Formal Verification of ZigBee-Based Routing Protocol for Smart Grids

Adnan Rashid
https://orcid.org/0000-0002-9729-584X
National University of Sciences and Technology, Pakistan

Osman Hasan
https://orcid.org/0000-0003-2562-2669
National University of Sciences and Technology, Pakistan

INTRODUCTION

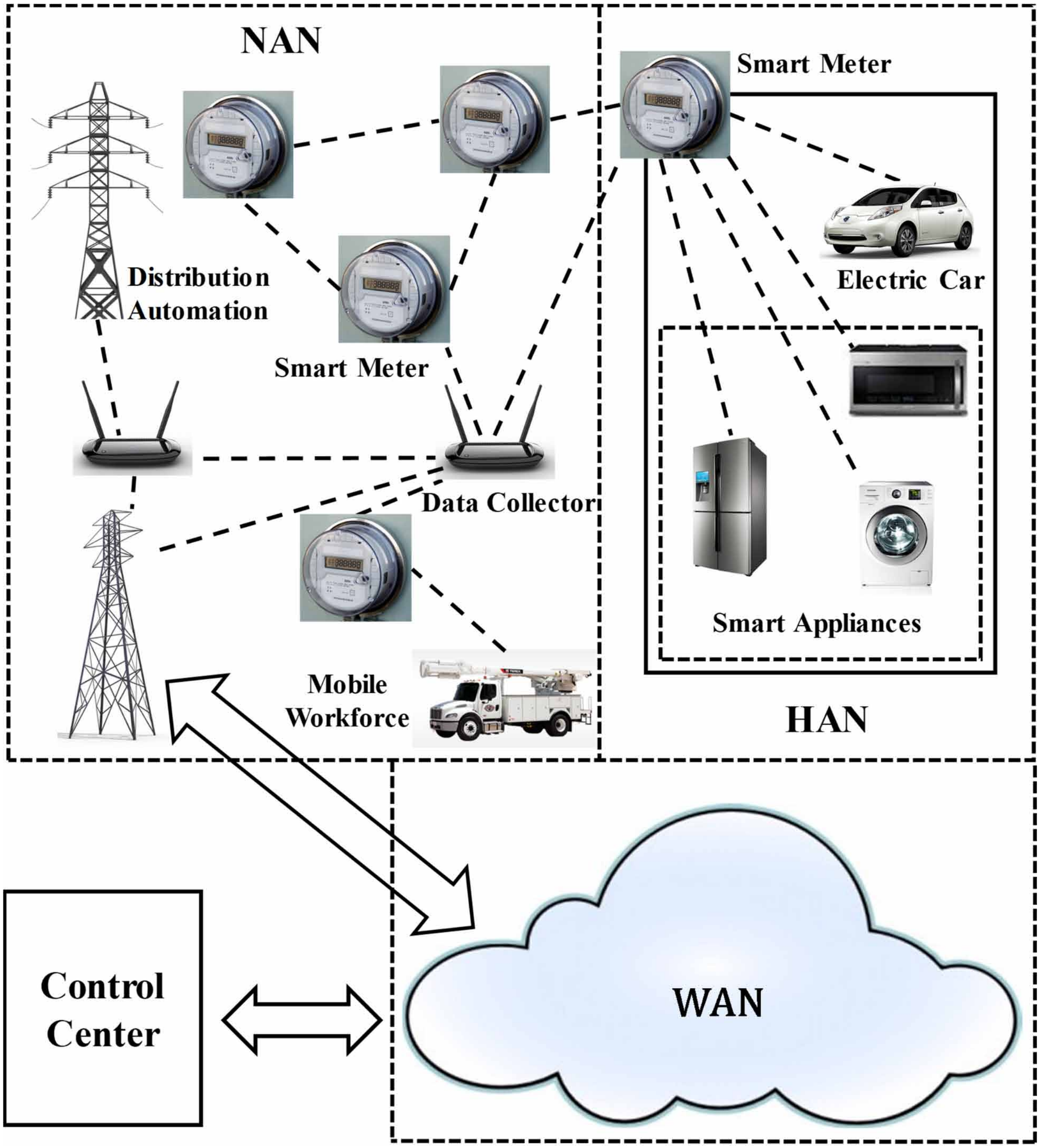
With the growing demand of electricity, the traditional power transmission grids are unable to manage instant upsurges in the utilization of electricity and thus often lead to blackouts and power outages (Gao, Xiao, Liu, Liang & Cheng, 2012). Smart grid technology (Farhangi, 2010), which is a digital upgradation of the traditional grids, can overcome the above-mentioned issues and thus provides reliable, secure, safe and efficient power distribution and management. A smart grid system is mainly composed of control centers, mobile workforce, substations, utilities and the customer premises (Saputro, Akkaya & Uludag, 2012). A reliable communication link between all these building blocks is the core element that governs the efficiency of the overall system. Some of the extensively used communication networks in this technology are wide area network (WAN), neighborhood area network (NAN) and home area network (HAN) (Saputro et at., 2012). For example, HAN is located in the customer premises and thus provides a connection between the smart meter and the home appliances. These networks and their interrelationships are depicted in Figure 1.

The routing protocols play a vital role in establishing a secure and reliable communication link between the components of the above-mentioned smart grid networks. In this regard, the ZigBee (Kinney et al., 2003) routing protocol is commonly used in wireless HAN and provides a reliable, secure and uninterrupted connection between various components of HAN, i.e., home appliances and the smart meter (Saputro et at., 2012). Conventionally, the analysis of these network protocols is performed using live testing and computer-based simulations (Lundgren, 2002) using network simulators (McCanne, Floyd, Fall & Varadhan, 1997). These techniques can analyze a limited number of possible scenarios and thus there is always a chance of missing a corner case, which compromises the accuracy of the analysis. Moreover, round-off errors due to the involvement of the computer arithmetic also introduce some approximations and thus we cannot certify a complete absence of bugs. Therefore, considering the safety-critical applications of power and electricity in our daily tasks, these inaccurate results may lead to many unwanted consequences.

