A Decentralized Scheduling Function for TSCH-Based Wireless Networks

Wei Yang[1(✉)], Yuanlong Cao[1], Xun Shao[2], Hao Wang[1], Zhiming Zhang[1], and Qinghua Liu[1]

[1] School of Software, Jiangxi Normal University, Nanchang, China
yw@jxnu.edu.cn

[2] School of Regional Innovation and Social Design Engineering, Kitami Institute of Technology, Kitami, Japan

Abstract. Time-Slotted Channel Hopping (TSCH) is the emerging standard for industrial automation and process control low-power and lossy networks. Scheduling function is very crucial in the TSCH-based wireless networks, which defines the packets select which cells to send/receive. But the current industrial wireless standard does not define the function how to adds/deletes/relocates cells between neighbors. In this paper, we propose a decentralized scheduling function, which fully consider the use probability and distance of cell rather than simple random selection. We implement a decentralized scheduling function in 6TiSCH simulator and evaluate its performance experimentally in different cases. The experimental results show that our proposed scheme can reach lower end-to-end latency, as well as no extra costs.

Keywords: Time-slotted channel hopping (TSCH) · Decentralized scheduling · Wireless sensor network

1 Introduction

Industrial wireless sensor networks will be widely used in the field of industrial process automation in the future [1, 2]. The technology can greatly improve the efficiency and quality of industrial production. Industrial automation applications have strict requirements for low power consumption, high reliability and real-time performance of wireless network. These demands have been difficult to fulfill until the introduction of the Time-Synchronized Mesh Protocol (TSMP) [3] in 2008. Industrial wireless standards such as WirelessHART (2008) [4] and ISA100.11a (2011) [5] adopt TSMP technology. In April 2012, time-slotted channel hopping (TSCH), which is based on TSMP technology, became an important medium access control (MAC) layer protocol of the IEEE802.15.4e standard [6].

In TSCH-based wireless networks, all nodes keep high-precision time synchronization. Parts of nodes in the networks can cooperatively go to sleep at the same time when they have no data to send. It can achieve a low-power wireless network and

C. T. Calafate et al. (Eds.): MONAMI 2021, LNICST 418, pp. 194–200, 2022.
https://doi.org/10.1007/978-3-030-94763-7_15

prolong network life, which is very important in the battery-powered industrial applications. External interference and multi-fading seriously affects the reliability of wireless communication. The nodes in TSCH-based wireless networks, adopt channel hopping technology to combat the challenge. Research has proved it can reach 99.9% reliability [7, 8], which can be equip to the reliability of wired industrial networks. IEEE.802.15.4e standard defines time-slotted communication mechanisms for a pair of nodes exchanging packet in a cell (see Fig. 1). The communication of nodes happens at a [slotOffset, channelOffset] location according to the network schedule. However, it does not provide policies pertaining to the time for adding/deleting cells or the cells to select. Therefore, the scheduling function is crucial for solving the problems above.

The existing scheduling algorithms in TSCH-based wireless networks can be classified into two categories: centralized scheduling and distributed scheduling. Centralized scheduling algorithms (e.g., traffic aware scheduling algorithm (TASA) [9] and adaptive multi-hop scheduling (AMUS) [10]) rely on a center node to compute the schedule of all nodes in the networks. It is only suitable for static networks. However, numerous mobile nodes exist in industrial wireless applicants, and the nodes traffic are not fixed. Most scheduling algorithms for 6TiSCH networks involve distributed scheduling (e.g., on-the-fly (OTF) [11], SF0 [12], low latency scheduling function (LLSF) [13], recurrent low-latency scheduling function (ReSF) [14]).

However, the current distributed scheduling algorithms are affected by some challenges. First, the end-to-end latency in a multi-hop network is excessively high and uncertain. Some packets such as the alarm information in industrial process control systems must be transmitted to the border router in a timely manner. It requires the scheduling algorithms to coordinate a low latency multi-hop path. Second, schedule collisions often occur in distributed scheduling algorithms. Two pairs of neighbor nodes are scheduled in the same cell to send/receive data. Schedule collisions can further affect the performance of the network (e.g., reliability, power, and latency). Therefore, an efficient schedule function for TSCH-based wireless industrial networks must be identified. In the paper, we will propose our scheduling function for TSCH-based wireless networks.

The remainder of this article is organized as follows. Section 2 proposes a decentralized scheduling function for TSCH-based wireless networks. Section 3 evaluates the performance of the proposed scheme. Finally, Sect. 4 concludes this paper and presents future work.

