Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 881–888

# Approaches of Wireless Sensor Network Dependability Assessment

Antonio Coronato and Alessandro Testa

Institute for High-Performance Computing and Networking - CNR
Via P. Castellino 111, 80125 Napoli, Italy
{antonio.coronato, alessandro.testa}@na.icar.cnr.it

***Abstract*—The extensive use of the Wireless Sensor Networks (WSNs) in main critical scenarios stresses the need to verify their dependability properties at design time to prevent wrong design choices and at runtime in order to make a WSN more robust against failures that may occur during its operation. In literature, several approaches have been proposed in order to evaluate the dependability of a WSN during its inception and its operating.**

**In this paper we present a survey on these adopted techniques reporting aspects and characteristics of some research studies. Moreover, by means of a comparison grid, we analyze the current state-of-the-art of the approaches of WSN dependability assessment in order to identify the most performant and to discuss the ongoing challenges.**

***Index Terms*—Wireless Sensor Networks, Dependability, Reliability, Fault-Tolerance**

## I. Introduction

NOWADAYS Wireless Sensor Networks (WSNs) are usually involved for critical systems monitoring [1][2] and in smart environments [3], thus the recent scientific production on WSNs dependability assessment is grown .

Unexpected events, such as node crash and packet loss, may affect the dependability of the WSN and hence it is necessary evaluate its robustness from the early stages of the development process (*design phase*) onwards to minimize the chances of unexpected problems during use. It is also crucial to monitor a WSN at runtime (*operating phase*) and to detect undesired effects that cannot be analyzed before the WSN deployment.

The approaches adopted in literature to assess WSN dependability, at *design time* or *runtime*, can be categorized in four classes: *experimental*, *simulative*, *analytical* and *formal*.

The first allows to analyze dependability at runtime by means of experiments. Experimental methods are used to evaluate a real system and they require the deployment of a real WSN. They are useful in operating phase since by means of them we can perform experiments directly on the real system from which we collect data. Among the experimental approaches we consider the *Fault Injection* techniques[4] which allow to evaluate the dependability of a real WSN by injecting faults.

The simulative approaches make use of well-known simulators that can be adopted at design time; this kind of approach consists in modeling the WSN and making an estimate of the dependability. In a similar way analytical approaches are conceived; the difference is in considering mathematical models to check the WSN behavior during the design. Finally, formal approaches use specifications of correctness and they can be adopted to assess WSN dependability both at design time and at runtime. Formal approaches offer a new opportunity for studying the WSN dependability.

The aim of this paper is to revise experimental, simulative, analytical and formal approaches and tools currently used in the field of WSN dependability assessment, including related studies. We want to provide a survey on the main approaches adopted for the WSN dependability assessment evaluating the best choice to perform dependability assessment of WSNs. A comparison of related work is presented to summarize the state-of-art and reason about what is still missing and the ongoing challenges.

The rest of the paper is organized as follows. In Section II experimental approaches are presented; the Section III includes the simulative approaches. An analysis of analytical approaches is documented in Section IV and the formal approaches are discussed in Section V. Finally in Section VI we report a comparison of the discussed papers and the Section VII ends the paper with conclusions.

## II. Experimental Approaches

Experimental approaches are used to measure the WSN dependability directly from a real system, during its operation. In the prototyping phase, it is possible to perform an accelerated testing, for example by forcing a fault (by means of *Fault Injection* (FI) [4], [5]); it is also possible to collect occurring failures directly from system (by means of *Field Failure Data Analysis (FFDA)* techniques [6]).

FI is defined as the dependability validation technique based on the observation of the system behavior under the presence of faults which are deliberately introduced into the system [7].

Typically FI is used to i) assess the dependability level of a target system, such as an operational systems, a system

This work has been partially supported by the ”Se@me: Sustainable e-maritime @ssistance for Maritime Employees, Passengers and Yachtsmen” project, funded under the ”POR Campania 2007/2013 ” initiative of the Regione Campania - Italy.

TABLE I
FAULT INJECTION TOOLS

| Tool | Technique | Fault Model |
|---|---|---|
| XCEPTION [8] | SWIFI with exception trigger | Transient faults |
| FERRARI [9] | SWIFI with interrupt, fork, trap | Transient and permanent faults |
| FIAT [10] | SWIFI with exception trigger | Bit-flip faults in the memory |
| NFTAPE [11] | SWIFI with exception trigger | Several types of faults (*arbitrary model*) |
| MESSALINE [12] | HWIFI with forcing and insertion | Faults of type *stuck-at-0*, *stuck-at-1*, *logical bridging*, *physical bridging* |
| AVR-INJECT [4] | SWIFI with exception trigger | Bit-flip faults in the memory area, code area and special registers |

prototype, or an emulated execution environment (the last two options are used especially in the pre-deployment phase of the system), and ii) to shed some light on the design choice of a system, for instance, showing its potential dependability bottlenecks. FI tries to determine whether the response of a system matches its specifications in the presence of a defined space of faults.