Formal methods (Hasan & Tahar, 2015) have the ability to overcome these inherent limitations of the above-mentioned conventional techniques. The formal methods based system analysis mainly involves developing mathematical model of the given system based on some logic and verification of its intended properties using deductive reasoning. The involvement of the logical modeling and deductive reasoning enhances the chances of catching subtle errors that are often ignored by the traditional methods. For

DOI: 10.4018/978-1-7998-3479-3.ch069

Figure 1. Smart grid Communication Network



example, for a 32-input adder, we require $2^{64} = 1.8446744 \text{ x } 10^{19}$ test vectors in the case of computer-based simulations and it is checked for a limited number of these test vectors to reduce the number of computational overheads. Thus, there is always a chance of missing a bug due to not using the test vector to catch it. However, in the case of formal methods, we need to develop an implementation and specification of the circuit using some appropriate logic and verify that the implementation meets the specifications of the underlying system. The two most commonly used formal methods include model checking (Baier & Katoen, 2008) and theorem proving (Harrison, 2009).

Model checking is an automated analysis technique, which involves the state-space based modeling of the underlying system and rigorously verifying its properties specified in an appropriate logic. The discrete-time systems are commonly verified using this technique due to its state-based nature. Moreover, it has been frequently used for analyzing concurrent systems (Clarke, 1986; Gradara, 2007). However, model checking is not very suitable for systems exhibiting the continuous behavior due to its inherent state-space explosion problem (Baier & Katoen, 2008), i.e., the issues related to the limited amount of computer memory and resources for larger systems.

On the other hand, theorem proving is a formal analysis technique, which involves developing a mathematical model of the system to be analyzed in an appropriate logic and verification of the properties of interest using computer based tools known as theorem provers/proof assistants. Based on the decidability or undecidability of the underlying logic, theorem proving can be termed as automated or interactive. Propositional logic is decidable and thus the sentences/formulas expressed in this logic can be verified automatically. However, one of its main limitations is less expressiveness and thus it cannot be utilized for analyzing all sort of systems. Higher-order logic is the most expressive form of logic and it allows modeling and verifying the complex systems exhibiting the continuous behavior. However, due to the undecidable nature, the verification of theorems about sentences expressed in higher-order logic requires a lot of user guidance in an interactive manner.

Due to the involvement of a lot of human interaction, theorem proving is not preferred for the systems exhibiting the discrete-time behavior. However, the automatic nature of model checking along with the suitability for formally analyzing the concurrent systems makes it an ideal candidate for the verification of network and routing protocols. This is one of the major factors for choosing model checking for the formal verification of the ZigBee-based routing protocol for smart grids.

In this chapter, we use the UPPAAL model checker (Behrmann, David & Larsen, 2004) to formally verify the ZigBee protocol for smart grids. The primary motivation for choosing UPPAAL is that it can handle time in its true form (as clock variable) and the fact that it has been used for modeling and the automatic verification of many systems, in particular the network and routing protocols. To illustrate the effectiveness of our idea, we use our UPPAAL model to verify the collision avoidance, bounded liveness and deadlock freeness properties of the smart grids' ZigBee-based protocol.

RELATED WORK

Model checking has been widely used to verify a variety of communication protocols for smart grids. Fehnker et al. (2007) formally verified a light-weight medium access control (LMAC) protocol, which operates in multi-hop energy-constrained wireless sensor networks (WSNs). Through this exercise, the authors mainly detected and resolved the collision between the data packets using UPPAAL model checker. Similarly, a formal modeling and verification of the ad hoc on-demand distance vector (AODV) protocol operating in wireless mesh network (WMN) is presented in Fehnker et al. (2012). The authors used the algebra for wireless networks (AWN) process algebra to model the AODV routing protocol. Next, this AWN model is converted into its equivalent UPPAAL model, which is further used for verifying its properties based on its different dynamic topologies. Hofner et al. used statistical model checker SMC-UPPAAL for the formal quantitative analysis of AODV and dynamic MANET on-demand (DYMO) protocols for WMNs (Hofner & McIver, 2013).

Malik et al. (2013) proposed a framework for the formal analysis of inter control center communications protocol (ICCP) used in exchanging data and control in different control centers. The authors modeled the protocol using an iterative process and verified its safety and critical security properties using the UPPAAL model checker. Similarly, Tschirner et al. (2008) used UPPAAL to verify the quality-of-service (QoS) of biomedical sensor networks (BSNs). The authors mainly verified the network connectivity, packet delivery and deadlock freeness properties using their UPPAAL model. Somappa et al. (2015) provided the verification of the dual-mode adaptive MAC (DMAMAC) protocol, which is used in process control applications. The authors used UPPAAL to formally verify the state-switching and deadlock freeness properties of the DMAMAC protocol. Similarly, Rashid et al. presented the formal analysis of the ZigBee routing protocol using UPPAAL (Rashid, Hasan & Saghar, 2015) for smart grid applications. The context of this book chapter is mainly focused on this work as, to the best of our knowledge, this is the only work that is related to the verification of communication protocols for smart grids.

Kwiatkowska et al. (2002) used probabilistic model checking to formally analyze the IEEE 802.11 wireless local area network (WLAN) protocol using a two step process. The first step involves the construction of a model based on probabilistic timed automaton. In the next step, the property-preserving semantics are used to convert this automaton model into a finite-state Markov decision process, which is verified using the PRISM model checker. Similarly, Ballarini & Miller (2006) used PRISM to formally analyze the sensor-MAC (S-MAC) protocol for WSN with respect to energy consumption and performance.

Fehnker et al. (2009) proposed a framework to analyze the wireless network protocols, which is based on both the simulation and model checking techniques. The authors first modeled the wireless protocol using CaVi, which is a graphical user interface used for the modeling of networks. In the next step, this CaVi model is translated to the Castalia simulator and PRISM models. Finally, the experiments are performed on the simulator to judge the performance of the protocol. Also, the performance properties of the protocol are verified using the PRISM model checker. Similarly, Rossi et al. (2013) presented formal verification and performance analysis of the cognitive wireless networks based on their discrete time Markov chain (DTMC) models using PRISM.

Bhargavan et al. (2002) formally verified the AODV and routing information protocol (RIP) using the SPIN model checker along with the HOL theorem prover. Similarly, Renesse & Aghvami (2004) used SPIN to verify wireless ad-hoc routing protocol (WARP). Islam et al. (2006) formally modeled the dynamic host configuration protocol (DHCP) using process meta language (PROMELA) and formally verified its model using SPIN. A detailed account of the automated formal analysis of routing protocols for WSNs can be found in Chen et al. (2013). Moreover, a survey of the formal analysis of communication network protocols can be found in Qadir & Hasan, (2015).