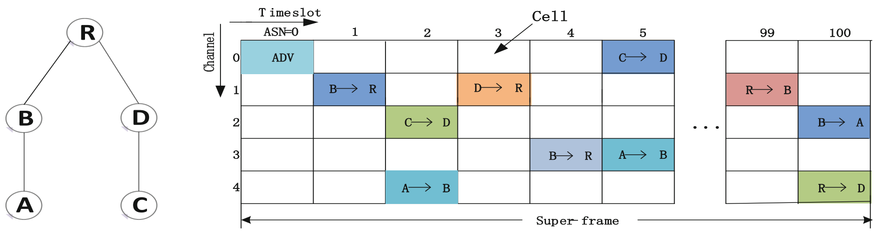


Fig. 1. Example of communication in TSCH-based wireless networks.

2 Decentralized Scheduling Function

Industrial wireless networks have strict requirements for real-time and reliability. The goal of resource scheduling algorithm is to provide low-latency end-to-end delay and no conflict communication. The new node joins and synchronizes to the TSCH-based wireless networks after receiving the EB packets from the multiple neighbor nodes. It may choose a prefer neighbor node as its parent according to the JP value, which is contained in the EB packets. Then the preferred parent node will allocate first Tx and Rx negotiated cells to the new joining node by 6P command. The autonomous cells are only used to send EB and DIO packets. The new node need to add Tx and Rx cells with its preferred parent after it continuously generate packets. Therefore, when to add or delete timeslots and which timeslots to select is a critical problem in the scheduling algorithm.

According to the latest 6TiSCH minimal scheduling function documentation, it specifies when to add or delete timeslots such as adapting to traffic, switching parent and handling schedule collisions. But it did not define how to select timeslots. The SF0 [12] which is proposed by 6TiSCH working group just used simple random timeslots selection. Time slot selection is very important in the resource scheduling algorithm, and it will directly affect the end-to-end delay and reliability of network. The goal of our scheme is to minimize the end-to-end packet transmission delay. It selects a suitable cell from the available cell list based on the use probability and distance of cell, which is different from random timeslot selection in SF0 [12] and LLSF [13].

Table 1. System symbols.

Symbol	Description
List_Tx	Available Tx cell
List_Rx	Available Rx cell
P_{ti}	Packet transmission probability of the i-th Tx cell
P_{ri}	Packet receiving probability of the i-th Rx cell
D_{ri}	The distance between the i-th Rx cell and the selected Tx cell
Slotframe_length	The length of slotframe

Table 1 shows the system symbols in the decentralized scheduling function. Here, we introduce the concepts of probability and distance. The core idea of our scheme is to select a most suitable Tx cell from List_Tx, which can make the sum of the distances to all Rx cell the shortest, thereby ensuring the lowest time delay. There may be multiple Rx cell, and different Rx cell have different probability of receiving data packets. The packet receiving probability of the i-th Rx cell can be counted. Each Rx cell will record the number of received data packets. The total data packet is the sum of the multiple Rx cell from the same source node, and the probability P_{ri} is the ratio of the received data packet of the i-th Rx cell to the total data packet. Each Tx cell and Rx cell has an ASN value (typical one ASN value represents 15 ms). The distance between the Tx cell and Rx cell can be calculated by the Eq. (1). Assuming the number of available Rx cell in the

List_Rx is k, we can calculate the sum of the distances from the one selected Tx cell to all Rx cell based on the Eq. (2). Repeating the above method, we can calculate the distance of all the available Tx cell in the List_Tx. Finally, we select the smallest distance from the calculation results, and can easily deduce which Tx cell is. In the following, we will take an example to further illustrate.

$$D_r = \begin{cases} T_x\_ASN - R_x\_ASN, \; T_x\_ASN > R_x\_ASN \\ Slotframe\_length + T_x\_ASN - R_x\_ASN, \; T_x\_ASN < R_x\_ASN \end{cases} \quad (1)$$

$$Sum\_D_r = P_{r1} * D_{r1} + P_{r1} * D_{r2} + \ldots + P_{rk} * D_{rk}, \; P_{r1} + P_{r2} + \ldots + P_{rk} = 1; \quad (2)$$