The implementation of tools for injecting faults has been the focus of several studies. Table I reports a summary of well known tools for fault injection in WSN and in other types of systems. Beyond their inherent differences, they operate in a similar way: each of them performs a study of the fault-free target, obtaining a 'gold file'; then, it injects a fault (obtaining the 'fault file') and it compares the gold file with the fault file, to evaluate the system behavior in response to the fault. Among of the tools mentioned in the table I, there is the AVR-Inject Tool which has been conceived to operate with WSNs. Unfortunately the AVR-Inject tool cannot be used at design time since it needs a prototype of the system, an assembly code that runs on the sensors and thus it needs very detailed information in design phase. Cinque et al. in [13] perform a fault-injection campaign in order to analyze the dependability for three different WSN operating systems (TinyOS, MantisOS, LiteOS). They consider a fault model without taking in account some dependability metric.

*Field Failure Data Analysis* (FFDA) [14] of a system represents the set of fault forecasting techniques which are performed at runtime. By means of this analysis, the dependability attributes of an actual and deployed system are measured considering real conditions. A system which is in normal operation is observed and the natural occurring errors and failures are monitored and recorded in log files. The FFDA is not practical, not feasible for the WSNs since they do not provide log and they have to be lightweight [15].

Other experimental approaches are described in [16] and [17].

In [16] authors present a deployment of 27 Crossbow Mica2 motes that compose a WSN. They describe a Structure-Aware Self-Adaptive WSN system (SASA) designed in order to detect changes of the network due to unexpected collapses and to maintain the WSN integrity. Detection latency, system errors, network bandwidth and packet loss rate are measured; coverage and connection resiliency metrics are not considered. A large scale simulation is performed in order to evaluate the system scalability and reliability.

Pennington et al. [17] assert that, due to the high complexity of the WSN dynamics, it is difficult to predict problems that may occur after the deployment of the WSN. Therefore, in their paper they propose a Integrity-Checking framework which considers real inputs for the testing and validation process. No case study is shown for framework evaluation.

Experimental approaches for WSN dependability assessment allow to gain insight in the actual failure behavior of WSNs and to establish the reliability degree of the network. However, results are difficult to reproduce and for this reason research studies on WSNs have migrated towards simulative and analytical approaches.

## III. SIMULATIVE APPROACHES

A simulative approach for assessing WSNs usually makes use of behavioral simulators, i.e., tools able to reproduce the expected behavior of a system by means of a code-based description. Behavioral simulators allow to reproduce the expected behavior of WSN nodes on the basis of the real application planned to be executed on nodes. However, it is not always possible to observe non-functional properties of WSNs by means of simulative approaches, since models need to be redefined and adapted to the specific network to simulate.

Typical simulative approaches to evaluate WSN fault/failure models are provided in [18] and [19].

In [18] authors address the problem of modeling and evaluating the reliability of the communication infrastructure of a WSN. Authors assume that failures can be categorized in node and network failures.

The first on-line model-based testing technique [19] has been conceived to identify the sensors that have the highest probability to be faulty. The effectiveness of the approach is evaluated in the presence of random noise using a system of light sensors; a fault classification taxonomy is introduced.

Some work like [20] and [21] provide code generation of wireless sensor network applications to perform behavioral simulation and performance analysis.

In [20], a framework for modeling, simulation and code generation of WSNs is presented. The framework is based on Simulink, Stateflow and Embedded Coder; it allows engineers to simulate and automatically generate code of sensor network applications based on MathWorks tools. By means of this tool, an application developer can configure the connectivity of the sensor nodes and can start simulation and functional verification of the application. This framework is able to generate the complete application code for several target operating systems (e.g. TinyOS and MantisOS) from the simulated model.

In [21] a model-driven development process (MDD) is presented to obtain a major effort of optimization for WSN applications. In this work a set of modeling languages is the starting point for code generation and performance analysis.

Finally, the network lifetime is analyzed in [22]; to calculate the lifetime of a WSN, the authors perform simulation by means of a Castalia-based approach [23] that models path-loss.

### A. Simulators

Several simulators for WSNs have been proposed in literature, such as *NS-2*, *OMNet++*, *Prowler*, *TOSSIM*, *OPNET* and *Avrora*.

NS-2 [24] is an event-based simulation tool for WSN. It is amply adopted in academic research being open source and easy to use. The simulations are written with C++/C languages and they can be observed graphically by Network AniMator (NAM).

OMNeT++ [25] is a component-based discrete network simulator. Even this simulator is based on C++ language and it has graphical tools for simulation building and evaluating results in real time. The most recent simulation environment built on OMNeT++ is Castalia [23]. This framework was realized for Wireless Sensor Networks, Body Area Networks [26] and networks of low-power embedded devices and it allows to test distributed algorithms and protocols for WSN considering some features of a real WSN like wireless channel, power consumption and considering a real node behavior. Castalia can be used to simulate a wide set of wireless sensor platforms.