The area of formal verification of communication network and routing protocols for smart grid systems has not been fully explored yet. There is only a single work reported in the literature, in which the authors presented the formal analysis of the ZigBee routing protocol for smart grids using UPPAAL (Rashid et al., 2015). Therefore, we have chosen to focus on this work in this chapter. Moreover, we have further elaborated the work, which ensures its better understanding to the non-formal methods community and thus addresses a wider audience.

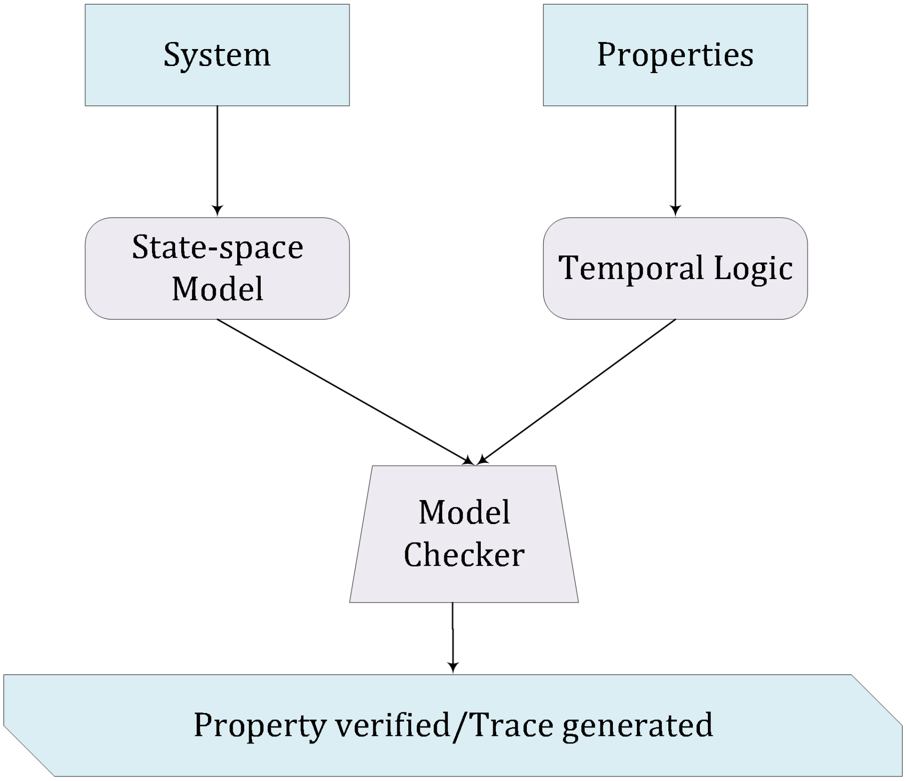
BACKGROUND

In this section, we briefly describe model checking and the UPPAAL model checker to facilitate the understanding of the formal modeling and verification of the ZigBee-based routing protocol presented later in the chapter.

Model Checking

Model checking (Clarke et al., 1999) is one of the major formal techniques, which is commonly used for analyzing concurrent systems, such as network and routing protocols. The model checking-based analysis involves the construction of a state-space model of the given system and specification of the intended properties of the system in temporal logic (Pnueli, 1977), as depicted in Figure 2. Next, both the model and properties are given to the model checker, which explores the state-space exhaustively and automatically verifies the given system based on these properties. In the case of failure of a property, the model checker generates an error-trace, which helps a user to identify and rectify the error found in the system's model. For the larger systems, this technique is subject to the problem of state-space explosion (Baier & Katoen, 2008) caused by the limited availability of computer memory and other computational resources. However, the abstraction of the model or usage of the efficient bounded and symbolic model checking techniques can avoid this problem.

Figure 2. Model Checking



UPPAAL Model Checker

UPPAAL (Behrmann et al., 2004) model checker is extensively used to formally analyze real-time systems. The behavior of the given system is represented as a network of timed automata. The synchronization between each of the individual automaton is done using the phenomenon of channels, which are of two types: namely binary and broadcast synchronization channels. The binary channel is used for synchronizing one automaton with the other, whereas, in the case of broadcast channel, a single automaton is synchronized will all the channels with enabled transitions using *committed locations* in UPPAAL. These modeling and verification features enable UPPAAL as a commonly used tool to formally analyze concurrent systems, such as communication network and routing protocols (Yi, Pettersson & Daniels, 1994).

We can formally verify the UPPAAL model of a system using the properties specified in computational tree logic (CTL) (Orgun & Ma, 1994), which are the formulas in temporal logic and support specifications based on temporal, path and logical operators. The temporal operators involve next (X), always ([ ]) and eventually (< >). Similarly, the path operators are thereexists (E) and forall (A). Whereas, negation (!), disjunction (||), conjunction (&&) and implication (®) are the logical operators. Table 1 presents the CTL property specification and their graphical representation.

ZIGBEE ROUTING PROTOCOL

ZigBee (Saputro et al., 2012) is a technology, which resides at the top of networking standard IEEE 802.15.4 and is widely used in HANs for smart grids. It exhibits many interesting characteristics, like low power, long battery life and low data rates. These distinguishing features make it a suitable candidate to operate in short range frequencies. In the HAN, ZigBee is mainly responsible for an uninterrupted communication between various smart appliances, i.e., lightings, air conditioner, refrigerator, oven and fans etc. It is build up of four layers that are stacked over each other in the network and include MAC, physical, network and application layers. IEEE 802.15.4 standard is used to define the physical and MAC layers, whereas the application and network layers are based on ZigBee specification (Gomez & Paradells, 2010). ZigBee-based routing protocol uses the master-slave strategy for routing purposes and its architecture is depicted in Figure 3. Here, the ZigBee coordinator is a master and the slaves include various home appliances, i.e., oven, television and refrigerator etc (Hafeez, Kandil, Al-Omar, Landolsi & Al-Ali, 2014). The master (ZigBee coordinator) is mainly responsible for establishing a reliable and uninterrupted communication in the network and thus it directly communicates with all slaves (home appliances) and also manages their intercommunication by coordinating with the home appliances.

The ZigBee-based routing protocol provides the following advantages in HANs:

1. It consumes low power and helps the batteries to last longer for the ZigBee devices present in the network.
2. It helps in avoiding conflicts and collision between different nodes by providing a reliable and highly secured network.
3. It has a very low latency, i.e., about *15*ms to *30*ms.
4. It has abilities of self-organization in a network (Huq & Islam, 2010).

