

Research Article

Model-Checking Driven Design of QoS-Based Routing Protocol for Wireless Sensor Networks

Zhi Chen,^{1,2,3} Ya Peng,^{1,2} and Wenjing Yue^{2,4}

¹College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

²Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

³State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

⁴College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Correspondence should be addressed to Wenjing Yue; yuewj@njupt.edu.cn

Received 7 February 2015; Revised 26 May 2015; Accepted 27 May 2015

Academic Editor: Jesus Corres

Copyright © 2015 Zhi Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate and reliable routing protocols with Quality of Service (QoS) support determine the mission-critical application efficiency in WSNs. This paper proposes a model-checking design driven framework for designing the QoS-based routing protocols of WSNs, which involves the light-weight design process, the timed automata model, and the alternative QoS verification properties. The accurate feedback of continually model checking in the iterative design process effectively stimulates the parameter tuning of the protocols. We demonstrate the straightforward and modular characteristics of the proposed framework in designing a prototype QoS-based routing protocol. The prototype study shows that the model-checking design framework may complement other design methods and ensure the QoS implementation of the QoS-based routing protocol design for WSNs.

1. Introduction

Wireless sensor networks (WSNs) as the multihop self-organizing networks usually provide vital support for mission-critical applications [1] but have many design challenges such as routing [2], topology control [3], and coverage [4]. Quality of Service (QoS) support may determine the routing efficiency and effectiveness of real-time WSNs due to the constrained energy supply, bandwidth, and delay [5].

In order to design the QoS-based routing protocols for WSNs, we usually verify and evaluate the correctness and performance of these protocols using testing, simulation, and formal verification. Testing usually analyzes the implementations of these protocols and finds the protocol defects, but it is often achieved at a cost and cannot analyze all the conditions of these protocols. Simulation is commonly used in QoS-based routing protocol analysis, but it cannot analyze all the protocol behaviors. Formal verification, for example, model checking, can describe and analyze QoS-based routing protocols accurately and can make up the deficiency of testing and simulation [6–8]. Moreover, model checking can drive

the design of some applications and systems, such as the interactive systems [9] and the web application [10] and may design the QoS-based routing protocols from model driven engineering to verification driven engineering [11].

In order to effectively design the QoS-based routing protocols for the mission-critical applications of WSNs, the paper proposes a model-checking driven design framework and demonstrates the effectiveness and advantages by the accurate feedback of continually model checking in the iterative design process. The rest of the paper is organized into five sections. Section 2 briefly introduces related work. Section 3 presents the model-checking driven design framework. Section 4 introduces a prototype of the QoS-based routing protocol. Section 5 presents the protocol verification and design improvement. Finally, the conclusions are offered in Section 6.

2. Related Work

Akkaya and Younis [12] presented an energy-aware QoS routing protocol for WSNs with best-effort traffic and validated

```

Input the network  $(0, 1, \dots, n - 1)$ , QoS metrics  $(m_0, m_1, \dots, m_{k-1})$ 
Design QoS SRP: the prototype of the QoS – based routing protocol with the metric  $m_0$ 
 $i \leftarrow 0$ 
while  $(i < k)$  do
  Model QoS SRP using Timed Automata
  Verify QoS SRP via Model Checking and Output the verification results
  Evaluate the verification results and improve QoS SRP
   $i \leftarrow i + 1$ 
  if  $(i < k)$ 
    Redesign QoS SRP with the metrics  $m_0, \dots, m_i$ 
  endif
endwhile
Output QoS SRP

```

PSEUDOCODE 1: Model-checking driven design process.

the effectiveness of the protocol through simulation. Ben-Othman and Yahya [13] proposed a QoS aware multipath routing protocol based on the concept of service differentiation to control the delay and then used the NS-2 simulations to evaluate the performance of the protocol for WSNs. Sun et al. [14] presented a game-theoretic approach to coordinate the QoS routing for WSNs and used NS-2 to verify its performance with simulations. Cheng et al. [15] used the geographic opportunistic routing for QoS provisioning in WSNs and evaluated the protocol through NS-2 simulations and tests on the hardware nodes. Hu et al. [16] proposed a multihop heterogeneous cluster-based optimization algorithm (MHCOA) for WSNs and also used NS-2 simulation to show the better performance of MHCOA in heterogeneous WSNs.