Prowler [27] is an event-driven WSN simulator conceived to operate in Matlab environment. Initially it was realized to simulate MICA motes but then it has been extended also for more general platforms. Advantages of Matlab environments are simple implementing of applications, friendly GUI interface and good visualization facilities. By means of this simulator, it is possible to perform deterministic simulation to test application code of a WSN application and to perform probabilistic simulation to observe the behavior of the sensor nodes.

TOSSIM [28], [29] is the simulator built for TinyOS applications. Actually it is an emulator rather than a simulator since it runs actual application code; it allows to simulate the hardware of a sensor but it does not provide information about WSN dependability. Moreover TOSSIM is provided of a visualization tool, TinyViz.

OPNET [30] is a discrete event, object oriented network simulator. This tool was developed initially for military purposes but its large use grew as much to be considered also for commercial use. OPNET is a powerful software that can be used for research purpose and also as a network design tool.

Finally Avrora [31] is a simulator that adopts an approach which is more oriented to the verification of behavioral properties or performance indicators, and not oriented to the observation of dependability properties. It is a low-level emulator of the AVR processor mainly used to test the behavior of WSNs application prior to their deployment. It executes the disassembled code instruction per instruction and emulates the hardware of the processor and the hardware of the node (memory, LEDs, sensors, radio channel, etc.).

## IV. Analytical Approaches

The study of the performance and dependability of WSNs can be performed by means of analytical models [32]. Some of these models are based on a mathematical representation of the WSN characteristics and are solved by means of simulation.

In [33] authors introduce an approach for the automated generation of WSN dependability models, based on a variant of Petri nets.

An analytical model to predict the battery exhaustion and the lifetime of a WSN, *LEACH*, is discussed in [34].

In [35] the authors present a network state model used to forecast the energy consumption of a sensor.

AboElFotoh et al. [36] present a probabilistic technique to observe the WSN behavior and discuss about dependability of a WSN; they suppose that the main causes of the failures are related to the crashes, power failures and natural causes. The authors evaluate dependability on the basis of the number of packets received by the sink in a deterministic time (*decision interval*). The dependability is computed evaluating the delay of the expected message.

In [37] authors develop an analytical model to investigate the relation between energy saving and system performance and to observe the effects when sensor sleep/active mode vary. By means of this model, authors can obtain several performance metrics, such as the distribution of the data delivery delay. This work adopts analytical model specifically representing the sensor in sleep/active mode considering channel contention and routing issues. In this work authors model a WSN by means of Markovian techniques; they assess dependability using data delivery resiliency and power consumption metrics.

A linear programming model [38] is introduced to address the problem of "multi-hop lifetime aware routing". The authors propose a Garg-Konemann-based approach to obtain the minimum cost arborescence for reaching the sink node optimizing the lifetime of sensor nodes.

Finally in [39] the node aging problem is addressed. The authors try to solve this problem by associating a survivor function for each sensor node (using *Weibull* distribution). The aim of this work is to demonstrate that the node aging process has an important impact on the connectivity at the increasing of the hop distance. By means of a mathematical analysis and a simulation, they observe that nodes at first hop consume their energy because of the aggregation with children nodes. Hence, they assert that the consumption is related to the number of children nodes.

Each analyzed work, which applies a simulative or analytical approach, defined its own fault model making simple assumptions on network topology and on power consumption; results cannot be generalized since they are obtained by means of abstract simulations. Thus there is a lack of realistic fault models and this is a limit of these approaches. Therefore it is necessary a further kind of approach.

## V. Formal Approaches

Formal approaches offer a new opportunity for the dependability study of WSNs both before and after its deployment. They are based on formal verification that consists in checking of the correctness of a system taking in account specifications or properties, using formal methods. Until now there is no work that has proven how to use an unique formal approach to perform dependability assessment at design time and runtime.

The formal verification is performed by providing a proof on an abstract mathematical model of the system. Typically to model systems we can consider labeled transition systems, timed automata, finite state machines, Petri nets, process algebra, hybrid automata, formal semantics of programming languages such as axiomatic semantics, operational semantics and denotational semantics.

In this section we focus on main formal approaches proposed in literature such as *model checking* and *Event Calculus*. Then we discuss about papers in which formal methods have been applied for dependability assessment of WSNs.

### A. Model Checking

One of the well known formal approaches is *model checking* [40]. This technique consists of a systematically exhaustive exploration of the mathematical model (this is possible for finite models, but also for some infinite models where infinite sets of states can be effectively represented finitely by using abstraction or taking advantage of symmetry). Usually this consists of exploring all states and transitions in the model, by using smart and domain-specific abstraction techniques to consider whole groups of states in a single operation and reduce computing time. Implementation techniques include state space enumeration, symbolic state space enumeration, abstract interpretation, symbolic simulation, abstraction refinement. The properties to be verified are often described in temporal logics, such as linear temporal logic (LTL) or computational tree logic (CTL) [41]. The great advantage of model checking is that it is often fully automatic; its primary disadvantage is that it does not in general scale to large systems; symbolic models are typically limited to a few hundred bits of state, while explicit state enumeration requires the state space being explored to be relatively small.