Table 1. Property Specification in Computational Tree Logic (CTL)

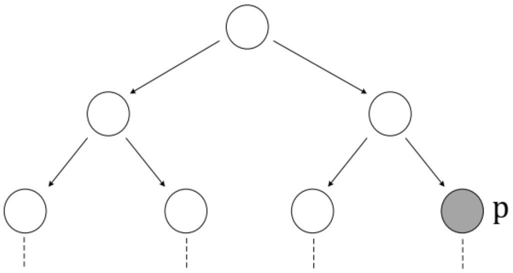
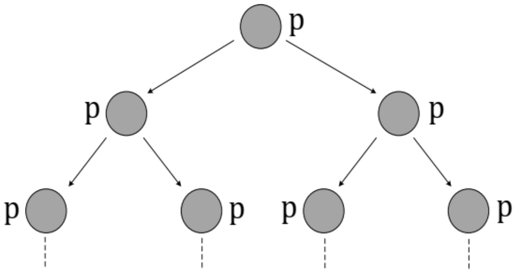
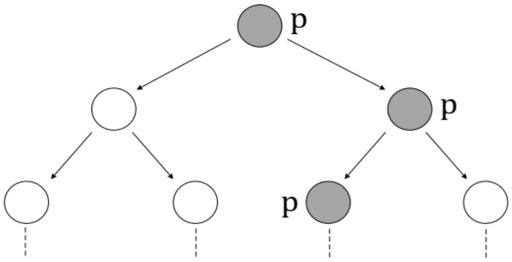
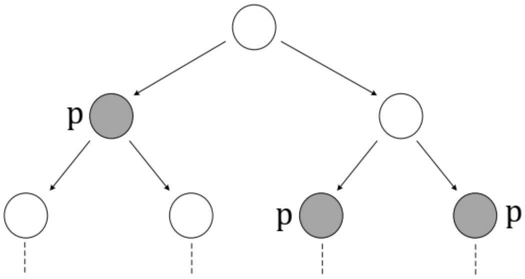
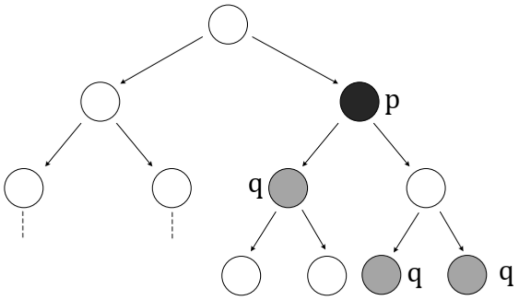
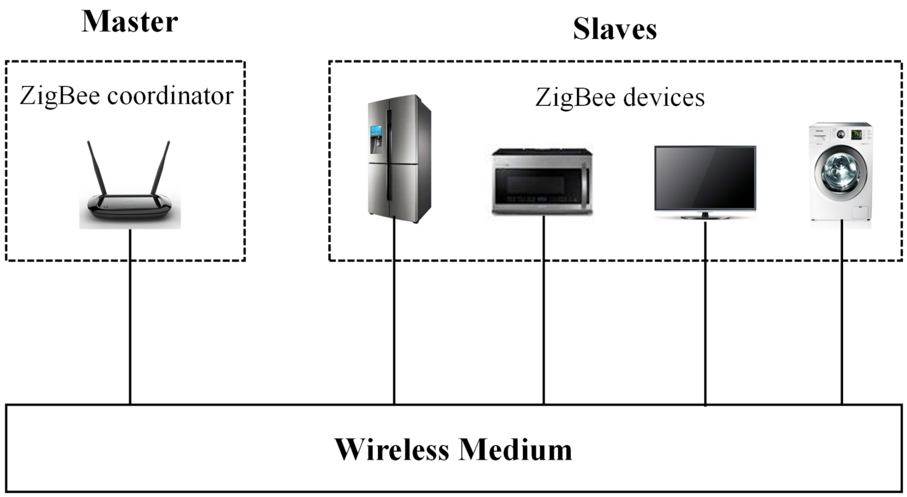
CTL Property	Description	Graphical Representation
E < > p	There exists a path where the property p eventually holds	
A [ ] p	p always holds for all the paths	
E [ ] p	There exists a path where p always holds	
A < > p	p will eventually hold for all paths	
P ® q	Whenever the property p holds q will eventually hold	

Figure 3. Architecture of Zigbee Protocol



FORMAL MODEL FOR ZIGBEE-BASED ROUTING PROTOCOL

The ZigBee-based routing protocol uses the master-slave strategy and thus the main components of its real-time UPPAAL model are the *master* and *slaves* automata modeling the ZigBee coordinator and smart appliances, respectively. The master broadcasts the data to slaves via a wireless channel. Therefore, we require a *medium* automaton that models the channel in our UPPAAL model. The data sending and receiving abilities of a slave are modeled using the *user* automaton. Finally, to model the time bound in the data transmission, we use a *test* automaton, which is later used to verify the bounded liveness property of the underlying system. We consider the following assumptions for our UPPAAL model:

1. Lossless medium for the communication between various components, i.e., every data sent on the channel will surely be received.
2. Two slaves are considered for our UPPAAL model.

The timed automaton corresponding to the master process is depicted in Figure 4. The master automaton initiates the process by sending an enquiry to the first slave, which is represented by the initial transition, i.e., mas_1 to mas_2. The data (enquiry/message) that needs to be transmitted, is modeled by the variable data and an enquiry is represented by data:=0. Similarly, the update sla_num:=1 represents the first slave. Now, the channel synchronization med_free? is used in the next transition to ensure that the master waits until the data is broadcasted to all the slaves. In the third transition, the broadcasted data from a slave is received by the master. We also specified a condition t_mas<=3 working as a clock invariant in the master state mas_3 to ensure that the data is received within 3 time units, which actually models the assumption of the lossless medium. Finally, an increment in the variable sla_num in transition from mas_4 to mas_2 shows that an enquiry is sent to next slave from the master automaton.

The medium automaton is depicted in Figure 5. The data sent by the master is received by the medium process by synchronizing the input channel sen_med? with the master. Next, the medium process broadcasts the received data to all the slaves using the phenomenon of *committed locations*, which enables the broadcast feature for the UPPAAL model. In the case, if a collision is occurred on the medium, the given system will move to coll_state.