Akbaş and Turgut [17] presented a routing protocol with QoS support for wireless sensor and actor networks and carried out a series of simulations in OPNET modeler to analyze the performance of the protocol. Hammoudeh and Newman [18] presented an adaptive routing protocol with QoS metrics to meet application requirements of WSNs and used the Dingo WSN simulator to evaluate the performance of the protocol. Tschirner et al. [19] presented the automata-based models for biomedical sensor networks (BSN) and successfully used the driven model-checking technique to complement the simulation techniques to validate QoS properties of BSN.

In the above-mentioned related work, the simulation-based method is the main design verification technique for the QoS-based routing protocols of WSNs [20], but it is possible that some errors which may only occur under special conditions cannot be found in the simulations; thus the QoS metrics may not meet the routing requirements of WSNs in the mission-critical applications.

3. Model-Checking Driven Design Framework

To ensure that the QoS-based routing protocol can save energy, reduce the transmission delay, increase the probability of successful transmission to the sink node, and provide high QoS for WSNs we propose the model-checking driven

design framework consisting of the design process, the timed automata model, and the verification properties.

3.1. Design Process. In the model-checking driven design framework for the QoS-based routing protocols of WSNs, the model-checking driven design process shown in Pseudocode 1 uses the iterative and incremental development. A QoS-based routing protocol of WSNs is designed through repeated cycles and in smaller parts at one time and iteratively enhanced through model checking until the full protocol is implemented. At each iteration, the design improvement is made and one new QoS metric is added.

In the model-checking driven design process, we assume there are n nodes in WSNs and m QoS metrics $(m_0, m_1, \dots, m_{k-1})$ from the application requirements of WSNs and, firstly, design one prototype of the QoS-based routing protocol using the QoS metric of m_0 , namely, QoS SRP, which we can react with and is simple enough to be understood and implemented easily; then QoS SRP is modeled using timed automata [21, 22], verified via model checking, and improved as a result of evaluating the verification results. After taking the first iteration of the design improvement, we add one new QoS metric as the new feature to redesign QoS SRP and continue to perform the process.

3.2. Timed Automata Model. In the model-checking driven design framework, we define the timed automata model, including the timed automaton of the sink node and the timed automaton of the sensor nodes according to the behavior of nodes in WSNs.

The timed automaton of the sink node is formally described as

$$\begin{aligned}
 & (Initial, [y < = INITIAL_DELAY], []) \\
 & \xrightarrow{y==INITIAL_DELAY} (Received, [x < = R], [broadcast!]), \quad (1)
 \end{aligned}$$

$$\begin{aligned}
 & (Received, [x < = R], []) \\
 & \longrightarrow (Initial, [y < = INITIAL_DELAY], [stop?]). \quad (2)
 \end{aligned}$$

The timed automaton of the sensor nodes is formally described as

$$(Listen, [time \leq backoff], []) \quad (3)$$

$$\xrightarrow{time==backoff} (Broadcasting, [], [broad?]),$$

$$(Tx, [x \leq TRANSMIT], [stop!]) \quad (4)$$

$$\rightarrow (Broadcasting, [], []),$$

$$(Broadcast, [WaitTime \leq TIMEOUT], []) \quad (5)$$

$$\xrightarrow{WaitTime==TIMEOUT} (Tx, [x \leq TRANSMIT], []),$$

$$(Broadcast, [WaitTime \leq TIMEOUT], []) \quad (6)$$

$$\rightarrow (Received, [x \leq R], [start?]),$$

$$(Received, [x \leq R], []) \quad (7)$$

$$\rightarrow (Broadcasting, [], [stop?]).$$

In the timed automata model of WSNs, *broad* is the probe information sending channel of the sink node, *start* is the message transmitting starting channel of the sensor nodes, and *stop* is the message transmitting ending channel of the sensor nodes. When the timed automata work, the clock values increase all with the same speed, and, along the state transitions in every automaton of the sink node or one sensor node, clock values being compared to integers form the guards, which may enable or disable state transitions and inhabit the possible behaviors in the mission-critical applications of WSNs.

3.3. Verification Properties. In the model-checking driven design framework, we select four CTL properties, including no deadlock, network connectivity, delivery rate of data packet, and transmission delay [19], which may be used for model checking.

