Formal Analysis of a ZigBee-based Routing Protocol for Smart Grids using UPPAAL

Adnan Rashid and Osman Hasan
School of Electrical Engineering and Computer Science
National University of Sciences and Technology
Islamabad, Pakistan
Email: {adnan.rashid,osman.hasan}@seecs.nust.edu.pk

Kashif Saghar
Centers of Excellence in Science and Applied Technology (CESAT),
Islamabad, Pakistan
Email: kashif.saghar@gmail.com

***Abstract*—Smart grid is an emerging technology which integrates the modern communication network to the traditional power grids. The performance and efficiency of the smart grid mainly depends on reliable communication between its different components and in turn on the routing protocols that establish this communication network. ZigBee protocol is a widely used routing protocol in the home area networks of the smart grids. Traditionally, these protocols are analysed using computer simulations and net testing. All these methods are error-prone and thus cannot provide an accurate analysis, which poses a serious threat to the safety-critical domain of smart grids. To guarantee the correctness of analysis, we propose to use model checking for the verification of the ZigBee routing protocol. We used UPPAAL model checker to formally model the ZigBee routing protocol and verified it using the collision avoidance and liveness properties.**

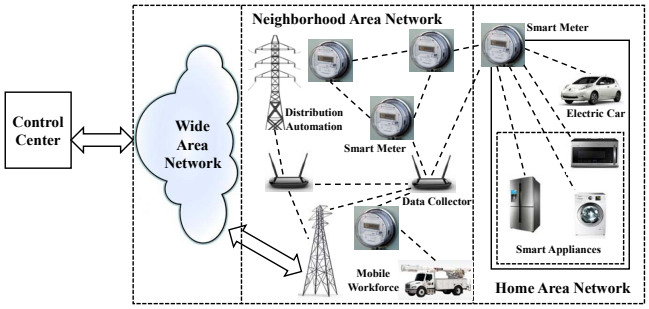

Fig. 1. Smart grid Communication Network

I. Introduction

The traditional power transmission grids cannot cope with the rapid increase in the electricity usage in our daily lives and industry. Some of the foremost issues faced by the traditional grids include the increase in the power outages and blackouts [1], the instability of these systems due to the addition of new users based on the rapid increase in the population and the contribution to the environmental degradation due to carbon emissions [2]. Smart grid technology [3], which is a digital upgrade to the traditional grids, has the tendency to overcome all of the above-mentioned issues while improving the reliability, communication, security, efficiency and safety. The main components of smart grids are control centers, substations, customer premises, utilities and mobile workforce [4]. The efficiency of smart grids highly depends on the information communication and interactivity between these components. Some of the widely used networks in this context include the home area network (HAN), neighbourhood area network (NAN) and wide area network (WAN) [4]. The relationship between these HAN, NAN and WAN is depicted in Figure 1.

Smart grid routing protocols are mainly responsible for the secure and reliable communication over the above-mentioned networks and thus play a vital role in preventing the SG system failures. ZigBee [5] is a widely used wireless routing protocol to maintain a reliable and secure communication between the home appliances and the smart meter in HAN [4]. Traditionally, the analysis of all of these protocols is carried out using simulations and live testing [6]. The Network simulator [7] is a very common simulation tool used for the analysis of routing protocols. However, given their inherent sampling based nature, both of these methods do not ensure a complete absence of functional bugs in the protocols. Moreover, the usage of computer arithmetic based manipulation in simulation adds another aspect of approximation, due to the associated round-off errors, in the analysis results. There could be dire consequences of these inaccurate results due to the safety and financial critical applications of electricity in our daily lives.

Formal methods [8] are capable of overcoming the above-mentioned inaccuracy limitation and have been successfully used to guarantee correctness of many real-world systems. The formal methods based analysis of systems involves the development of a mathematical model of the given system and its verification based on mathematical and deductive reasoning, which increases the chances to catch errors that are often ignored by the traditional techniques. The two major formal analysis techniques include model checking [9] and theorem proving [10]. Model checking is an automatic verification technique that works on a state-space model of the system under analysis. It works quite well for concurrent system of discrete nature but for the continuous systems, it suffers from the state-space explosion problem [9], i.e., the problem of high computational resource and memory requirement to rigorously analyze system models with a huge number of

states. On the other hand, theorem proving allows handling a wide variety of systems by leveraging upon the highly expressive higher-order logic, but it involves a lot of human interaction and lacks automation when dealing with the more expressive higher-order logic. The fact that the model checking is used for the verification of concurrent systems and it is an automatic verification method makes it the most appropriate formal verification method for the accurate analysis of the communication network protocols.

