Formal Reliability Analysis of Wireless Sensor Network Data Transport Protocols using HOL

Waqar Ahmed and Osman Hasan
School of Electrical Engineering and Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
Email: {waqar.ahmad,osman.hasan}@seecs.nust.edu.pk

Sofiène Tahar
Department of Electrical and Computer Engineering
Concordia University
Montreal, QC, Canada
Email: tahar@ece.concordia.ca

***Abstract*—In recent times, Wireless Sensor Networks (WSNs) have shown a great potential for monitoring physical or environmental conditions in a variety of safety and financial-critical applications, ranging from medicine to transportation and surveillance. Given the extreme conditions of most of the WSN environments, it is very important to make WSN communication resilient to network failures. Various data transport protocols have been proposed in the literature to serve this purpose. The reliability of these WSN data transport protocols is usually assessed by using Reliability Block Diagrams (RBDs). Traditionally, RBD-based reliability analyses of WSN data transport protocols is done using paper-and-pencil proofs or computer simulations, which cannot ascertain absolute correctness due to their inherent incompleteness. As a complementary approach, we propose to use the higher-order-logic theorem prover HOL to conduct the RBD-based reliability analysis of WSN data transport protocols. In particular, the paper provides a higher-order-logic formalization of series, parallel and parallel-series RBDs. These RBDs are then used to do the formal reliability analysis of the end-to-end (e2e) data transport mechanism, and the Event to Sink Reliable Transport (ESRT) and Reliable Multi-Segment Transport (RMST) data transport protocols.**

I. Introduction

Wireless Sensor Networks (WSNs) [33] are wireless networks consisting of spatially distributed sensors and have been increasingly utilized to monitor and collect field data from surrounding environment (e.g. temperature, humidity) for many safety-critical applications including, home automation, external environmental monitoring and object tracking. To ensure continued transfer of field information to the remote users and make the WSN resilient to network failures, several data transport protocols, such as Event to Sink Reliable Transport (ESRT) [27] and Reliable Multi-Segment Transport (RMST) [31], are developed. Due to the safety-critical nature of some of the WSN applications, such as aircraft control [32] and forest fire detection [34], the reliability analysis of these WSN data transport protocols must be carried out beforehand to ensure the reliable delivery of information from the field to the remote user, where appropriate decisions can be made, to avoid catastrophic events. For instance, the summer 2008 wildfire outbreak in California forests resulted in killing 32 people, injuring at least 34 individuals and destroyed large portions of the forests [15]. A reliable WSN could have been used to avoid these damages.

Reliability Block Diagrams (RBDs) [30], which are graphical structures consisting of blocks and connectors (lines), are commonly utilized to model the behaviour of WSN data transport protocols in a WSN and thus to analyze the effect of network failures on the overall WSN reliability [28]. Traditionally, this RBD-based analysis of WSN data transport protocols has been done using paper-and-pencil proof methods and computer simulations. The paper-and-pencil method begins by representing the transmission of message and routing operation of the given WSN data transport protocol by an appropriate RBD configuration and the assignment of failure distributions to these data transport operations. Usually, *exponential* or *Weibull* distributions, with failure rate λ and time-to-failure random variable, say X, are used in order to express the reliability of these data transport operations. The reliability of these data transport operations, along with the RBD of a WSN data transport protocol, is then used to analytically derive mathematical expressions for the overall WSN data transport protocol reliability. Due to the involvement of manual manipulation and simplification, this kind of analysis is error-prone and the problem gets more severe while analyzing large systems. On the other hand, RBD-based computer simulators, such as ReliaSoft [25] and ASENT Reliability analysis tool [4], can be utilize to provide a more scalable solution for the reliability analysis of WSN data transport protocols. These tools generate samples from the exponential and Weibull random variables to model the reliabilities of the network components. This data is then manipulated using computer arithmetic and numerical techniques to compute the reliability of the complete communication network. However, they cannot ensure absolute correctness as well due to the involvement of pseudo random numbers and numerical methods and the inherent sampling based nature of computer simulations.

Formal methods [17], which are computer based mathematical reasoning techniques, can be used to overcome the inaccuracy limitations of the paper-and-pencil proof methods and simulation for WSN data transport protocols. The main idea behind the formal analysis of any given 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. Formal methods are mainly categorized into two mainstream techniques: model checking [5] and theorem proving [16]. 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

978-1-4673-7701-0/15/$31.00 ©2015 IEEE

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 the reliability analysis of many real-world systems, including power generation plants [26], aerospace systems [9] and simple oil and gas pipelines [3]. However, due to the state-based nature of model checking, it suits the Markov chain based reliability analysis quite well. Whereas, in the context of RBD based reliability analysis, model checking can be used to analyze the properties of dynamic RBDs (DRBDs) only [26].