Typically model checking allows to verify if a defined property of a system is satisfied. Thus, the limit of this technique is related to the prediction of a sequence of events. In other words, by means of model checking, an user is able to control if, given an event, the correctness properties are satisfied but is not able to know what will be the behavior of the system after that given event (e.g. node crash or packet loss).

### B. Event Calculus

*Event Calculus* was proposed for the first time in 1986 by Marek Sergot and Robert Kowalski [42] and then it was extended by Murray Shanahan and Rob Miller in the 1990s [43]. This language belongs to the family of logical languages and it is commonly used for representing and reasoning of the events and their effects [44]. *Fluent*, *event* and *predicate* are the basic concepts of Event Calculus [45]. For every timepoint, the value of fluents or the events that occur can be specified.

This language is also named *narrative-based*: in the Event Calculus, there is a single time line on which events occur and this event sequence represents the *narrative*.

The most important and used predicates of Event Calculus are: *Initiates*, *Terminates*, *HoldsAt* and *Happens*.

Since the normal and failing behavior of a WSN can be characterized in terms of an event flow (for instance, a node is turned on, a packet is sent, a packet is lost, a node stops to work due to crash or battery exhaustion, or it gets isolated from the rest of the network due to the failure of other nodes, etc.), Event Calculus, that is an event-based formal language, can be used to formally specify the occurrence of such events and the response of the WSN to them, to check if given correctness properties are verified. Moreover dependability metrics can be valuated by analyzing the *narrative* generated by a Event Calculus reasoner based on the specification of the target WSN.

Finally several techniques are considered to perform automated reasoning in Event Calculus, such as *satisfiability solving*, *first-order logic automated theorem proving*, *Answer Set Programming (ASP)* and logic programming in *Prolog*.

To check the proposed correctness properties defined in Event Calculus, the most common adopted reasoner is the *Discrete Event Calculus (DEC) Reasoner*. The DEC Reasoner [46], [47] uses satisfiability (SAT) solvers [48] and by means of this we are able to perform reasoning like deduction, abduction, post-diction, and model finding. It is documented in details in [49] in which its syntax is explained (e.g. the meaning of the symbols used in the formulas).

### C. Formal approaches for WSN

Lifetime of WSN is defined and evaluated in [50] by means of a mathematical formalism. In this work a generic definition of sensor network lifetime is presented and it is conceived in such way to incorporate different application requirements, such as i) number of alive nodes, ii) time latency in the delivery process, iii) delivery ratio, iv) connectivity, v) coverage, and vi) availability.

Recently, different formal methods and tools have been applied for the modeling and analysis of WSNs, such as [51], [52] and [53].

In [51] authors apply a formal tool to wireless sensor networks, *MEDAL*. They propose a formal language to specify the WSN and a tool to simulate it. However, the formal specification has to be rewritten if the WSN under study changes.

In [52] authors propose a methodology for modeling, analysis and development of WSNs using a formal language (PAWSN) and a tool environment (TEPAWSN). They consider only power consumption as dependability metric that is necessary but not sufficient to assess the WSN dependability (e.g. other problems of WSN such as the isolation problem of a node have been analyzed) and also they apply only simulation.

In [53] authors describe a model-driven performance engineering framework for WSNs (called Moppet). This framework uses the Event Calculus formalism to estimate the performance of WSN applications in terms of power consumption and lifetime of each sensor node; other dependability metrics like coverage, connection resiliency and data delivery resiliency are not considered. The features related to a particular WSN have to be set in the framework every time that a new experiment starts.

There are some papers ([54],[55],[56]) that have considered the formal method in real-time contexts.

In [54] authors model and study WSN algorithms using the Real-Time Maude formalism. Though authors adopt this formalism, they use NS-2 simulator to analyze the considered scenarios making the work very similar to simulative approaches.

The work presented in [55] describes a new formal model for the specification and the validation of WSN. Authors assert the use of rigorous formal method in specification and validation can help designers to limit the introduction of potentially faulty components during the construction of the system. They consider a WSN as a Reactive Multi-Agent System consisting of concurrent reactive agents. In this paper dependability metrics are not treated and calculated and authors just describe the structure of a Reactive Decisional Agent by means of a formal language. Also, no case studies are reported to validate their proposal.

Patrignani et al. in [56] consider policies to monitor wireless sensor network applications in a WSN middleware characterized by a Component and Policy Infrastructure (CaPI); by means of a formalization they are able to catch dangerous or undesired effects which may compromise the correct behavior of a WSN application. In this work it has been developed a prototype that operates on the basis of a application topology in terms of communicating nodes and a set of properties to satisfy. Even if authors confirm that one of the most important benefits of formal approach is that problems occurring at runtime can be detected, they model a static and not dynamic network configuration, focusing only on security (encryption and decryption messages) and resource usage problems and in their scenario they do not consider other dependability metrics (coverage, data delivery resiliency, ...).