(1) *No Deadlock.* This property can be formally specified as follows:

$$A [] \text{ not deadlock.} \quad (8)$$

(2) *Network Connectivity.* Any node of WSNs should communicate with the sink node, no matter directly or through the multihop paths within a certain time, and the isolated nodes should not exist in theory. This property can be formally specified as follows:

$$A \langle \rangle \text{ received}[i] > 0. \quad (9)$$

(3) *Packet Delivery Success Rate.* The channel access failure, data packet collision, information transmission error caused by the thermal noise, and external interference may result in the loss of the packets. Packet delivery success rate refers to the ratio of the number of packets successfully received by the sink node and the number of packets sent to the sink node by the sensor nodes. For example, if the sensor nodes send 10

packets to the sink node, we need to verify the packet delivery rate reaching 90%, this property can be formally specified as follows:

$$A [] (\text{periods}[X] \geq 10) \text{ imply } (\text{received}[X] \geq 9). \quad (10)$$

(4) *Transmission Delay.* Transmission delay is the effectiveness of data packets, which must be transmitted to the sink node through the multihop paths in a bounded time. This property of the transmission delay time not exceeding M of sending a packet to the sink node by the sensor node i can be formally specified as follows:

$$A [] \text{ time}[i] \leq M. \quad (11)$$

4. A Prototype of the QoS-Based Routing Protocol: QoSRP

We design a prototype of the QoS-based routing protocol for WSNs, namely, QoSRP. Assuming there are n nodes in WSNs, including one sink node and $n - 1$ sensor nodes, i represents one node, where $i \in \{0, 1, \dots, n - 1\}$ and the node 0 is the sink node. $w(i, j)$ is the link cost between the node i and the node j , where $w(j, i) = w(i, j)$, $w(i, i) = 0$, $i \in \{0, 1, \dots, n - 1\}$, and $j \in \{0, 1, \dots, n - 1\}$. $d[i]$ is the end-to-end optimal transmission cost of the node i and the sink node. We define $G = (V, S)$, where $V \cup S = \{0, 1, \dots, n - 1\}$, V represents the set of sensor nodes without the optimal next-hop neighbor nodes, and S is the set of sensor nodes with the optimal next-hop neighbor nodes.

(1) *Initialization.* In an initial state, every node in WSNs broadcasts the “detecting” message consisting of its location information, ideal transmitting radius, and residual energy information by flooding communication and calculates the link costs between it and other nodes according to the received “detecting” messages in the network. We select two QoS metrics, including the transmitting delay and the residual energy over the data transfer paths for calculating the link costs. All link costs of $w(i, j)$ are stored in the node i and if the node i does not receive the “detecting” message of the node j , $w(i, j) = \text{MAXCOST}$, where MAXCOST is the ideal maximum value. Then, the sink node starts the process of obtaining the least cost paths. At first, $S = \{0\}$, therefore, we can get

$$d[i] = \begin{cases} w(i, 0), & i \neq 0, \\ 0, & i = 0. \end{cases} \quad (12)$$

(2) *Iteratively Updating the End-to-End Optimal Transmission Costs.* For the node i in WSNs, where $i \in S$, it updates its $d[i]$ according to (3). Consider

$$d[i] = \min \{d[i], d[k] + w(k, j)\}, \quad k \in V. \quad (13)$$

According to (14), QoSRP selects $k \in S$ as the node in the optimal path, removes k from S , and adds k into V . Consider

$$d[k] = \min \{d[i] \mid i \in S\}. \quad (14)$$

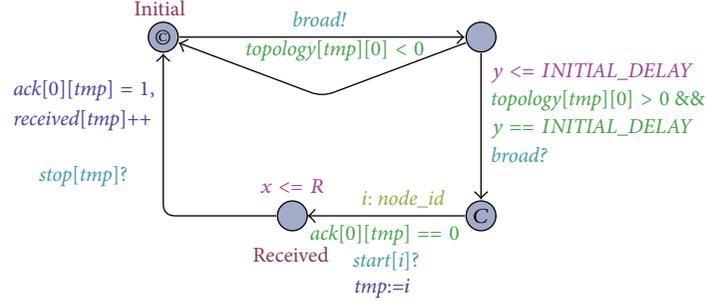


FIGURE 1: Timed automaton model of Sink.