Figure 4. The Master Automaton

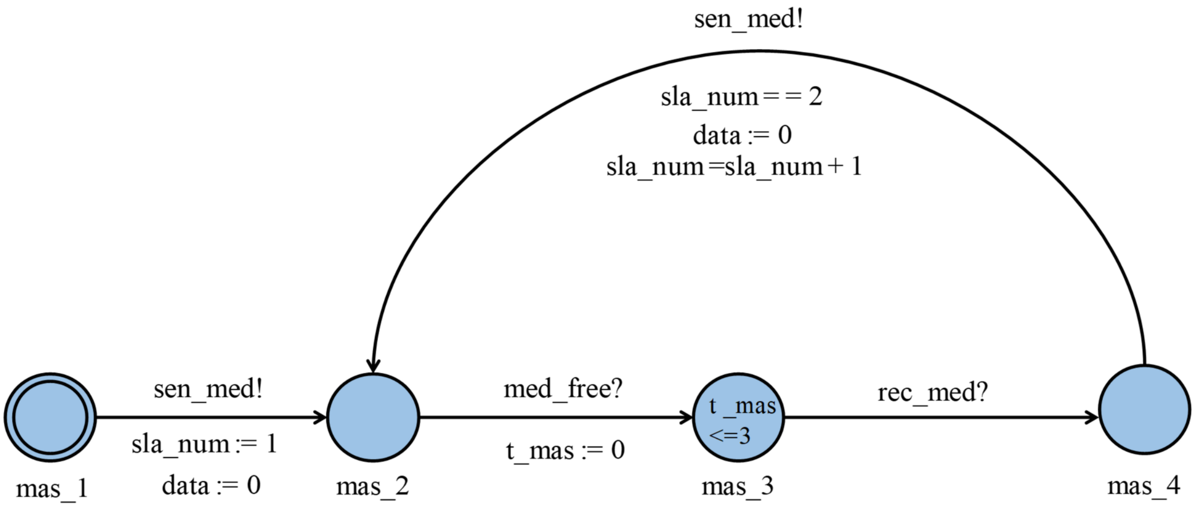
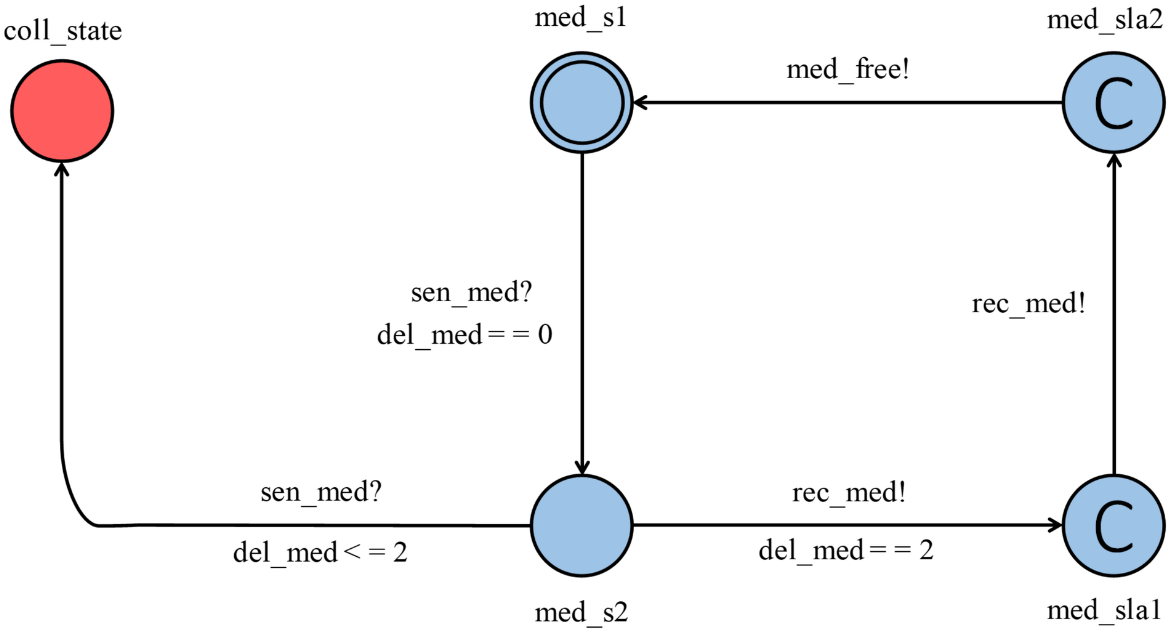
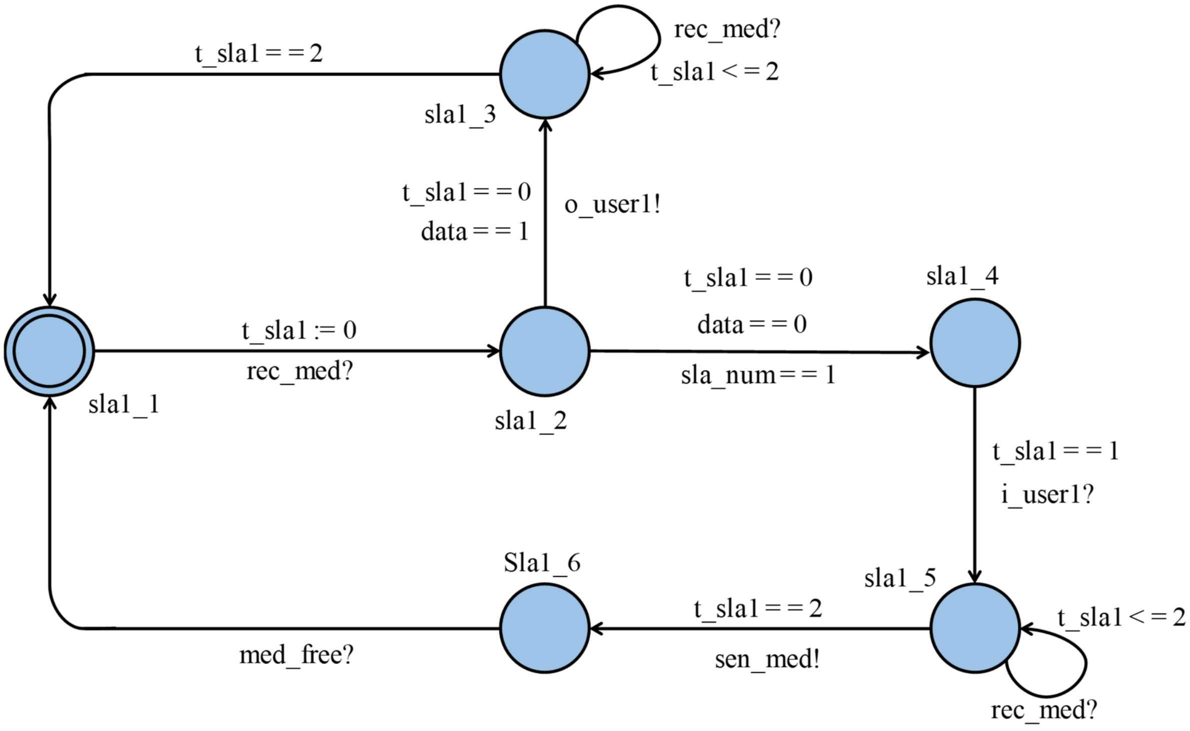