To the best of our knowledge, no prior work regarding the formal analysis of SG ZigBee routing protocol exists so far. In order to fill this gap, we propose to use the UPPAAL model checker [11], which is a widely used tool for the analysis and verification of the real-time systems, specifically the communication protocols, to ensure accurate results in the domain of routing protocol analysis of smart grids. For illustrating the effectiveness of this idea, we verified the collision avoidance and the liveness properties of the ZigBee routing protocol used in the context of smart grids.

The remainder of the paper is organized as follows: Section II presents the related work. We provide a brief overview of the formal verification and the UPPAAL model checker in Section III. Section IV describes the ZigBee routing protocol that is used in the context of smart grids. In Section V, we present our formal modeling and verification results obtained using UPPAAL. Finally, Section VI concludes the paper.

II. Related Work

Model checking has been extensively used to verify various smart grid communication network protocols. Fehnker et al. presented the formal analysis of the LMAC protocol, which is a medium access control (MAC) protocol used in multi-hop wireless sensor network (WSN) [12]. The authors used the UPPAAL model checker for the verification of the protocol and collisions between the packets were detected and resolved. Similarly, Fehnker et al. also carried out the formal analysis of the ad hoc on-demand distance vector (AODV) routing protocols used in wireless mesh network (WMN) [13]. The authors modeled the AODV routing protocol in the process algebra AWN and obtained the equivalent UPPAAL model from the AWN model and verified the properties depicting the dynamic topologies. Hofner et al. performed the formal quantitative analysis of AODV and DYMO protocol for wireless mesh networks using the statistical model checker SMC-UPPAAL [14].

Kwiatkowska et al. proposed to model and analyze the IEEE 802.11 wireless local area network (WLAN) protocol using the probabilistic model checker PRISM [15]. The authors first constructed a probabilistic timed automaton model and then verified the finite-state Markov decision process which is obtained from automaton model via property-preserving semantic. Similarly, Ballarini et al. presented the verification of the S-MAC protocol for WSN using the PRISM model checker [16]. It was ensured that WSNs are optimal with respect to energy consumption.

Renesse et al. presented the modeling and verification of the wireless ad-hoc routing protocol (WARP) by using the SPIN model checker [17]. Similarly, Bhargavan et al. carried out the verification of the routing information protocol (RIP) and AODV routing protocol using the HOL theorem prover along with the SPIN model checker [18]. For more details, we would refer the interested reader to a very comprehensive survey on the application of formal methods to the communication networks e.g., network routing protocols [19]. To the best of our knowledge, no formal verification method has been used to verify the ZigBee protocol in the context of smart grids, which is the main focus of this paper.

III. Preliminaries

A brief introduction to model checking and the UPPAAL model checker is presented in the following subsections to facilitate the understanding of the rest of the paper.

A. Model Checking

Model checking [20], i.e., a mainstream formal verification technique, is widely used for the analysis of the communication networks and protocols. In this method, the model of the system, in the form of state space or automata, and the intended properties of the system are given to the model checker. The verification of the system based on these properties is done automatically and exhaustively. In case, if a system property fails, the model checker provides an error trace depicting the possible error. When the state-space model of the system becomes very large, it becomes difficult to explore all the state-space due to the enormous computational time and memory requirements, which results into the state-space explosion problem [9]. This problem can be avoided by abstracting the system model or sometimes by using the efficient symbolic and bounded model checking techniques.

B. UPPAAL Model Checker

UPPAAL [11] is a model checker used for the modeling, simulation and verification of real-time systems. The underlying system's behaviour is modeled as a network of timed automata. Each of the individual automaton in the network are synchronised based on two mechanisms, which are the binary synchronisation channels and the broadcast synchronisation channels. In binary synchronisation channels, one automaton is synchronised with exactly one other automaton, where as in broadcast channel, one automaton is synchronised with all of the other automata that have enabled transitions. These modeling aspects of the UPPAAL make it suitable to be used for the formal verification of a wide range of communication protocols [21].