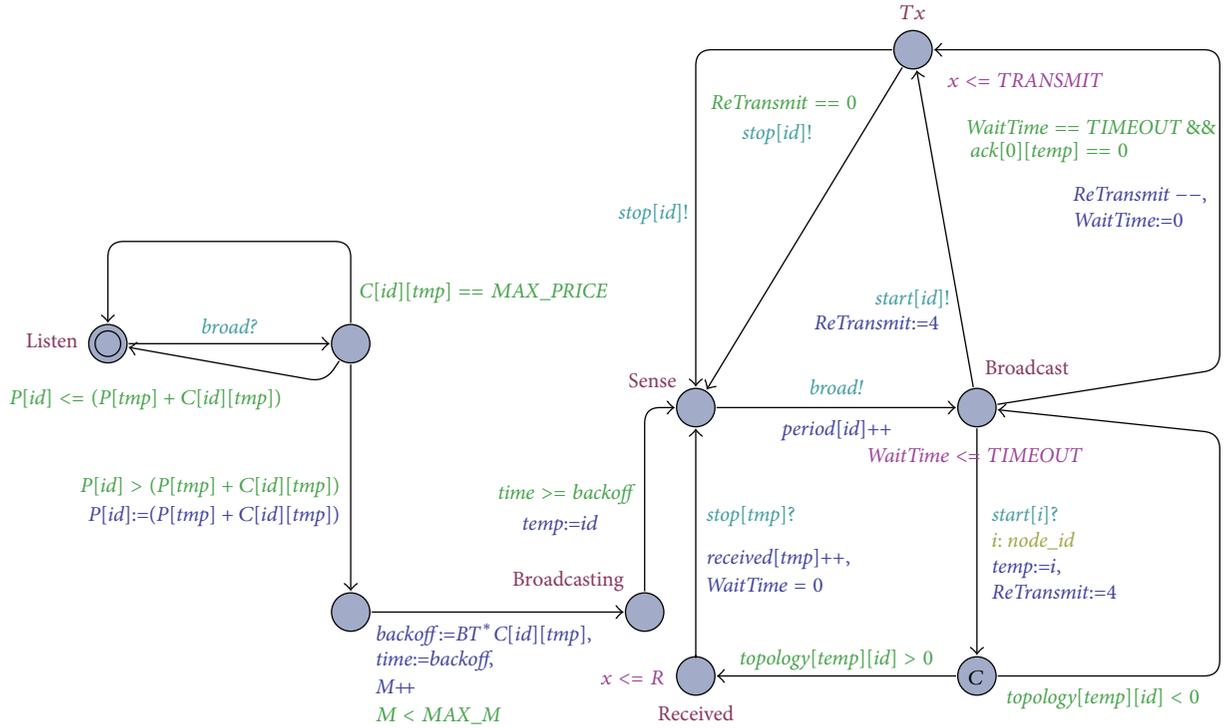


FIGURE 2: Timed automaton model of Node(id).

(3) *Finding the Optimal Path and Steadily Sending Data.* The sink node periodically broadcasts a message to form one optimal path; other nodes select a node as the parent node within the set of V until all nodes can route to the sink node. After forming one optimal path, the sensor nodes steadily send data to the sink node via the path in one period.

5. Protocol Verification and Design Improvement

5.1. Protocol Modeling. We model QoS SRP based on the timed automata model using the model checker UPPAAL [23]. In usual scenarios, all sensor nodes are modeled with the parametric timed automata in UPPAAL.

First of all, we set up a set of identifiers $node_id$ to record all nodes. Assuming that there are n sensor nodes in WSNs,

$node_id = \{0, 1, 2, \dots, n - 1\}$. In QoS SRP, the identifier of the sink node is 0, and the identifier of one other node is id , where $id \in \{1, 2, \dots, n - 1\}$. The behaviors of one sensor node in WSNs can be described using two-timed automaton: Sink and Node(id). The timed automaton of Sink shown in Figure 1 describes the sink node, and the timed automaton of Node(id) shown in Figure 2 is responsible for message sending and receiving of other sensor nodes.

Not all nodes in a real-time system are turned on simultaneously, and we constrain the turn-on times of sensor nodes in $[0, INITIAL_DELAY]$ during which any sensor node can be turned on. In Figure 1, $ack[0][tmp]$ indicates whether an acknowledge message of the node tmp is received by the sink node and before the sink node receives the acknowledge message, $ack[0][tmp] = 0$.

$start[node_id]$ and $stop[node_id]$, respectively, indicate that one node starts transmitting messages and ends a

TABLE 1: Verification results in the first iteration.