In this paper, given the involvement of several elements of continuous and random nature, we propose to conduct the formal RBD-based reliability analysis of WSN data transport protocols [28] within the sound core of a higher-order-logic theorem prover [16]. In particular, we formally model the *end-to-end* (e2e) message delivery, Event to Sink Reliable Transport (ESRT) [27] and Reliable Multi-Segment Transport (RMST) [31] data transport protocols in higher-order logic. For this purpose, we build upon the recently proposed higher-order-logic formalization of series RBD, which has been used to conduct reliability analysis of simple oil and gas pipeline [3]. However, this foundational formalization of a series RBD [3] is not sufficient enough to allow the proposed modelling since these protocols involve redundancy and thus their modelling requires parallel RBD configurations. So, we have extended the series RBD formalization [3] to parallel and parallel-series RBD configurations in this paper. The paper also provides the formal verification of the reliability expression for the e2e message delivery and the ESRT and RMST protocols within the sound core of a higher-order-logic theorem. To the best of our knowledge, no formal method have been used to conduct the RBD-based reliability analysis of these WSN data transport protocols, where accuracy of the analysis is a dire need.

II. Related Work

The probabilistic model checking tool, PRISM [24], has been frequently used for the validation of Medium Access Control (MAC) protocols for WSNs [13], [14], [35]. Besides model checking, higher-order-logic theorem proving has also been used for analyzing WSN algorithms. For instance, in [7], a WSN algorithm is formally modelled, within the PVS system, and the feasibility of this approach is illustrated by manually analyzing the trace execution of the Surge algorithm [6], and formally verifying the correctness of the message delivery for the reverse path forwarding algorithm [7]. Recently, formal probabilistic analysis of the k-set randomized scheduling in WSNs has been conducted using the HOL theorem prover [12]. However, these above-mentioned works have not been primarily focused on the reliability analysis of WSNs.

Colored Petri nets (CPN) have been used to model dynamic RBDs (DRBDs) [26], 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 DRBDs models to identify design flaws [26]. Similarly, the probabilistic model checker, PRISM [20], has been used for the quantitative verification of various safety and mission-critical systems, such as failure analysis for an airbag system and the reliability analysis of an industrial process control system and the Herschel-Planck satellite system [23]. However, due to the state-based models, only state related property verification, like deadlock checks, reachability and safety properties, is supported by these approaches, i.e., we cannot verify generic reliability relationships for the given systems using the approaches, presented in [26], [23]. Given the safety-critical nature of WSN transport protocols and thus the dire need of absolute accuracy, we did not choose these model checking and numerical methods based solutions for our work.

The foremost requirement for reasoning about reliability related properties of a system in a theorem prover is the availability of the higher-order-logic formalization of probability theory. Mhamdi's probability theory formalization [21], which is based on extended-real numbers (real numbers including $\pm\infty$), has been recently used to reason about the RBD-based analysis of a series pipelines structure [3] and Fault Tree-based [18] formal failure analysis of satellite's solar array [2], which involves multiple exponential random variables. In the current work, we extend the formalization of [3] to formally reason about parallel and parallel-series RBDs as well. This extension would widen the scope of formal RBD analysis as most of the real-world systems require parallel or a combination of series and parallel RBDs for modeling their respective behaviors.

III. Preliminaries

To facilitate the understanding of the rest of the paper, this section provides a brief introduction to the HOL4 theorem prover [29] and the main formalizations that we build upon, i.e., probability theory [21] and reliability theory [3].

A. HOL4 Theorem Prover

HOL4 is a higher-order-logic theorem prover and is primarily based on the Church's type theory [11] and Hindley-Milner polymorphism [22]. Higher-order logic [10] is an expressive logic that can be used to formally express any system model or mathematical expression that can be described in a closed form. Thus, it supports the formalization of all foundations of RBDs, such as probability theory, recursive definitions and continuous random variables. Similarly, interactive theorem provers allow the users to guide the computer-based proof tools for verifying goals expressed in undecidable logic, such as higher-order logic. The HOL4 core contains 5 axioms and 8 inference rules only and soundness is guaranteed by ensuring that a new theorem can only be verified by applying these basic axioms and primitive inference rules or some previously verified theorems.

The system verification process using the HOL4 theorem prover is generally conducted in three steps: Firstly, the system is modelled as a higher-order-logic function. Secondly, the system properties that have to be verified are formalized as higher-order-logic proof goals. Finally, these proof goals are discharged using appropriate tactics along with the existing library of formally verified results. A number of sound and complete first-order logic automated reasoners are available in HOL that aid the user by automating some parts of the proofs. To facilitate the verification involved in the formal RBD analysis, we propose to develop a library of formally

verified foundational mathematical results in this domain. For this purpose, we utilized the HOL theories of Booleans, lists, sets, positive integers, *real* numbers, measure and probability in our work. In fact, one of the primary motivations of selecting the HOL4 theorem prover for our work was to benefit from these built-in mathematical theories. Table I provides the mathematical interpretations of some frequently used HOL symbols and functions, which are inherited from existing HOL theories.