The UPPAAL model is formally verified using the property specification expressed as a formula of computational tree logic (CTL) [22], i.e., a temporal logic, which provides the formal specification of the system properties using logical, temporal and path operators. The logical operators are conjunction (`&&`), disjunction (`||`), negation (`!`), implication (`->`) and equality (`<->`). The temporal operators are always (`[ ]`),

TABLE I
CTL PROPERTY SPECIFICATION

CTL Property	Meanings
E <> p	There exists a path where property p eventually holds
A [] p	For all paths p always holds
E [] p	There exists a path where p always holds
A <> p	For all paths p will eventually hold
p $\longrightarrow$ q	Whenever p holds q will eventually hold

eventually (<>), next (X) and until (U) whereas forall (A) and thereexists (E) are the path operator. The CTL property specification operators and their meanings are presented in Table I.

IV. ZIGBEE ROUTING PROTOCOL

ZigBee [4] is a wireless networking technology which adapts the IEEE 802.15.4 standard. Its distinguishing characteristics include low data rate, low power, long life battery and operation using the short range frequencies. It is commonly used for communication between the smart appliances, such as refrigerator, air conditioner, lightings, ovens etc. The ZigBee protocol consists of four networking layers stack namely, the physical layer, the medium access control (MAC) layer, the network layer and the application layer. The network and application layer are defined by the ZigBee specification and the MAC and physical layers are defined based on IEEE 802.15.4 standard [23]. For the routing purposes, it uses the master-slave strategy where the ZigBee coordinator acts as master and the smart appliances act as slaves [24]. The master-slave strategy, depicting the architecture of the ZigBee based protocol for smart grid HAN, is shown in Figure 2. The ZigBee coordinator (master) communicates directly to each of the smart appliances (slaves) and it also manages the communications between the slaves by coordinating with them to maintain smooth and reliable communication in the network.

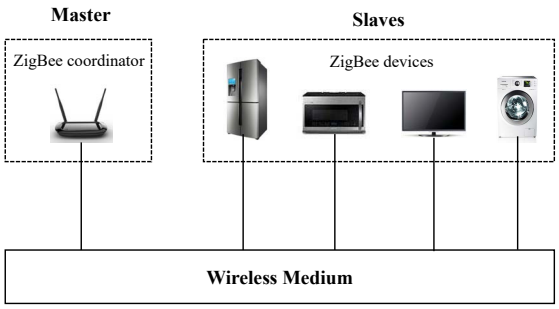

Fig. 2. Zigbee Protocol Architecture

The advantages of ZigBee based network in the context of smart grid HANs are: (i) Low power consumption, which makes the batteries of the ZigBee devices last long. (ii) Highly secured connection and reliable network, which allows avoiding collisions and conflicts between nodes. (iii) Minimum latency of about 15ms to 30ms. (iv) Self-organising network capabilities [25].

V. VERIFICATION IN UPPAAL

In this section, we present the formal modeling and analysis of the ZigBee routing protocol for smart grids.

A. Formal Modeling of ZigBee Routing Protocol

As the ZigBee protocol is based on the master-slave architecture, therefore its UPPAAL model mainly consists of the real-time automaton of master (ZigBee coordinator) and slave (smart appliance). The master process needs to broadcast the message/data to the slaves, therefore we also need a sperate automaton for the medium (wireless channel) in our UPPAAL model. The User automaton is used to capture the sending and receiving capabilities of the slave. A test automaton is used to capture the time limit in the sending and receiving of the data. i.e, the bounded liveness property of the overall protocol system. The assumptions for our design are: (i) The medium under consideration is a lossless communication medium, i.e., every sent data (enquiry/packet) will surely be received. (ii) We consider two slaves for our model.