Property	Network size	Results	CPU (s)	Memory (MB)
$A[] \text{ not deadlock}$	5	Y	258.23	31.36
$A[] \text{ not deadlock}$	6	Y	5640.44	1933.5
$A \langle \rangle \text{ received}[2] > 0$	5	Y	326.67	93.45
$A \langle \rangle \text{ received}[2] > 0$	6	Y	633.52	530.5
$A[] \text{ periods}[1] < 6$	5	N	12.24	112.56
$A[] \text{ periods}[1] < 7$	5	Y	13.34	112.55
$A[] \text{ periods}[3] < 2$	5	N	11.86	112.62
$A[] \text{ periods}[3] < 3$	5	Y	12.11	112.58
$A[] \text{ periods}[5] < 2$	5	N	12.56	112.62
$A[] \text{ periods}[5] < 3$	5	Y	12.97	112.55
$A[] \text{ periods}[1] < 5$	6	N	12.83	112.57
$A[] \text{ periods}[2] < 3$	6	N	56.74	630.29
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[2] \geq 8)$	5	Y	11.64	112.59
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[2] \geq 9)$	5	N	5.25	107.63
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[3] \geq 7)$	5	Y	11.58	112.53
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[3] \geq 8)$	5	N	4.12	106.66
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[4] \geq 5)$	5	Y	11.63	112.64
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[4] \geq 6)$	5	N	4.95	106.68
$A[] \text{ time}[2] < 50$	5	N	0.763	26.02
$A[] \text{ time}[2] < 80$	5	N	63.63	161.99
$A[] \text{ time}[2] < 100$	5	Y	360.5	431.13

message transmission; $start[node_id]?$ and $stop[node_id]?$ are synchronized with $start[node_id]!$ and $stop[node_id]!$. $broad!$ means that the sink periodically sends probe information and is synchronization with $broad?$; $topology[node_id][node_id]$ indicates the connection matrix of the nodes in WSNs. $received[node_id]$ represents the number of packets sent to the sink node by the node $node_id$ and also ensures that the node can eventually connect to the sink node if $received[node_id] > 0$.

In Figure 2, the time automaton of $Node(id)$ is divided into two phases. The first phase in $Node(id)$ is to find out the minimum cost path from the node to the sink node, and the second phase is to transmit data. In $Node(id)$, $received[node_id]$ represents the number of packets received by the node from the other node $node_id$, and $periods[node_id]$ represents the number of packets sent by the node $node_id$. We can compare $received[node_id]$ and $periods[node_id]$ to illustrate the packet transmission success rate.

In Figure 2, $P[Node_id]$ indicates the shortest distances of the node $node_id$ to the sink node, and $C[Node_id][Node_id]$ is the link communication cost between two nodes. In $Node(id)$, the variable of tmp records the identifier of the node sending messages. $Node(id)$ follows the task processing mechanism of first-come, first-served in TinyOS [24] and considers the transmission delay in modeling message sending. Message transmission may need a few seconds, change the order of message sending and receiving, and affect the protocol implementation, so it cannot be ignored. Therefore we add the variable of x simulating the message delay, where $x \leq DELAY$ and $DELAY$ is the maximum value of the message transmission delay.

5.2. Protocol Verification. We verify QoS SRP about the properties such as no deadlock, network connectivity, delivery rate of data packet, and transmission delay [19]. Table 1 is the verification results in the first iteration in which the QoS metric is the residual energy over the data transfer paths, and it shows that the performance properties were not satisfied if the sensor nodes send fewer packets to the sink node, and the packet successful delivery rates did not satisfy the network QoS requirements.

5.3. Design Improvement. According to the verification results shown in Table 1, the performance of the current QoS SRP still requires to be improved, so we continue to tune the parameters shown in Figures 1 and 2, such as $INITIAL_DELAY$, $TRANSMIT$, MAX_M , $TIMEOUT$, and MAX_PRICE , and optimize QoS SRP through model checking.

Now we combine the two QoS metrics of the transmitting delay and the residual energy over the data transfer paths to redesign QoS SRP. The pseudocode of the redesigned QoS SRP in one period is given in Pseudocode 2. According to the model-checking driven design framework, the new QoS SRP is reevaluated through model checking and Table 2 presents the verification results which show a better performance of the protocol.