In [57] a methodology to investigate the correctness of the design of a WSN from the point of view of its dependability is proposed. The methodology is based on the *event calculus* formalism and it is backed up by a support tool aimed to simplify its adoption by system designers. The tool allows to specify the target WSN in a user-friendly way and it is able to generate automatically the event calculus specifications used to check correctness properties and evaluate dependability metrics such as coverage and connection resiliency but not data delivery resiliency and power consumption.

TABLE II
APPROACH CLASSIFICATION

| Approach | Assessment | |
|---|---|---|
| | Design time | Runtime |
| **Experimental** | × | ✓ |
| **Simulative** | ✓ | × |
| **Analytical** | ✓ | × |
| **Formal** | ✓ | ✓ |

## VI. DISCUSSION

All the analyzed work provides interesting methods and/or techniques which give a contribution for the dependability assessment in WSN. These methods have been grouped in four categories: experimental, simulative, analytical and formal.

In table II a classification of the presented approaches is shown. Experimental methods are used to evaluate a real system and therefore they need for a existent prototype; they are useful at runtime since through these methods do experiments directly on the real system from which they collect data. Simulative and analytical may be adopted in the design phase: they model a system and make an estimate of reliability before of the system release. Finally formal methods make use of correctness specifications and they can be used at design time and at runtime too by means of runtime verification techniques.

Moreover, in this section, it is shown and discussed a comparison of the related work presented in the previous sections in which it emerges a lack of a work that allows to perform WSN dependability assessment both at design and at runtime.

In the grid, shown in figure 1, on the rows there is the analyzed work (approaches, tools and models); on the columns there are the properties chosen to highlight the differences.

In particular we have considered the following features:

- *Experimental Approach* to determine if the related work is based on experiments;
- *Simulative Approach* to determine if the related work is based on simulations;
- *Analytical Approach* to determine if the related work is based on analytical models;
- *Formal Approach* to determine if the related work is based on some formal method (e.g. model checking, Event Calculus) and in particular if the work adopts an approach that provides *Separated specifications*: we want to verify if the related work applies a modular solution considering two logical sets of specifications: a general correctness specification, valid independently of the particular WSN under study, and a structural specification related to the properties of the target WSN (e.g., number of nodes, topology, channel quality, initial battery charge);
- *Design time* to determine if the related work performs dependability assessment at design time;
- *Runtime* to determine if the related work performs dependability assessment at runtime;
- *WSN Dependability metrics* to determine if the related work considers the following dependability metrics: coverage, connection resiliency, data delivery resiliency,