The automaton of the master process is given in the Figure 3. The master process starts from the first location named `master_1` and sends an enquiry to the first slave. This is represented by a transition from location `master_1` to `master_2`. The variable `data` is defined for the data that needs to be processed. The update `data:=0` in our model means that, the data is an enquiry. The update `slave_number:=1` refers to the first slave. Now, the master will wait until the data has been broadcasted to all of the slaves which is indicated by the synchronisation of the `medium_free?` with the medium. In the next transition, the master will receive the broadcasted data from a slave. To model the assumption of the lossless wireless medium, in the state `master_3`, we defined a clock invariant `t_mas<=3` which shows that the data will be received within 3 time unit. In the transition from location `master_4` to `master_2`, the master will send an enquiry to next slave, which is shown by the increment of shared variable `slave_number`.

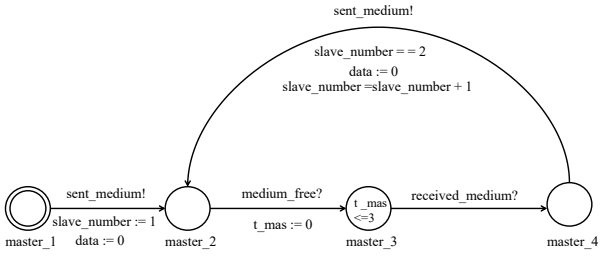

Fig. 3. UPPAAL Model for Master

The UPPAAL model for medium is given in the Figure 4. The medium process receives data by the synchronisation of the input action `sent_medium?` with the master, delays it for couple of time units and broadcasts it by using the UPPAAL feature of the *committed locations*, which ensures that no action can interleave the broadcast. The underlying system will enter into `collision_state` on occurrence of a collision on the medium.

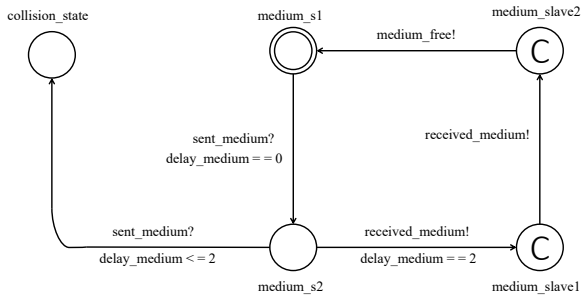

Fig. 4. UPPAAL Model for Medium

Each of the slaves are modeled as individual automaton. However, both the slave automata are same, having couple of different parameters. The automaton of the first slave process is given in Figure 5. The slave process receives data by synchronising with the medium by means of input action `received_medium?`, which is depicted by the transition from `slave1_1` to `slave1_2`. There are two output transitions from the location `slave1_2`. The slave enquires about data by transition from `slave1_2` to `slave1_3` whereas the slave sends data to its user by means of a transition from `slave1_2` to `slave1_4`. In both of these situations, the slave will delay and will not respond to any user for some time units and this behaviour is modeled by the self transitions on location `slave1_3` and `slave1_5`, respectively.

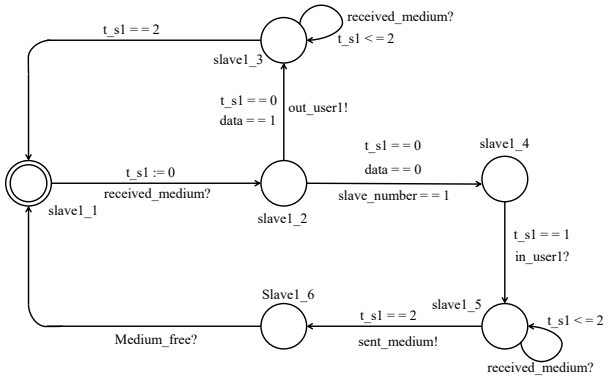

Fig. 5. UPPAAL Model for First Slave

To capture the data sending and receiving, we model a user automata for each of the slave. The user automaton for the first slave process is given in the Figure 6. The user automaton uses the output actions `send_u1` and `received_u1` to send and receive data, respectively.