TABLE I: HOL Symbols and Functions

HOL Symbol	Standard Symbol	Meaning
::	$cons$	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	$element$	n^{th} element of list L
MEM a L	$member$	True if a is a member of list L
λx.t	$\lambda x.t$	Function that maps x to $t(x)$
SUC n	$n+1$	Successor of a num

B. Probability Theory and Random Variables

Based on the measure theoretic foundations, a probability space is defined as a triple (Ω, Σ, Pr), 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 Pr is a measure with domain Σ and is 1 for the whole sample space. In the HOL4 probability theory formalization [21], given a probability space p, the functions `space` and `subsets` return the corresponding Ω and Σ, respectively. Based on this definition, all the basic probability axioms have been verified. Now, a random variable is a measurable function between a probability space and a measurable space, which essentially is a pair $(S, \mathcal{A})$, where S denotes a set and $\mathcal{A}$ represents a nonempty collection of sub-sets of S. A random variable is termed as discrete if S is a set with finite elements and continuous otherwise.

The probability that a random variable X is less than or equal to some value x, $Pr(X \leq x)$ 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 [3]:

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

where the variables p, X and x represent a probability space, a random variable and a $real$ number, respectively. The function `Normal` 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.

$$R(t) = Pr(X > t) = 1 - Pr(X \leq t) = 1 - F_X(t) \quad (1)$$

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 [21], Equation (1) has been formalized as follows [3]:

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

The series RBD, presented in [3], is 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