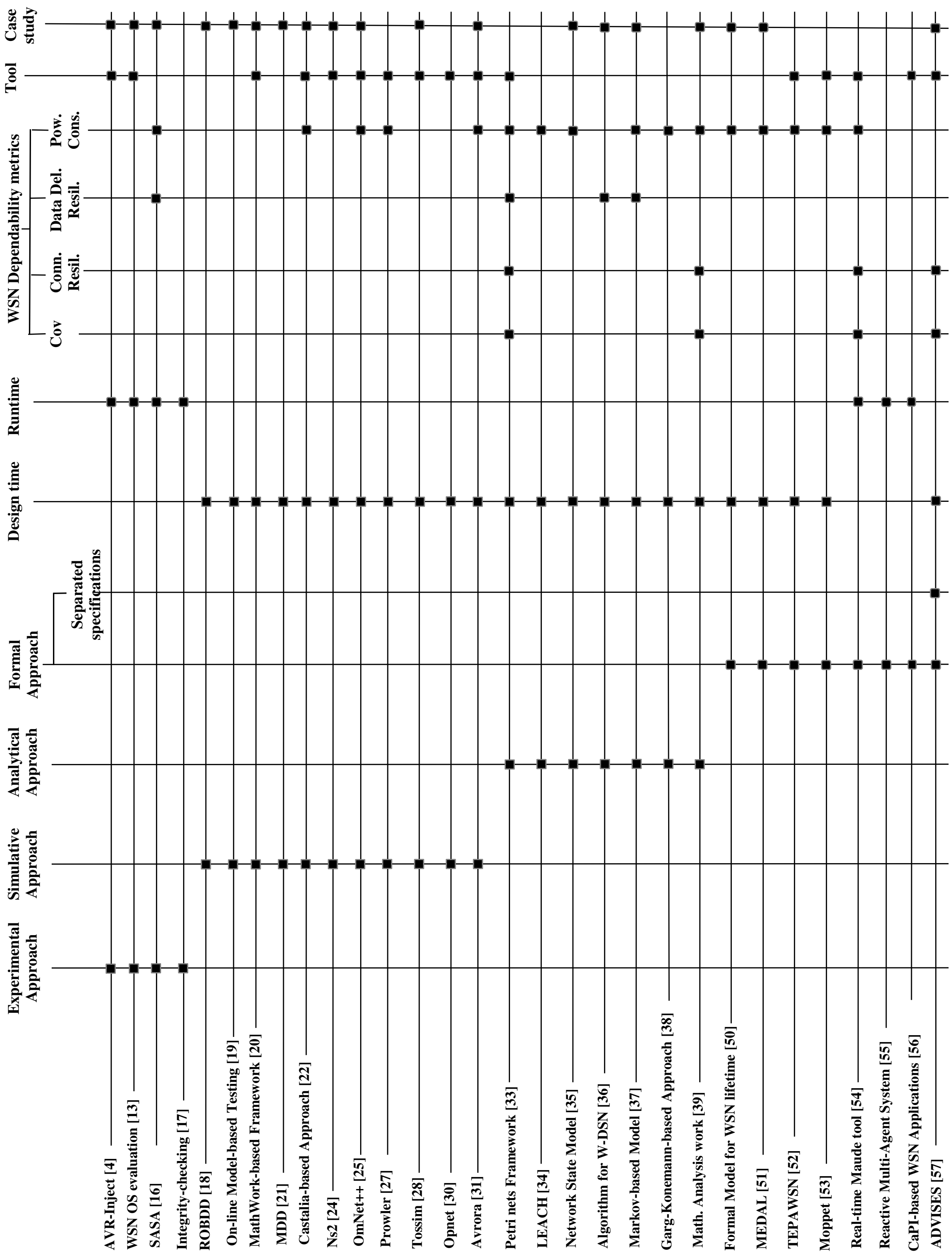


Fig. 1. Comparison of WSN dependability assessment studies

power consumption;
- *Tool* to determine if the related work proposes a novel tool to support designers;
- *Case study* to determine if the related work considers case studies in order to validate the proposed work.

From the survey of the literature it is possible to assert that among the most important dependability metrics, the power consumption is the only one that has been considered extensively, instead data delivery resiliency and connection resiliency are the least analyzed.

The majority of papers propose a tool and present results by means of a case study observing the behavior of the WSN under determined circumstances.

Therefore, looking the figure 1, there is no work that describes a framework conceived in order to perform WSN dependability assessment both at design and runtime measuring all the main dependability metrics. Many studies address the WSN dependability assessment at design time, few studies at runtime.

Moreover, we think that formal methods may be considered as a new and attractive solution for the assessment of dependability both at design time that at runtime by defining one specification for the system suitable for both purposes since the lack of a formal approach that can be applied for doing static and dynamic assessment of WSN dependability remains an open issue.

Thus, in the field of WSN research, a study of a framework that applies an approach to assess WSN dependability by means of a formal approach, before and after the deployment of a WSN, can be advantageous and innovative.

## VII. Conclusions

In this paper, we have reported a survey on the approaches of WSN dependability assessment grouped in experimental, simulative, analytical and formal. What appears clear is that the path towards the production of an optimal approach to check the dependability level of WSN both at design and runtime is still long, and more research effort is needed to reach this compelling goal.

To achieve this goal, we think that applying formal techniques is a good approach since they could join the benefits of the experimental approaches (for dependability evaluation at runtime) and the simulative and analytical approaches (for dependability evaluation at design time). The idea of performing a complete check of the dependability degree on the WSN behavior, to enforce the fulfillment of correctness properties, seems a promising one to achieve more stable and dependable WSN-based systems in the future.

## References


[1] C. Di Martino, G. D'Avino, and A. Testa, "icaas: An interoperable and configurable architecture for accessing sensor networks," *International Journal of Adaptive, Resilient and Autonomic Systems (IJARAS)*, vol. 1, no. 2, pp. 30–45, 2010.

[2] A. Coronato, G. Pietro, J.-H. Park, and H.-C. Chao, "A framework for engineering pervasive applications applied to intra-vehicular sensor network applications," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 137–147, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11036-009-0163-8

[3] A. Coronato, G. De Pietro, and M. Esposito, "A semantic context service for smart offices," in *Hybrid Information Technology, 2006. ICHIT '06. International Conference on*, vol. 2, 2006, pp. 391–399.

[4] M. Cinque, D. Cotroneo, C. Di Martino, S. Russo, and A. Testa, "Avr-inject: A tool for injecting faults in wireless sensor nodes," in *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, may 2009, pp. 1 –8.

[5] M. Cinque, D. Cotroneo, C. Di Martino, and A. Testa, "An effective approach for injecting faults in wireless sensor network operating systems," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*. IEEE, 2010, pp. 567–569.

[6] M. Cinque, "Dependability evaluation of mobile distributed systems via field failure data analysis," Ph.D. dissertation, PhD Thesis, Dipartimento di Informatica e Sistemistica, Universitá di Napoli Federico II, www. mobilab. unina. it/tesiDottorato. html, 2006.

[7] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, "Fault injection techniques and tools," *Computer*, vol. 30, no. 4, pp. 75–82, 1997.

