2013) A combined structural and dynamic modelling approach for dependability analysis in smart grid. In: Symposium on applied computing, Mar 18–22. ACM, Coimbra, pp 660–665
[YCPV13] Yu R, Chen Y, Pan J, Vesel RW (2013) Generic reliability evaluation method for industrial grids with variable frequency drives. Energy Power Eng 5(04):83
[YZN+12] Yuksel E, Zhu H, Nielson HR, Huang H, Nielson F (2012) Modelling and analysis of smart grid: a stochastic model checking case study. In: International symposium on theoretical aspects of software engineering, July 4–6. IEEE, Beijing, pp 25–32
[ZJLS12] Zeng R, Jiang Y, Lin C, Shen X (2012) Dependability analysis of control center networks in smart grid using stochastic petri nets. IEEE Trans Parallel Distrib Syst 23(9):1721–1730

Received 27 April 2019
Accepted in revised form 23 December 2019 by Jin Song Dong
Published online 27 January 2020