$$Pr(\bigcap_{i=1}^{N} L_i) = \prod_{i=1}^{N} Pr(L_i) \quad (2)$$

This concept has been formalized as follows [3]:

```
⊢ ∀ 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 one or more events n taken from any permutation of the given list L, the property $Pr(\bigcap_{i=1}^{N} L_i) = \prod_{i=1}^{N} Pr(L_i)$ holds.

IV. Formalization of Reliability Block Diagrams

In this section, we present the HOL formalization of series, parallel and parallel-series RBD configurations by utilizing *list* as a basic data-type. The list data-type allows us to verify the corresponding generic reliability expressions. Moreover, the definitions, presented in this sections, are mainly recursive and the proof of the theorems are inductive in nature.

A. 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 Figure 1. If $A_i(t)$ is a mutually independent event that represents the reliable functioning of the i^{th} component of a serially connected system with N components at time t, then the overall reliability of the complete system can be expressed as [8]:

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

Fig. 1: Series RBD Configuration

We formalized the serial RBD configuration as follows:

Definition 1: `⊢ ∀ p L. series_struct p L = inter_list p L`

The function `series_struct` takes a list of events L corresponding to the failure of individual components of the given system and the probability space p and returns the series structure event of the complete system. The function `inter_list` returns the intersection of all of the elements of the given list and the whole probability space, if the given list is empty. Based on this function definition, the result of Equation (3) can be formally verified as follows:

Theorem 1:
```
⊢ ∀ p L. prob_space p ∧
∼NULL L ∧ mutual_indep p L ⇒
 (prob p (series_struct p L) =
 list_prod (list_prob p L))
```

The first assumption ensures that p is a valid probability space based on the probability theory in HOL4 [21]. The next two assumptions guarantee that the list of events, representing the reliability of individual components, must have at least one event and the reliability events are mutually independent. The conclusion of the theorem represents Equation (3). It is important to note that our `series_struct` definition accepts a list of reliability events and it is thus different from the corresponding formalization, presented in [3], which accepts a list of random variables and is not general enough to cater for nested RBDs.

B. Formalization of Parallel Reliability Block Diagram

The reliability of a system with parallel connected sub-modules, depicted in Figure 2, mainly depends on the component with the maximum reliability. In other words, the system will continue functioning so long as at least one of its components remains functional. If the event $A_i(t)$ represents the reliable functioning of the i^{th} component of a system with N parallel components at time t, then the overall reliability of the system can be mathematically expressed as [8]:

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

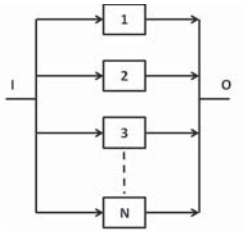

Fig. 2: Parallel RBD Configuration

Now, the reliability of a system with a parallel structure is defined as:

Definition 2: `⊢ ∀ L . parallel_struct L = union_list L`

The function `parallel_struct` accepts a list of reliability events and returns the parallel structure reliability event, where the function `union_list` recursively performs the union operation on the given list of reliability events.

Based on the above definition, we first formally verify the following lemma that provides an alternate expression for the parallel structure in terms of the series structure:

Lemma 1:
```
⊢ ∀ L p. (prob_space p) ∧
(∀ x'. MEM x' L ⇒ x' ∈ events p) ⇒
  (prob p (parallel_struct L) =
  1 - prob p (inter_list p (compl_list p L))
```

where the function `compl_list` returns a list of events such that each element of this list is the difference between the probability space p and the corresponding element of the given list.

Now, we can formally verify Equation (4) as follows:

Theorem 2:
```
⊢ ∀ p L. (prob_space p) ∧
∼NULL L ∧ (mutual_indep p L) ∧
(∀ x'. MEM x' L ⇒ x' ∈ events p) ⇒
 (prob p (parallel_struct L) =
 1 - list_prod
  (one_minus_list (list_prob p L)))
```

The above theorem is verified under the same assumptions as Theorem 1. The conclusion of the theorem represents Equation (4) where, the function `one_minus_list`, which accepts a list of $real$ numbers $[x1, x2, x3, \cdots, xn]$ 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., $[1-x1, 1-x2, 1-x3, \cdots, 1-xn]$. The proof of Theorem 2 is primarily based on Lemma 1 and Theorem 1 along with the fact that given the list of n mutually independent events, the complement of these n events are also mutually independent.

C. Formalization of Parallel-Series Reliability Block Diagram

Most safety-critical systems in the real-world contain many reserved sub-stages for backup in order to ensure reliable operation. If the components in these reserved *subsystems* are connected serially then the structure is called a parallel-series structure, as depicted in Figure 3. If $A_{ij}(t)$ is the event

corresponding to the reliability of the j^{th} component connected in a i^{th} subsystem at time t, then the reliability of the complete system can be expressed as follows:

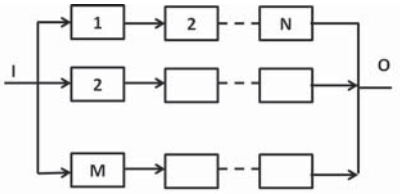