[8] H. Madeira and J. G. Silva, "Xception: Software fault injection and monitoring in processor functional units," in *in Processor Functional Units, DCCA-5, Conference on Dependable Computing for Critical Applications*, 1995, pp. 135–149.

[9] G. Kanawati, N. Kanawati, and J. Abraham, "Ferrari: a flexible software-based fault and error injection system," *Computers, IEEE Transactions on*, vol. 44, no. 2, pp. 248–260, Feb 1995.

[10] J. Barton, E. Czeck, Z. Segall, and D. Siewiorek, "Fault injection experiments using fiat," *Computers, IEEE Transactions on*, vol. 39, no. 4, pp. 575–582, Apr 1990.

[11] D. T. Stott, B. Floering, D. Burke, Z. Kalbarczyk, and R. K. Iyer, "Nftape: A framework for assessing dependability in distributed systems with lightweight fault injectors," in *In Proceedings of the IEEE International Computer Performance and Dependability Symposium*, 2000, pp. 91–100.

[12] J. Arlat, M. Aguera, L. Amat, Y. Crouzet, J.-C. Fabre, J.-C. Laprie, E. Martins, and D. Powell, "Fault injection for dependability validation: a methodology and some applications," *Software Engineering, IEEE Transactions on*, vol. 16, no. 2, pp. 166–182, Feb 1990.

[13] M. Cinque, C. D. Martino, and A. Testa, "Analyzing and modeling the failure behavior of wireless sensor networks software under errors," in *IWCMC*, 2012, pp. 1136–1141.

[14] G. Carrozza and M. Cinque, "Modeling and Analyzing the Dependability of Short Range Wireless Technologies via Field Failure Data Analysis," *Journal of Software*, vol. 4, pp. 707–716, 2009.

[15] G. Carrozza, "Software faults diagnosis in complex ots-based critical systems," Ph.D. dissertation, PhD Thesis, Dipartimento di Informatica e Sistemistica, Universitá di Napoli Federico II, www. mobilab. unina. it/tesiDottorato. html, 2008.

[16] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, no. 2, pp. 10:1–10:29, Apr. 2009. [Online]. Available: http://doi.acm.org/10.1145/1498915.1498916

[17] S. Pennington, T. Bauge, and B. Murray, "Integrity-checking framework: An in-situ testing and validation framework for wireless sensor and actuator networks," in *Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on*, 2009, pp. 575–579.

[18] A. Shrestha, L. Xing, and H. Liu, "Infrastructure communication reliability of wireless sensor networks," in *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*, 29 2006-oct. 1 2006, pp. 250 –257.

[19] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "On-line fault detection of sensor measurements," in *Sensors, 2003. Proceedings of IEEE*, vol. 2, oct. 2003, pp. 974 – 979 Vol.2.

[20] M. Mozumdar, F. Gregoretti, L. Lavagno, L. Vanzago, and S. Olivieri, "A framework for modeling, simulation and automatic code generation of sensor network application," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, june 2008, pp. 515 –522.

[21] R. Shimizu, K. Tei, Y. Fukazawa, and S. Honiden, "Model driven development for rapid prototyping and optimization of wireless sensor network applications," in *Proceedings of the 2nd Workshop on Software Engineering for Sensor Network Applications*, ser. SESENA '11. New York, NY, USA: ACM, 2011, pp. 31–36. [Online]. Available: http://doi.acm.org/10.1145/1988051.1988058

[22] K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostarda, and H. Muccini, "Path loss effect on energy consumption in a wsn," in *Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on*, march 2012, pp. 569 –574.

[23] A. Boulis, "Castalia: revealing pitfalls in designing distributed algorithms in wsn," in *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2007, pp. 407–408.

[24] "The Network Simulator NS-2," http://www.isi.edu/nsnam/ns/.

[25] C. Mallanda, A. Suri, V. Kunchakarra, S. S. Iyengar, R. Kannan, A. Durresi, and S. Sastry, "Simulating Wireless Sensor Networks with OMNeT++." [Online]. Available: http://csc.lsu.edu/sensor_web/final%20papers/OMNet++-IEEE-Computers.pdf

[26] K. Y. Yazdandoost, K. Sayrafian-Pour *et al.*, "Channel model for body area network (ban)," *IEEE P802*, vol. 15, 2009.

[27] A. K. Sharma and D. Gupta, "Performance evaluation of routing protocols for wsns based on energy-aware routing with different radio models," *International Journal of Computer Applications*, vol. 3, no. 12, pp. 6–14, July 2010, published By Foundation of Computer Science.

[28] P. Levis and N. Lee, *Tossim: A simulator for tinyos networks*, 2003, p. 24.

[29] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 126–137. [Online]. Available: http://doi.acm.org/10.1145/958491.958506

[30] P. Jurčík and A. Koubâa, "The ieee 802.15.4 opnet simulation model: Reference guide v2.0," [online] http://www.open-zb.net/publications/HURRAY_TR_070509.pdf, IPP-HURRAY!, Tech. Rep., May 2007.