6. Conclusions

The QoS-based routing protocols are the mission-critical application requirements of WSNs and involve many verification and evaluation techniques such as testing, simulation, and formal verification. This paper proposes a model-checking driven framework for designing these protocols,

```

Input the network  $(0, 1, \dots, n - 1)$ , QoS metrics (Residual_Energy, Transmitting_Delay)
Output QoS-RP: the improved prototype of the QoS – based routing protocol
BEGIN
  for node  $(i \leftarrow 0 \text{ to } n - 1)$ 
    Broadcast Detecting_Message(Locationi, Transmitting_radiusi, Residual_energyi)
  endfor
  for node  $(i \leftarrow 0 \text{ to } n - 1)$ 
    for node  $(j \leftarrow 0 \text{ to } n - 1)$ 
      // Calculate  $w(i, j)$  only for verifications
       $w(i, j) = \frac{\|Transmitting\_radius\|}{\|Residual\_energy_i\|}$  //  $\| \cdot \|$  for the normalization function
    endfor
  endfor
  for node  $(i \leftarrow 1 \text{ to } n - 1)$ 
    Update  $d[i]$ 
  endfor
  repeat
    Adjust  $G \leftarrow (V, S)$ 
    Evaluate QoS
  until Find one optimal path
  for node  $(i \leftarrow 1 \text{ to } n - 1)$ 
    Sending data via the optimal path
  endfor
END

```

PSEUDOCODE 2: Pseudocode of the redesign QoS-RP.

TABLE 2: Verification results in one next iteration.

Property	Network size	Results	CPU (s)	Memory (MB)
$A[] \text{ not deadlock}$	5	Y	259.15	32.36
$A[] \text{ not deadlock}$	6	Y	5587.29	2001.5
$A \langle \rangle \text{ received}[2] > 0$	5	Y	324.65	94.66
$A \langle \rangle \text{ received}[2] > 0$	6	Y	613.37	535.12
$A[] \text{ periods}[1] < 6$	5	Y	13.16	112.55
$A[] \text{ periods}[1] < 7$	5	Y	13.55	112.79
$A[] \text{ periods}[3] < 2$	5	Y	11.59	112.16
$A[] \text{ periods}[3] < 3$	5	Y	12.51	113.69
$A[] \text{ periods}[5] < 2$	5	Y	12.97	112.79
$A[] \text{ periods}[5] < 3$	5	Y	12.48	112.28
$A[] \text{ periods}[1] < 5$	6	Y	12.65	112.32
$A[] \text{ periods}[2] < 3$	6	N	56.29	631.55
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[2] \geq 8)$	5	Y	12.15	114.43
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[2] \geq 9)$	5	Y	5.65	108.63
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[3] \geq 7)$	5	Y	12.01	113.18
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[3] \geq 8)$	5	Y	4.23	105.08
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[4] \geq 5)$	5	Y	12.03	111.98
$A[] (\text{periods}[2] \geq 10) \text{ imply } (\text{received}[4] \geq 6)$	5	Y	4.98	105.76
$A[] \text{ time}[2] < 50$	5	N	0.754	25.92
$A[] \text{ time}[2] < 80$	5	Y	63.78	162.03
$A[] \text{ time}[2] < 100$	5	Y	361.4	430.18

including the iterative design process, the timed automata model, and the alternative verification properties. We design a prototype of the QoS-based routing protocol and demonstrate continuing improving the protocol using the proposed framework and UPPAAL model checking. The results of the prototype study show that the model-checking driven design framework is a straightforward and modular method and supports the light-weight iterative redesign for designing

QoS-based routing protocols of WSNs, and the feedback of continually model checking accurately drives the performance improving of the protocols of real-time WSNs.

Conflict of Interests

The authors declare no conflict of interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 60905040), the Basic Research Program of Jiangsu Province (Natural Science Foundation) (Grant no. BK20131382), the 11th Six Talent Peaks Program of Jiangsu Province (Grant no. XXRJ-009), China Postdoctoral Science Foundation (Grant no. 2013M531393), and Jiangsu Planned Projects for Postdoctoral Research Funds (Grant no. 1102102C).