Fig. 3: Parallel-Series RBD Configuration

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

Now using Equation (5), the reliability of the parallel-series structure can be formalized in HOL4 as follows:

Definition 3: `⊢ ∀ p L.`
```
parallel_series_struct p L =
parallel_struct (list_inter_list p L)
```

The function `parallel_series_struct` accepts a two dimensional list L, i.e., a list of lists, along with a probability space p and returns the corresponding reliability event of the system constituted from the parallel connection of the serial stages. The function `parallel_struct`, given in Definition 2, is used to model the parallel connection while the function `list_inter_list` is used to model the serial stages:

Definition 4: `⊢ ∀ p L.`
```
list_inter_list p L=MAP (λa.inter_list p a) L
```

The `list_inter_list` function takes a list of lists L and probability space p and returns a list by mapping the `inter_list` function on every element of the given two dimensional list.

Now, we define a recursive function to model the right-hand-side of Equation (5) in HOL4 as follows:

Definition 5: `⊢ ∀ p.`
```
list_rel_list_prod p [] = [] ∧
∀ p h t. list_rel_list_prod p (h::t) =
list_prod (list_prob p h)::
 list_rel_list_prod p t
```

The function `list_rel_list_prod` accepts a two dimensional list of events, representing the time to failure of individual components connected in a parallel-series structure along with the probability space p and returns a list of product of reliabilities of the components connected serially at every stage. The functions `list_prod` and `list_prob` are used to model the product of reliabilities and the events corresponding to the component functioning reliably at the desired time, respectively.

Now, we can formally model Equation (5) as follows:

Theorem 3:
```
⊢ ∀ p L. (prob_space p) ∧
(∀z. MEM z L ⇒ ∼NULL z) ∧
(mutual_indep p (FLAT L)) ∧
(∀x'. MEM x'(FLAT L) ⇒ x' ∈ events p) ⇒
 (prob p (parallel_series_struct p L) =
 1 − list_prod
  (one_minus_list(list_rel_list_prod p L)))
```

The first two assumptions in Theorem 3 are similar to the ones used in Theorem 2. The next three assumptions ensure that the sub-lists corresponding to the sub-stages are not empty and the reliability events corresponding to the sub-components of the parallel-series structure are valid events of the given probability space p and are mutually independent. The HOL4 function `FLAT` is used to convert the two dimensional list into a single list. The conclusion models the right-hand-side of Equation (5). The proof of the above theorem uses the result of Theorem 1 and a lemma which states that given the list of mutually independent reliability events, the reliability event, associated with a sub-component, is independent in probability with the event, corresponding to the reliability of a sub-block of the overall parallel-series structure.

The formalization reported in this paper so far took about 200 man-hours and more then 4000 lines of HOL4 proof script, which is available for download at [1]. The most challenging part in the reasoning process was to verify that given the mutual independence of individual events, the event corresponding to a sub-configuration (series, parallel or parallel-series) is also mutually independent from other sub-configurations and individual sub-modules. The rest of the verification process was primarily based on probabilistic, set-theoretic and arithmetic simplification and some parts of the proofs were also handled automatically using various built-in automatic provers and simplifiers in HOL4.

The formal verifications of the above mentioned theorems, which are available in reliability textbooks, guarantee the correctness of our formal definitions. Moreover, the formal verification of these properties is expected to facilitate the process of formal reasoning about RBD-based analysis of WSN data transport protocols as will be demonstrated in the next section.

V. Formalization of Wireless Sensor Network Transport Protocols

To ensure reliable transport of data within a WSN, several end-to-end (e2e) data transport protocols, including Event to Sink Reliable Transport (ESRT) and Reliable Multi-Segment Transport (RMST), have been developed in the past few years. These data transport protocols provide resilience to different kinds of networking failures, such as communication failures and message losses. In this section, we present the RBD-based formal reliability analysis of WSN general end-to-end message delivery mechanism and the commonly used WSN data transport protocols, i.e., ESRT and RMST.

The ESRT belongs to the end-to-sink class of protocols, depicted in Figure 4(a), and achieves the optimal operating point by adjusting the reporting rate of sensor nodes depending upon the current network load. In this approach, the sink

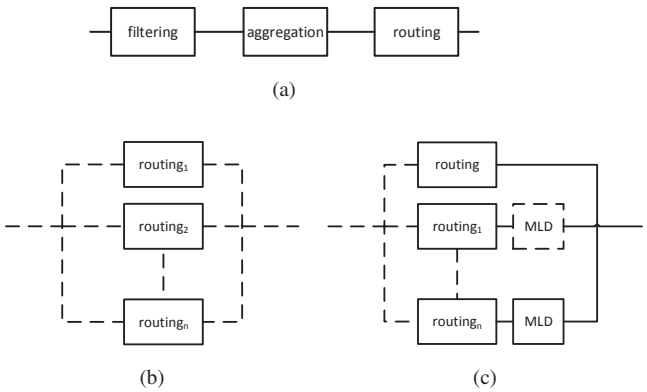

Fig. 4: RBDs for the (a) General e2e Data Transport Mechanism (b) ESRT Data Transport Protocol (c) RMST Data Transport Protocol