[31] B. L. Titzer, D. K. Lee, and J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147768

[32] C. Di Martino, "Resiliency assessment of wireless sensor networks: a holistic approach," Ph.D. dissertation, PhD Thesis, Dipartimento di Informatica e Sistemistica, Universitá di Napoli Federico II, www. mobilab. unina. it/tesiDottorato. html, 2009.

[33] C. Di Martino, M. Cinque, and D. Cotroneo, "Automated generation of performance and dependability models for the assessment of wireless sensor networks," *IEEE Trans. Comput.*, vol. 61, no. 6, pp. 870–884, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1109/TC.2011.96

[34] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, jan. 2000, p. 10 pp. vol.2.

[35] A. F. Mini, B. Nath, and A. A. F. Loureiro, "A probabilistic approach to predict the energy consumption in wireless sensor networks," in *In IV Workshop de Comunicao sem Fio e Computao Mvel. So Paulo*, 2002, pp. 23–25.

[36] H. AboElFotoh, S. Iyengar, and K. Chakrabarty, "Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures," *Reliability, IEEE Transactions on*, vol. 54, no. 1, pp. 145 – 155, march 2005.

[37] C.-F. Chiasserini and M. Garetto, "Modeling the performance of wireless sensor networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, march 2004, pp. 4 vol. (xxxv+2866).

[38] J. Stanford and S. Tongngam, "Approximation algorithm for maximum lifetime in wireless sensor networks with data aggregation," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on*. IEEE, 2006, pp. 273–277.

[39] J.-J. Lee, B. Krishnamachari, and C.-C. J. Kuo, "Impact of energy depletion and reliability on wireless sensor network connectivity," in *Proceedings of SPIE*, vol. 5440, 2004, pp. 169–180.

[40] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu, *Bounded model checking*. Elsevier, 2003, vol. 58.

[41] T. Hafer and W. Thomas, "Computation tree logic ctl and path quantifiers in the monadic theory of the binary tree," *Automata, Languages and Programming*, pp. 269–279, 1987.

[42] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Gen. Comput.*, vol. 4, no. 1, pp. 67–95, Jan. 1986. [Online]. Available: http://dx.doi.org/10.1007/BF03037383

[43] R. Miller and M. Shanahan, "Reasoning about discontinuities in the event calculus," in *in Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*. Morgan Kaufmann, 1996, pp. 63–74.

[44] F. Van Harmelen, V. Lifschitz, and B. Porter, *Handbook Of Knowledge Representation*, ser. Foundations of Artificial Intelligence. Elsevier, 2008. [Online]. Available: http://www.worldcat.org/isbn/0444522115

[45] M. Shanahan, "The Event Calculus Explained," *Lecture Notes in Computer Science*, vol. 1600, pp. 409–430, 1999. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.3267

[46] E. T. Mueller, "Event calculus reasoning through satisfiability," *Journal of Logic and Computation*, vol. 14, p. 2004, 2004.

[47] E. Mueller, "Decreasoner," http://decreasoner.sourceforge.net.

[48] E. T. Mueller, "A tool for satisfiability-based commonsense reasoning in the event calculus," in *FLAIRS Conference'04*, 2004, pp. –1–1.

[49] E. T. Muller, "Discrete event calculus reasoner documentation," p. http://decreasoner.sourceforge.net/csr/decreasoner.pdf, 2008.

[50] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *Communications Letters, IEEE*, vol. 9, no. 11, pp. 976 – 978, nov. 2005.

[51] K. Kapitanova and S. Son, "Medal: A compact event description and analysis language for wireless sensor networks," in *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, 2009, pp. 1–4.

[52] K. L. Man, T. Vallee, H. Leung, M. Mercaldi, J. van der Wulp, M. Donno, and M. Pastrnak, "Tepawsn - a tool environment for wireless sensor networks," *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, pp. 730–733, May.

[53] P. Boonma and J. Suzuki, "Moppet: A model-driven performance engineering framework for wireless sensor networks," *Comput. J.*, vol. 53, no. 10, pp. 1674–1690, 2010.

[54] P. Ölveczky and J. Meseguer, "Specification and analysis of real-time systems using real-time maude," *Fundamental Approaches to Software Engineering*, pp. 354–358, 2004.

[55] R. Romadi and H. Berbia, "Wireless sensor network a specification method based on reactive decisional agents," in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, april 2008, pp. 1 –5.

[56] M. Patrignani, N. Matthys, J. Proenca, D. Hughes, and D. Clarke, "Formal analysis of policies in wireless sensor network applications," in *Software Engineering for Sensor Network Applications (SESENA), 2012 Third International Workshop on*, june 2012, pp. 15 –21.

[57] A. Testa, A. Coronato, M. Cinque, and J. C. Augusto, "Static verification of wireless sensor networks with formal methods," in *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*. IEEE, 2012, pp. 587–594.