Figure 5. The Medium Automaton



We modeled both the slaves as individual automata, which are almost same, except a couple of different parameters. Figure 6 depicts the automaton for the first slave process. The slave automaton initiates the process from sla1_1 and receives data as a result of synchronization with the medium using input action rec_med?, which is represented by the transition from sla1_1 to sla1_2. Now, the slave automaton at location sla1_2 has two possible transitions. The slave either sends an enquiry to the user by transitioning from sla1_2 to sla1_3 or sends a message by transitioning from sla1_2 to sla1_4. For both the cases, the slave will not answer to any user for a couple of time units, which is illustrated by the self-transitions on sla1_3 and sla1_5, respectively.

Figure 6. The First Slave Automaton



A user automaton is modeled for each of the slave processes, which captures the capabilities of data transmission. Figure 7 depicts the first slave's user automaton. It uses the output actions sen_u1 and rec_u1 for the data transmission.

We used a test automaton, as depicted in Figure 8, to verify the bounded liveness of the ZigBee protocol. It utilizes a clock variable t_tes to express the time bound in the transmission of data. If the transmitted data is not received within a certain time limit, the test automaton will enter into location sen_failed for the case when data is not sent within time limit. In the case, if the data is sent successfully and not received then it will be forced into rec_failed.

FORMAL VERIFICATION OF ZIGBEE-BASED ROUTING PROTOCOL

As the ZigBee-based routing protocol works on the master-slave architecture, it ensures that there is no collision during the data transmission from master to slaves or one slave to the other slave. Moreover, the sent data should be received within a time limit. Both of these properties authenticate our design and formal model based on UPPAAL model checker. We used the following CTL formula to verify the collision avoidance property:

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

Figure 7. First Slave's User Automaton

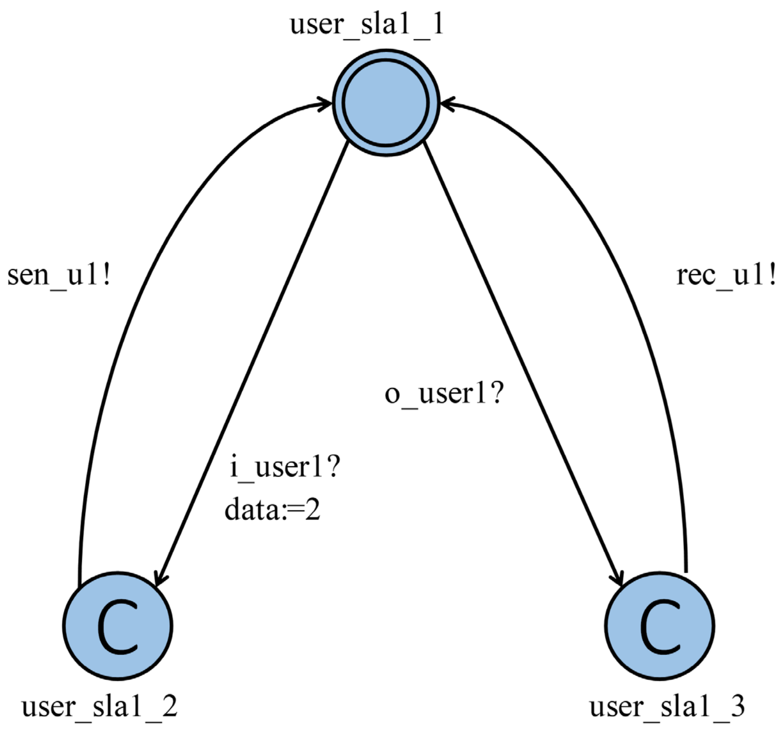
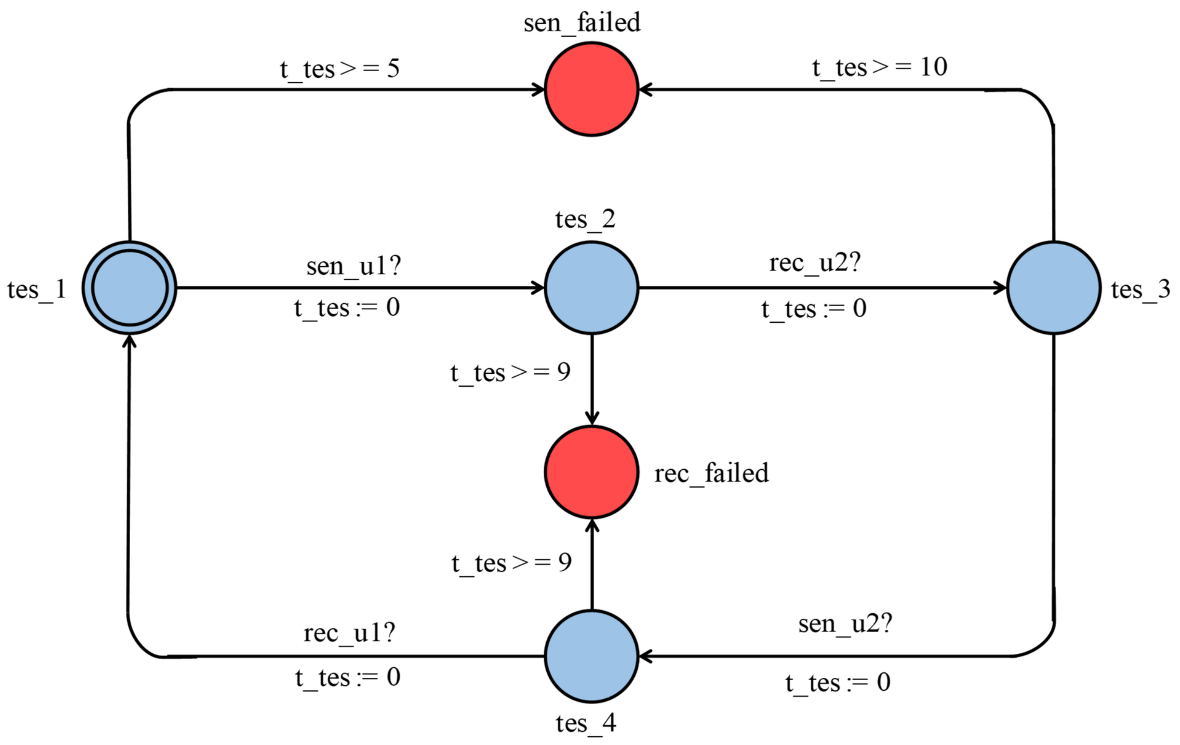