node is interested in the collective information coming from a number of nodes instead of the individual sensor report. Therefore, the data transport in ESRT consists of n-parallel routing blocks and this behavior can be modeled by using the parallel RBD configuration as shown in Figure 4(b) [28]. Similarly, RMST is an end-to-end protocol, which utilizes the Selective Negative Acknowledgment (NACK) retransmission mechanism to increase the reliability of data transport. There are two main data transport operations in RMST: (i) Routing is used to identify potential routes for data transport (ii) Message Loss Detection (MLD) is used to retransmit transport data and is thus an essential part of reliable data transmission. By incorporating these operations, the reliability of the RMST data transport mechanism, from the sensor nodes to the sink, can be modeled by using the series-parallel RBD configuration as shown in Figure 4(c) [28].

In order to formalize the WSN data transport protocols, presented in Figure 4, we first need to formally model the reliability events that are associated with operations of the WSN data transport protocols, such as routing and MLD. A reliability event list constructed from the list of random variables can be formalized in HOL4 as follows:

Definition 6:
```
⊢ ∀ p x.rel_event_list p [] x = [] ∧
∀ p x h t.rel_event_list p (h::t) x =
PREIMAGE h {y | Normal x < y} ∩ p_space p ::
  rel_event_list p t x
```

The function `rel_event_list` accepts a probability space p, a list of random variables, representing the failure time of individual components, and a real number x, which represents the time index at which the reliability is desired. It returns a list of events, representing the proper functioning of all individual components at time x.

Definition 7:
```
⊢ ∀ p L x.
List_rel_event_list p L x =
MAP (λa. rel_event_list p a x) L
```

The function `List_rel_event_list` accepts a probability space p, a list of random variables, representing the failure time of individual components, and a real number x, which represents the time index at which the reliability is desired. It returns a two dimensional list of events by mapping the function `rel_event_list` on every element of the given two dimensional list of random variables, which in turn models the proper functioning of all individual components at time x. The HOL4 formalization of the exponential distribution for parallel-series network is as follows:

Definition 8:
```
⊢ ∀ p X l. exp_dist p X l =
∀ x. (CDF p X x = if 0 ≤ x then
 1 - exp (-l * x) else 0)
```

The function `exp_dist` guarantees that the CDF of the random variable X is that of an exponential random variable with a failure rate l in a probability space p. We classify a list of exponentially distributed random variables based on this definition as follows:

Definition 9:
```
⊢ ∀ p L. list_exp p [] L = T ∧
∀ p h t L. list_exp p (h::t) L =
exp_dist p (HD L) h ∧ list_exp p t (TL L)
```

The function `list_exp` accepts a list of failure rates, a list of random variables L and a probability space p. It guarantees that all elements of the list L are exponentially distributed with the corresponding failure rates, given in the other list, within the probability space p. For this purpose, it utilizes the list functions `HD` and `TL`, which return the *head* and *tail* of a list, respectively. Next we model a two dimensional list of exponential distribution functions to model nodes connected in a series-parallel RBD as follows:

Definition 10:
```
⊢ (∀ p L.
list_list_exp p [] L = T) ∧
∀ h t p L. list_list_exp p (h::t) L =
list_exp p h (HD L) ∧ list_list_exp p t (TL L)
```

The `list_list_exp` function accepts two lists, i.e., a two dimensional list of failure rates and random variables L, corresponding to the components at each stage of a series-parallel RBD. It calls the function `list_exp` recursively to ensure that all elements of the list L are exponentially distributed with the corresponding failure rates, given in the other list, within the probability space p.

Now, we can verify the reliability expression of the e2e WSN data transport mechanism, shown in Figure 4(a), by using the formalised series RBD configuration in HOL4 as follows:

Theorem 4:
```
⊢ ∀ X_fil X_aggr X_rout C_fil C_aggr
C_rout p t.
0 ≤ t ∧ prob_space p ∧
(∀x'. MEM x'
 rel_event_list p [X_fil;X_aggr;X_rout] t ⇒
    x' ∈ events p) ∧
mutual_indep p
 rel_event_list p [X_fil;X_aggr;X_rout] t ∧
list_exp p [C_fil;C_aggr;C_rout]
    [X_fil;X_aggr;X_rout] ⇒
  prob p (series_struct p
   rel_event_list p [X_fil;X_aggr;X_rout] t =
  exp (-list_sum [C_fil;C_aggr;C_rout]*t)
```

where the function list_sum returns the sum of all the elements of the given failure rate list. The first assumption ensures that the variable t models time as it can acquire positive integer values only. The next assumption ensures that p is a valid probability space based on the probability theory in HOL4 [21]. The next two assumptions ensure that the events corresponding to the failures modeled, by the random variables X_oper_fil; X_oper_aggr; X_oper_rout are valid events from the probability space p and they are mutually independent. Finally, the last assumption characterizes the random variables X_oper_fil; X_oper_aggr; X_oper_rout, as exponential random variables with failure rates C_oper_fil; C_oper_aggr; C_oper_rout, respectively. The conclusion of Theorem 4 represents the reliability of the general e2e WSN data transport mechanism between sensors nodes to sinks in terms of their failure rates.

Similarly, the formally verified reliability expression for the ESRT WSN data transport protocol, by considering n-parallel routing nodes modeled as a parallel RBD configuration as shown in Figure 4 (b), is as follows:

Theorem 5:
```
⊢ ∀ X_rout_list C_rout_list p t.
(0 ≤ t) ∧ (prob_space p) ∧
mutual_indep p