References

- [1] M. A. Mahmood, W. K. G. Seah, and I. Welch, "Reliability in wireless sensor networks: a survey and challenges ahead," *Computer Networks*, vol. 79, pp. 166–187, 2015.
- [2] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, 2012.
- [3] Z. Chen, S. Li, and W. Yue, "SOFM neural network based hierarchical topology control for wireless sensor networks," *Journal of Sensors*, vol. 2014, Article ID 121278, 6 pages, 2014.
- [4] Z. Chen, S. Li, and W. Yue, "Memetic algorithm-based multi-objective coverage optimization for wireless sensor networks," *Sensors*, vol. 14, no. 11, pp. 20500–20518, 2014.
- [5] R. A. Uthra and S. V. K. Raja, "QoS routing in wireless sensor networks—a survey," *ACM Computing Surveys*, vol. 45, no. 1, article 9, 2012.
- [6] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, "Formal methods: practice and experience," *ACM Computing Surveys*, vol. 41, no. 4, article 19, 2009.
- [7] D. Le Metayer, "Formal methods as a link between software code and legal rules," in *Software Engineering and Formal Methods: 9th International Conference, SEFM 2011, Montevideo, Uruguay, November 14–18, 2011. Proceedings*, vol. 7041 of *Lecture Notes in Computer Science*, pp. 3–18, Springer, Berlin, Germany, 2011.
- [8] E. M. Clarke, J. O. Grumberg, and D. A. Peled, *Model Checking*, MIT Press, Boston, Mass, USA, 1999.
- [9] A. Cerone and N. Elbegbayan, "Model-checking driven design of interactive systems," *Electronic Notes in Theoretical Computer Science*, vol. 183, pp. 3–20, 2007.
- [10] F. M. Donini, M. Mongiello, M. Ruta, and R. Totaro, "A model checking-based method for verifying web application design," *Electronic Notes in Theoretical Computer Science*, vol. 151, no. 2, pp. 19–32, 2006.
- [11] F. Kordon, J. Hugues, and X. Renault, "From model driven engineering to verification driven engineering," in *Software Technologies for Embedded and Ubiquitous Systems*, vol. 5287 of *Lecture Notes in Computer Science*, pp. 381–393, Springer, Berlin, Germany, 2008.
- [12] K. Akkaya and M. Younis, "An energy-aware QoS routing protocol for wireless sensor networks," in *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems Workshops*, pp. 710–715, Providence, RI, USA, May 2003.
- [13] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849–857, 2010.
- [14] R. Sun, E. Ding, H. Jiang, R. Geng, and W. Chen, "Game theoretic approach in adapting QoS routing protocol for wireless multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 745252, 5 pages, 2014.
- [15] L. Cheng, J. Niu, J. Cao, S. K. Das, and Y. Gu, "QoS aware geographic opportunistic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1864–1875, 2014.
- [16] S. Hu, J. Han, X. Wei, and Z. Chen, "A multi-hop heterogeneous cluster-based optimization algorithm for wireless sensor networks," *Wireless Networks*, vol. 21, no. 1, pp. 57–65, 2015.
- [17] M. I. Akbaş and D. Turgut, "Lightweight routing with dynamic interests in wireless sensor and actor networks," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2313–2328, 2013.
- [18] M. Hammoudeh and R. Newman, "Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance," *Information Fusion*, vol. 22, pp. 3–15, 2015.
- [19] S. Tschirner, X. Liang, and Y. Wang, "Model-based validation of QoS properties of biomedical sensor networks," in *Proceedings of the 8th ACM International Conference on Embedded Software*, pp. 69–78, October 2008.
- [20] S. Sridevi and M. Usha, "Energy-aware QoS based routing protocols for heterogeneous WSNs—a survey," *International Journal of Computer Science and Business Informatics*, vol. 11, no. 1, pp. 1–19, 2014.
- [21] J. Bengtsson and Y. Wang, "Timed automata: semantics, algorithms and tools," in *Lectures on Concurrency and Petri Nets*, vol. 3098 of *Lecture Notes in Computer Science*, pp. 87–124, Springer, Berlin, Germany, 2004.
- [22] P. Fontana and R. Cleaveland, "A menagerie of timed automata," *ACM Computing Surveys*, vol. 46, no. 3, article 40, 2014.
- [23] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on uppaal," in *Formal Methods for the Design of Real-Time Systems*, vol. 3185 of *Lecture Notes in Computer Science*, pp. 200–236, Springer, Berlin, Germany, 2004.
- [24] P. Levis and D. Gay, *TinyOS Programming*, Cambridge University Press, Cambridge, UK, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