Figure 8. UPPAAL Model for Test Automaton



The verification of the above property guarantees that there would be no collision between the master and the slaves over the transmission channel. Next, we verify the bounded liveness property as:

$$\forall[\ ](\mathrm{T\_Autom.sen\_failed\ or\ T\_Autom.rec\_failed})$$

The verification of the above property certifies that the user-to-user communication delay is bounded. i.e., the system will transmit data within a time limit. Finally, we used the notion of deadlock present in UPPAAL to verify the deadlock freeness of the ZigBee protocol as:

$$\forall[\ ](\mathrm{not\ deadlock})$$

Equivalently, the formal verification of this property also ensures the collision avoidance and bounded liveness properties, i.e., the system will never enter in the deadlock states namely coll_state, sen_failed and rec_failed of the master and test automata, respectively. The verification of all the above-mentioned properties ensures the true functional behavior of the smart grids' ZigBee-based routing protocol.

One of the distinguishing features of our formal analysis presented above is that all the state-space is exhaustively explored during the process of verifying the ZigBee protocol. Whereas, in the case of the simulation-based analysis, only some test vectors are explored, which compromise the accuracy of the analysis. Moreover, the verification is done automatically.

CONCLUSION

In this chapter, we presented the formal verification of the ZigBee-based routing protocol, which is widely used in HANs of smart grids to provide secure and reliable connection between the smart meter and home appliances. The UPPAAL model checker was used to model the master-slave architecture of the protocol using real-time automata. We also verified some vital properties of the ZigBee protocol, namely collision avoidance, bounded liveness and deadlock freeness properties, which ensure the correct functioning of the protocol. Our formal analysis of the ZigBee-based routing protocol using UPPAAL illustrates the advantages of using model checking for analyzing smart grid routing protocols.

FUTURE RESEARCH DIRECTIONS

In future, we plan to verify some other routing protocols for smart grid systems, such as wirelessHART (Song et al., 2008) and Z-Wave (Galeev, 2006) routing protocol for HAN. WirelessHART is a secured time-division multiple access (TDMA) based wireless mesh networking technology. Similarly, Z-Wave is used to integrate sensors and actuators and to perform the smart grid HAN services. The other research direction is to use our proposed approach for formally verifying the routing protocols for WAN and NAN for smart grids.

REFERENCES

Baier, C., & Katoen, J. P. (2008). *Principles of Model Checking*. MIT Press.

Ballarini, P., & Miller, A. (2006). *Model Checking Medium Access Control for Sensor Networks. In Leveraging Applications of Formal Methods, Verification and Validation*. IEEE.

Behrmann, G., David, A., & Larsen, K. G. (2004) A Tutorial on UPPAAL. Formal Methods for the Design of Real-time Systems, (pp. 200–236). Springer Berlin Heidelberg.

Bhargavan, K., Obradovic, D., & Gunter, C. A. (2002). Formal Verification of Standards for Distance Vector Routing Protocols. *Journal of the Association for Computing Machinery*, *49*(4), 538–576. doi:10.1145/581771.581775

Chen, Z., Zhang, D., Zhu, R., Ma, Y., Yin, R., & Xie, F. (2013). A Review of Automated Formal Verification of Ad-hoc Routing Protocols for Wireless Sensor Networks. *Sensor Letters*, *11*(5), 752–764. doi:10.1166l.2013.2653

Clarke, E. M., Emerson, E. A., & Sistla, A. P. (1986). Automatic Verification of Finite-state Concurrent Systems using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems*, *8*(2), 244–263. doi:10.1145/5397.5399

Clarke, E. M., Grumberg, O., & Peled, D. (1999). *Model Checking*. MIT press.

Farhangi, H. (2010). The Path of the Smart Grid. *Power and Energy Magazine, IEEE*, *8*(1), 18–28. doi:10.1109/MPE.2009.934876

Fehnker, A., Fruth, M., & McIver, A. (2009). Graphical Modelling for Simulation and Formal Analysis of Wireless Network Protocols. *Methods. Models and Tools for Fault Tolerance*, *5454*, 1–24. doi:10.1007/978-3-642-00867-2_1

Fehnker, A., Glabbeek, R. V., Hofner, P., McIver, A., Portmann, M., & Tan, W. L. (2012) Automated Analysis of AODV Using UPPAAL. In Tools and Algorithms for the Construction and Analysis of Systems, (pp. 173–187). Springer.

Fehnker, A., Hoesel, L. V., & Mader, A. (2007) Modelling and Verification of the LMAC Protocol for Wireless Sensor Networks. In Integrated Formal Methods, (pp. 253–272). Springer.

Galeev, M. T. (2006). Catching the Z-Wave. *Embedded Systems Design*, *19*(10), 28.