 rel_event_list p X_rout_list t ∧
∀x'. MEM x'
 (rel_event_list p X_routing_list) t ⇒
x' ∈ events p ∧
list_exp p C_routing_list X_routing_list ⇒
 prob p (parallel_struct
  (rel_event_list p X_routing_list t)) =
 1 - list_prod
  (one_minus_exp t C_routing_list)
```

where the function one_minus_exp takes time index variable *t* and failure rate list and returns a list of one minus exponential function, i.e., each element of the list is of the form $1 - exp^{-c1*t}$, where *c1* is the arbitrary failure rate variable. The assumptions of the above theorem are similar to the ones used in Theorem 4 and the proof of Theorem 5 is based on Theorem 2 and some basic arithmetic lemmas and probability theory axioms.

Finally, a generic reliability expression for RMST data transport protocol, consisting of *r*-retransmissions and MLD operations, can be verified in HOL4 as follows:

Theorem 6:
```
⊢ ∀ X_rout X_MLD C_rout C_MLD p t.
(0 ≤ t) ∧ (prob_space p) ∧
(∀z. MEM z (List_rel_event_list p
 (RMST_rv_list X_rout X_MLD) t) ⇒ ∼NULL z) ∧
mutual_indep p
 (FLAT(List_rel_event_list p
  ([X_rout]::RMST_rv_list X_rout X_MLD) t)) ∧
PREIMAGE X_rout {y| y ≤ Normal t} ∈ events p ∧
PREIMAGE X_MLD {y| y ≤ Normal t} ∈ events p ∧
LENGTH (RMST_rv_list X_rout X_MLD) =
LENGTH (RMST_fail_rate C_rout C_MLD) ∧
list_list_exp p
 ([C_rout]::RMST_fail_rate C_rout C_MLD)
 ([X_rout]::RMST_rv_list X_rout X_MLD) ⇒
 prob p (parallel_series_struct p
  (list_rel_event_list p
   ([X_rout]::RMST_rv_list X_rout X_MLD) t)) =
 1 - list_prod (one_minus_list
  (list_exp_sum
   ([C_rout]::RMST_fail_rate C_rout C_MLD) t)
```

where the functions RMST_rv_list and RMST_fail_rate_list take random variables and failure rates associated with the *routing* and *MLD* operations and return a two dimensional list of random variables and failure rates, respectively, where each element of these two dimensional lists, which is itself a list, contains the random variables *X_routing* and *X_MLD* and failure rate variables *C_routing* and *C_MLD*. The function list_exp_sum accepts a two dimensional list of failure rates and a time index variable and returns a list of negative exponentials. The exponent of these exponentials is obtained by applying the function list_sum, which returns the sum of all the elements in a given list, on each member of a given two dimensional failure rate list. For example, list_exp_sum [[c1; c2; c3]; [c4; c5]; [c6; c7; c8] t = [exp -(c1+c2+c3)t; exp -(c4+c5)t; exp -(c6+c7+c8)t]. The proof of Theorem 6 involves Theorem 3 and some basic probability theory axioms and some properties of the exponential function exp. The reasoning process of Theorems 4, 5 and 6 took about more than 1000 lines of HOL4 script and was very straightforward compared to the reasoning for the verification of Theorems 1, 2 and 3 [1], which involved probability-theoretic guidance.

The distinguishing features of the formally verified Theorems 4, 5 and 6, compared to the reliability analysis of the WSN data transport protocols of Figure 4 in [28], includes their generic nature, i.e., all variables are universally quantified and thus can be specialized to obtain the reliability for any number of routing and MLD operations and for any given failures. Moreover, these theorems are guaranteed to be complete and true 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 theorem. In addition, these formally verified reliability theorems provide useful insight to the network design engineers to compare and correct their estimated reliability results, which are traditionally either obtained through manual manipulation or computer simulation. For instance, it is very handy to know that the reliability of these *e2e* data transport protocols increases with the increase in the number of data retransmissions. So, by keeping this in mind, our formalization facilitates the network design engineers to accurately determine the total number of retransmissions, which are required to achieve a desired level of reliability. To the best of our knowledge, the above-mentioned benefits are not shared by any other computer based reliability analysis approach for WSN data transport protocols.

VI. Conclusions

The accuracy of reliability analysis of WSNs has become a dire need these days due to their extensive usage in safety-critical applications, where an incorrect reliability estimate may lead to disastrous situations including the loss of innocent lives. In this paper, we presented a higher-order-logic formalization of commonly used RBD configurations, i.e., series, parallel and parallel-series, to facilitate the formal reliability analysis of WSN data transport protocols within a theorem

prover. Building upon the results presented in this paper, the formalization of other commonly used RBDs, including series-parallel and K-out-of-N, and the Weibull random variable is underway. Besides WSN, we also plan to utilize these foundational formalizations to conduct the formal failure and reliability analysis of the smart grid substations communication networks [19], which are quite similar to WSNs.

ACKNOWLEDGMENTS

This publication was made possible by NPRP grant # [5 - 813 - 1 134] from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the author[s].

REFERENCES

[1] W. Ahmad. Formal Reliability Analysis of Wireless Sensor Network Data Transport Protocols using HOL, Proof Script. http://save.seecs.nust.edu.pk/projects/rbd/wsn, 2015.