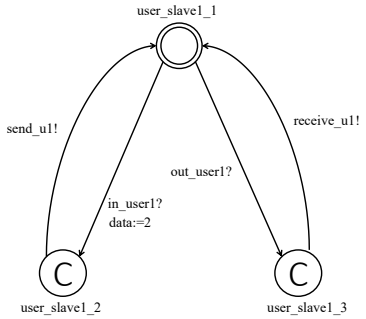

Fig. 6. UPPAAL Model for First Slave's User

Next, in order to check the bounded liveness property of the protocol, we generated a test automaton as shown in the Figure 7. The test automaton uses the clock variable `t_test` to model the time bound on receiving the message. If a sent message is not received within a certain time limit, the test automaton for the underlying system is forced to a state, which is represented by `sending_failed` for the case when data is not sent within time limit, whereas, for the case when data is sent but not received within time bound, then it enters into `receiving_failed`.

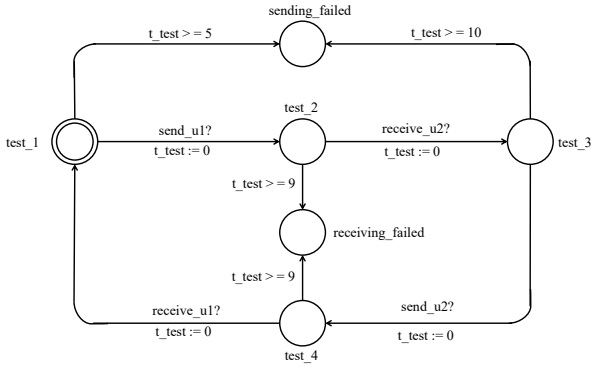

Fig. 7. UPPAAL Model for Test Automaton

B. Formal Verification of ZigBee Routing Protocol

We verified a couple of properties, namely the collision avoidance over wireless medium and the bounded liveness property, to authenticate our design and model. We verified the collision avoidance property by modeling it with the following CTL formula:

$$\forall [\,] \text{ (not medium.collision\_state)}$$

The above property ensures that there will be no collision between the slaves over the wireless medium. Similarly, the

bounded liveness property is modeled by the following CTL formula:

```
∀ [ ] not(Test_Automaton.sending_failed
  or Test_Automaton.receiving_failed)
```

The above property ensures that the delay of user-to-user communication is bounded. i.e., the system will send and receive data within a time limit. Next, we verified the deadlock freeness property for the underlying protocol, which is modeled by the following CTL formula:

```
∀ [ ] not deadlock
```

The above property ensures that the model is deadlock free. Equivalently, this property also authenticates that no collision will occur and the communication delay between the users will be bounded, i.e., the system will not enter in the deadlock states named `collision_state` of medium automaton, `sending_failed` and `receiving_failed` of test automaton. All of the above mentioned properties have been found to be true for our model, which basically authenticates the functionality of the ZigBee routing protocol.

The above formal analysis is of exhaustive nature, which distinguishes it from the traditional simulation techniques that lack this feature due to large number of possibilities. Moreover, the verification process is completely automatic.

VI. Conclusion

This paper presents a formal analysis of the ZigBee routing protocol for HAN of smart grids. The verification approach utilizes the UPPAAL model checker. In order to authenticate our formal model, we verify the collision avoidance and the bounded liveness property of the underlying protocol. The successful verification of these properties show the strength of using model checking, specifically the UPPAAL model checker, to analyze smart grid routing protocols. We plan to verify other smart grid routing protocols, such as WirelessHART [26] and Z-Wave [27] routing protocol for HAN, in future.

References

[1] Jingcheng Gao, Yang Xiao, Jing Liu, Wei Liang, and CL Philip Chen. A Survey of Communication/Networking in Smart Grids. *Future Generation Computer Systems*, 28(2):391–404, 2012.

[2] Ryan Hledik. How Green is the Smart Grid? *The Electricity Journal*, 22(3):29–41, 2009.

[3] Hassan Farhangi. The Path of the Smart Grid. *Power and Energy Magazine, IEEE*, 8(1):18–28, 2010.

[4] Nico Saputro, Kemal Akkaya, and Suleyman Uludag. A Survey of Routing Protocols for Smart Grid Communications. *Computer Networks*, 56(11):2742–2771, 2012.

[5] Patrick Kinney et al. Zigbee Technology: Wireless Control that Simply Works. In *Communications Design Conference*, volume 2, pages 1–7, 2003.