Gao, J., Xiao, Y., Liu, J., Liang, W., & Chen, C. L. P. (2012). A Survey of Communication/Networking in Smart Grids. *Future Generation Computer Systems*, *28*(2), 391–404. doi:10.1016/j.future.2011.04.014

Gomez, C., & Paradells, J. (2010). Wireless Home Automation Networks: A Survey of Architectures and Technologies. *IEEE Communications Magazine*, *48*(6), 92–101. doi:10.1109/MCOM.2010.5473869

Gradara, S., Santone, A., & Villani, M. L. (2007). Formal Verification of Concurrent Systems via Directed Model Checking. *Electronic Notes in Theoretical Computer Science*, *185*, 93–105. doi:10.1016/j.entcs.2007.05.031

Hafeez, A., Kandil, N. H., Al-Omar, B., Landolsi, T., & Al-Ali, A. (2014). Smart Home Area Networks Protocols within the Smart Grid Context. *Journal of Communication*, *9*(9), 665–671. doi:10.12720/jcm.9.9.665-671

Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press. doi:10.1017/CBO9780511576430

Hasan, O., & Tahar, S. (2015). Formal Verification Methods. In Encyclopedia of Information Science and Technology. IGI Global.

Hofner, P., & McIver, A. (2013). *Statistical Model Checking of Wireless Mesh Routing Protocols. In NASA Formal Methods*. Springer.

Huq, M. Z., & Islam, S. (2010). Home Area Network Technology Assessment for Demand Response in Smart Grid Environment. In *Australasian Universities Power Engineering Conference*, (pp. 1–6). IEEE.

Islam, S. M., Sqalli, M. H., & Khan, S. (2006). Modeling and Formal Verification of DHCP Using SPIN. *International Journal of Computer Science and Applications*, *3*(2), 145–159.

Kinney, P. (2003). Zigbee Technology: Wireless Control that Simply Works. *Communications Design*, *2*, 1–7.

Kwiatkowska, M., Norman, G., & Sproston, J. (2002) 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*, (pp. 169–187). Springer. 10.1007/3-540-45605-8_11

Lundgren, H. (2002) *Implementation and Real-World Evaluation of Routing Protocols for Wireless Ad Hoc Networks*. Academic Press.

Malik, S., Berthier, R., Bobba, R. B., Campbell, R. H., & Sanders, W. H. (2013). *Formal Design of Communication Checkers for ICCP using UPPAAL. In Smart Grid Communications*. IEEE.

McCanne, S., Floyd, S., Fall, K., & Varadhan, K. (1997). *Network Simulator NS-2*. Academic Press.

Orgun, M. A., & Ma, W. (1994). An Overview of Temporal and Modal Logic Programming. In Temporal Logic, (pp. 445–479). Springer.

Pnueli, A. (1977). *The Temporal Logic of Programs. In Foundations of Computer Science*. IEEE.

Qadir, J., & Hasan, O. (2015). Applying Formal Methods to Networking: Theory, Techniques, and Applications. *IEEE Communications Surveys and Tutorials*, *17*(1), 256–291. doi:10.1109/COMST.2014.2345792

Rashid, A., Hasan, O., & Saghar, K. (2015). *Formal Analysis of a ZigBee-based Routing Protocol for Smart Grids using UPPAAL. In High-Capacity Optical Networks and Enabling/Emerging Technologies*. IEEE.

Renesse, R. D., & Aghvami, H. (2004) Formal Verification of Ad-Hoc Routing Protocols using SPIN Model Checker. *Mediterranean Electrotechnical Conference*, 3, 1177–1182. 10.1109/MELCON.2004.1348275

Rossi, G. L. D., Gallina, L., & Rossi, S. (2013) Performance Analysis and Formal Verification of Cognitive Wireless Networks. In *European Workshop on Performance Engineering*, (pp. 236–250). Springer. 10.1007/978-3-642-40725-3_18

Saputro, N., Akkaya, K., & Uludag, S. (2012). A Survey of Routing Protocols for Smart Grid Communications. *Computer Networks*, *56*(11), 2742–2771. doi:10.1016/j.comnet.2012.03.027

Somappa, A. A. K., Prinz, A., & Kristensen, L. M. (2015). *Model-Based Verification of the DMAMAC Protocol for Real-time Process Control*. Verification and Evaluation of Computer and Communication Systems.

Song, J., Han, S., Mok, A., Chen, D., Lucas, M., Nixon, M., & Pratt, W. (2008). *WirelessHART: Applying Wireless Technology in Real-time Industrial Process Control. In Real-Time and Embedded Technology and Applications*. IEEE.

Tschirner, S., Xuedong, L., & Yi, W. (2008). *Model-based Validation of QoS Properties of Biomedical Sensor Networks. In Embedded Software*. ACM.

Yi, W., Pettersson, P., & Daniels, M. (1994). *Automatic Verification of Real-Time Communicating Systems by Constraint-Solving*. Formal Description Techniques.

ADDITIONAL READING

Boca, P., Bowen, J. P., & Siddiqi, J. (2009). *Formal methods: State of the Art and New Directions*. Springer Science & Business Media.

Gordon, M. J., & Melham, T. F. (1993). *Introduction to HOL A Theorem Proving Environment for Higher-order Logic*. Cambridge University Press.

Hall, A. (2007). Realizing the Benefits of Formal Methods. *Journal of Universal Computer Science*, *13*(5), 669–678.

Hossain, E., Han, Z., & Poor, H. V. (2012). *Smart Grid Communications and Networking*. Cambridge University Press. doi:10.1017/CBO9781139013468

Kuzlu, M., Pipattanasomporn, M., & Rahman, S. (2014). Communication Network Requirements for Major Smart Grid Applications in HAN, NAN and WAN. *Computer Networks*, *67*, 74–88. doi:10.1016/j.comnet.2014.03.029

McMillan, K. L. (1993). *Symbolic Model Checking. Symbolic Model Checking*. Springer. doi:10.1007/978-1-4615-3190-6

KEY TERMS AND DEFINITIONS

Formal Methods: Formal methods are the computer-based techniques that involve the development of the mathematical model of the given system based on some logic and verification of its intended properties using deductive reasoning.

Home Area Network (HAN): HAN is located in the customer premises and provides a connection between the smart meter and the home appliances.

Model Checking: Model checking is a formal verification method that involves the state-space based modeling of the given system and rigorously verifying its properties specified in an appropriate logic.

Smart Grids: Smart grids are a digital upgradation of the traditional grids providing reliable, secure, safe and efficient power distribution and management.

Timed Automata: Timed automata are the finite automata extended with a set of clock variables and are used for modeling the timing aspects of the system.

UPPAAL Model Checker: UPPAAL model checker involves modeling the behavior of the given system as a network of timed automata and is extensively used for formally analyzing the real-time systems.

Zigbee Routing Protocol: ZigBee routing protocol is commonly used in wireless HAN and provides a reliable, secure and uninterrupted connection between various components of HAN.