[2] W. Ahmed and O. Hasan. Towards Formal Fault Tree Analysis Using Theorem Proving. In *Conferences on Intelligent Computer Mathematics*, volume 9150 of *LNCS*, pages 39–54. Springer, 2015.

[3] W. Ahmed, O. Hasan, S. Tahar, and M. S. Hamdi. Towards the Formal Reliability Analysis of Oil and Gas Pipelines. In *Intelligent Computer Mathematics*, volume 8543 of *LNCS*, pages 30–44. Springer, 2014.

[4] ASENT. https://www.raytheoneagle.com/asent/rbd.htm, 2015.

[5] C. Baier and J. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[6] C. Bernardeschi, P. Masci, and H. Pfeifer. Early Prototyping of Wireless Sensor Network Algorithms in PVS. In *Computer Safety, Reliability, and Security*, volume 5219 of *LNCS*, pages 346–359. Springer, 2008.

[7] C. Bernardeschi, P. Masci, and H. Pfeifer. Analysis of Wireless Sensor Network Protocols in Dynamic Scenarios. In *Stabilization, Safety, and Security of Distributed Systems*, volume 5873 of *LNCS*, pages 105–119. Springer, 2009.

[8] R. Bilintion and R.N. Allan. *Reliability Evaluation of Engineering System*. Springer, 1992.

[9] M. Bozzano, A. Cimatti, J. P. Katoen, V. Y. Nguyen, T. Noll, and M. Roveri. The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems. In *Computer Safety, Reliability, and Security*, volume 5775 of *LNCS*, pages 173–186. Springer, 2009.

[10] C.E. Brown. *Automated Reasoning in Higher-order Logic*. College Publications, 2007.

[11] A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[12] M. Elleuch, O. Hasan, S. Tahar, and M. Abid. Formal Probabilistic Analysis of a Wireless Sensor Network for Forest Fire Detection. volume 122 of *EPTCS*, pages 1–9. 2013.

[13] A. Fehnker, L. Van Hoesel, and A. Mader. Modelling and Verification of the LMAC Protocol for Wireless Sensor Networks. In *Integrated Formal Methods*, volume 4591 of *LNCS*, pages 253–272. Springer, 2007.

[14] M. Fruth. Probabilistic Model Checking of Contention Resolution in the IEEE 802.15.4 Low Rate Wireless Personal Area Network Protocol. In *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, pages 290–297. IEEE, 2006.

[15] Hiroshi H. and C. N. Skinner. 2008 Forest Fires in the Northern California, USA. Technical Report, U.S. Forest Service, Pacific Southwest Research Station, Redding, California, USA, Octobor 2009; https://ams.confex.com/ams/pdfpapers/155842.pdf.

[16] J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.

[17] O. Hasan and S. Tahar. Formal Verification Methods. In *Encyclopedia of Information Science and Technology*, pages 7162–7170. IGI Global, 2014.

[18] IEC. International Electrotechnical Commission, 61025 Fault Tree Analysis. Technical report, https://webstore.iec.ch/publication/4311, 2006.

[19] M. G. Kanabar and T. S. Sidhu. Reliability and Availability Analysis of IEC 61850 based Substation Communication Architectures. In *Power & Energy Society General Meetings*, pages 1–8. IEEE, 2009.

[20] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Computer Aided Verification*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[21] T. Mhamdi, O. Hasan, and S. Tahar. On the Formalization of the Lebesgue Integration Theory in HOL. In *Interactive Theorem Proving*, volume 6172 of *LNCS*, pages 387–402. Springer, 2010.

[22] R. Milner. A Theory of Type Polymorphism in Programming. *Journal of Computer and System Sciences*, 17:348–375, 1977.

[23] G. Norman and D. Parker. *Quantitative Verification: Formal Guarantees for Timeliness, Reliability and Performance*. A Report by London Mathematical Society and Smith Institute, 2014.

[24] PRISM. www.cs.bham.ac.uk/∼dxp/prism, 2015.

[25] ReliaSoft. http://www.reliasoft.com/, 2015.

[26] R. Robidoux, H. Xu, L. Xing, and M. Zhou. Automated Modeling of Dynamic Reliability Block Diagrams Using Colored Petri Nets. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(2):337–351, 2010.

[27] Y. Sankarasubramaniam, O. B Akan, and I. F. Akyildiz. Esrt: event-to-sink Reliable Transport in Wireless Sensor Networks. In *Mobile ad hoc Networking & Computing*, pages 177–188. ACM, 2003.

[28] F. K. Shaikh, A. Khelil, and N. Suri. On Modeling the Reliability of Data Transport in Wireless Sensor Networks. In *Parallel, Distributed and Network-Based Processing*, pages 395–402. IEEE, 2007.

[29] K. Slind and M. Norrish. A Brief Overview of HOL4. In *Theorem Proving in Higher-order Logics*, volume 5170 of *LNCS*, pages 28–32. Springer, 2008.

[30] J. Soszynska. Reliability and Risk Evaluation of a Port Oil Pipeline Transportation System in Variable Operation conditions. *International Journal of Pressure Vessels and Piping*, 87(2-3):81–87, 2010.

[31] F. Stann and J. Heidemann. Rmst: Reliable Data Transport in Sensor Networks. In *Sensor Network Protocols and Applications*, pages 102–112. IEEE, 2003.

[32] R. K. Yedavalli and R. K. Belapurkar. Application of Wireless Sensor Networks to Aircraft Control and Health Management Systems. *Journal of Control Theory and Applications*, 9(1):28–33, 2011.

[33] J. Yick, B. Mukherjee, and D. Ghosal. Wireless Sensor Network Survey. *Computer Networks*, 52(12):2292–2330, 2008.

[34] Li. Yu, N. Wang, and X. Meng. Real-time Forest Fire Detection with Wireless Sensor Networks. In *Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1214–1217. IEEE, 2005.

[35] H. Zayani, K. Barkaoui, and R. Ben Ayed. Probabilistic Verification and Evaluation of Backoff Procedure of the WSN ECo-MAC Protocol. *Wireless & Mobile Networks*, pages 156–170, 2010.