[6] Henrik Lundgren. Implementation and Real-World Evaluation of Routing Protocols for Wireless Ad Hoc Networks. 2002.

[7] Steven McCanne, Sally Floyd, Kevin Fall, and Kannan Varadhan. Network Simulator NS-2, 1997.

[8] Osman Hasan and Sofiène Tahar. Formal Verification Methods. *Encyclopedia of Information Science and Technology, IGI Global Pub*, pages 7162–7170, 2015.

[9] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*, volume 26202649. MIT press Cambridge, 2008.

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

[11] Gerd Behrmann, Alexandre David, and Kim G. Larsen. A Tutorial on UPPAAL. In *Formal Methods for the Design of Real-time Systems*, volume 3185 of *Lecture Notes in Computer Science*, pages 200–236. Springer Berlin Heidelberg, 2004.

[12] Ansgar Fehnker, Lodewijk Van Hoesel, and Angelika Mader. Modelling and Verification of the LMAC Protocol for Wireless Sensor Networks. In *Integrated Formal Methods*, pages 253–272. Springer, 2007.

[13] Ansgar Fehnker, Rob Van Glabbeek, Peter Höfner, Annabelle McIver, Marius Portmann, and Wee Lum Tan. Automated Analysis of AODV Using UPPAAL. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 173–187. Springer, 2012.

[14] Peter Höfner and Annabelle McIver. Statistical Model Checking of Wireless Mesh Routing Protocols. In *NASA Formal Methods*, pages 322–336. Springer, 2013.

[15] Marta Kwiatkowska, Gethin Norman, and Jeremy Sproston. Probabilistic Model Checking of the IEEE 802.11 Wireless Local Area Network Protocol. In *Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, volume 2399 of *LNCS*, pages 169–187. Springer, 2002.

[16] Paolo Ballarini and Alice Miller. Model Checking Medium Access Control for Sensor Networks. In *Leveraging Applications of Formal Methods, Verification and Validation, 2006. ISoLA 2006. Second International Symposium on*, pages 255–262. IEEE, 2006.

[17] Ronan De Renesse and Hamid Aghvami. Formal Verification of Ad-Hoc Routing Protocols using SPIN Model Checker. In *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, volume 3, pages 1177–1182. IEEE, 2004.

[18] Karthikeyan Bhargavan, Davor Obradovic, and Carl A Gunter. Formal Verification of Standards for Distance Vector Routing Protocols. *Journal of the ACM (JACM)*, 49(4):538–576, 2002.

[19] Junaid Qadir and Osman Hasan. Applying Formal Methods to Networking: Theory, Techniques, and Applications. *Communications Surveys & Tutorials, IEEE*, 17(1):256–291, 2015.

[20] Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT press, 1999.

[21] Wang Yi, Paul Pettersson, and Mats Daniels. Automatic Verification of Real-Time Communicating Systems by Constraint-Solving. In *In Proc. of the 7th International Conference on Formal Description Techniques*, pages 223–238, 1994.

[22] Mehmet A Orgun and Wanli Ma. An Overview of Temporal and Modal Logic Programming. In *Temporal Logic*, pages 445–479. Springer, 1994.

[23] Carles Gomez and Josep Paradells. Wireless Home Automation Networks: A Survey of Architectures and Technologies. *IEEE Communications Magazine*, 48(6):92–101, 2010.

[24] Ayesha Hafeez, Nourhan H Kandil, Ban Al-Omar, T Landolsi, and AR Al-Ali. Smart Home Area Networks Protocols within the Smart Grid Context. *Journal of Communications*, 9(9):665–671, 2014.

[25] Md Zahurul Huq and Syed Islam. Home Area Network Technology Assessment for Demand Response in Smart Grid Environment. In *Universities Power Engineering Conference (AUPEC), 2010 20th Australasian*, pages 1–6. IEEE, 2010.

[26] Jianping Song, Song Han, Aloysius K Mok, Deji Chen, Margaret Lucas, and Mark Nixon. WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS'08. IEEE*, pages 377–386. IEEE, 2008.

[27] Mikhail T Galeev. Catching the Z-Wave. *Embedded Systems Design*, 19(10):28, 2006.