Assuming the networks consisting of 5 nodes and slotframe_length is 23, the node R is root. Here, taking node B as an example, it describes how the decentralized sched-uling function adds and deletes a Tx cell. Node B will periodically receive data pack-ets from node A, and usually there will be one packet per slotframe. The node B has 3 Rx cells from A (timeslot 4, 9 and 16), which is shown in Fig. 2. In order to be able to forward data packets from node A to the root node R, node B needs to add a Tx cell. Currently, node B has 3 available Tx cells (timeslot 8, 14 and 18). In our scheme, the node B should select one suitable Tx cell to minimize the end-to-end packet transmission delay. The sum of the distance from the one selected Tx cell to all Rx cell is shortest. And the calculation method of distance can refer to Eq. (2). Assum-ing the selected Tx cell is timeslot 14, we can calculate the distance Dr1 is 10, Dr2 is 5 and Dr3 is 21 based on the Eq. (1). Further assuming the probability Pr1 is 0.3, Pr2 is 0.6 and Pr3 is 0.1, we can calculate the sum of distance is 8.1 based on the Eq. (2). Using the same method, we can calculate the sum of distance value under different Tx cell. If the selected Tx cell is timeslot 8, the sum of distance value is 15.9. And if the se-lected Tx cell is timeslot 18, the sum of distance value is 9.8. By comparison, we can infer that the shortest distance can be achieved by selecting Tx timeslot 14.

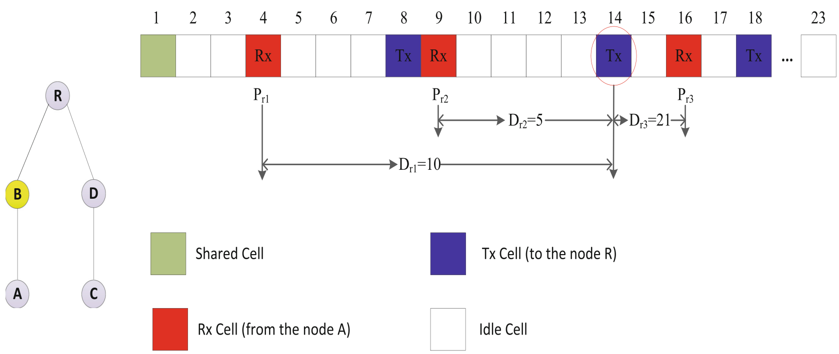


Fig. 2. Example of scheduling an Tx cell in the node B.

During a certain period of network running, the network traffic of node B is lower than the capacity. The node B determines to delete one of the Tx cells. Figure 4 depicts the general process of deleting Tx cells. Node B has 3 Rx cells from node A (timeslot

4, 9 and 16) and 4 Tx cells to node R (timeslot 8, 11, 14 and 18). To each Tx cell, node B counts the number of sending packets. It is easy to derive Pti, which is the packet transmission probability of the i-th Tx cell. Here, the Tx cell with the lowest packet transmission probability needs to be deleted. In Fig. 3 the Tx cell (timeslot 11) need to remove.

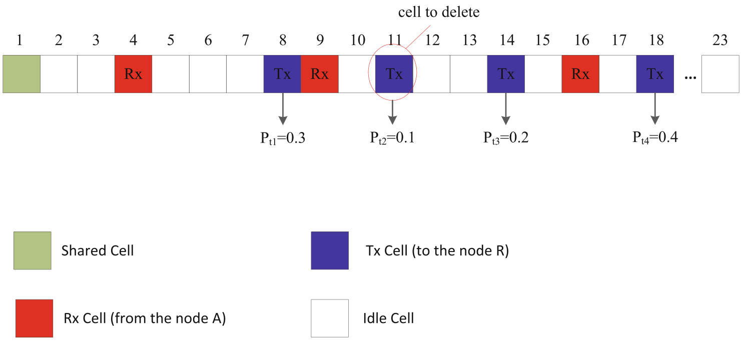


Fig. 3. Example of unscheduling an Tx cell in the node B.

3 Simulation Setup and Results

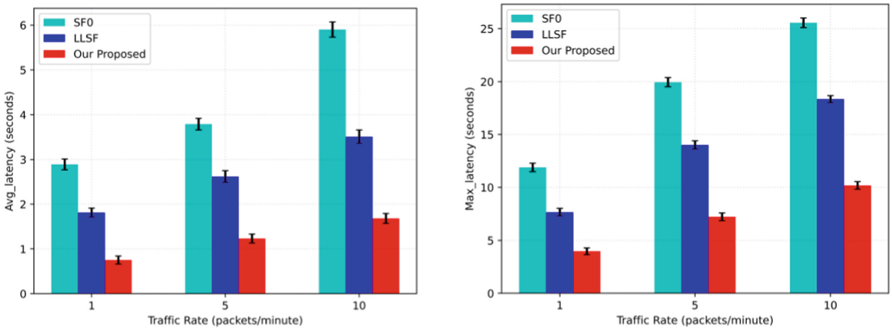
3.1 Simulation Setup

We adopt 6TiSCH open-source simulator [15] to evaluate our decentralized scheduling function. The 6TiSCH simulator is a discrete-event simulator written in Python, which is fully support TSCH protocol stack. The experiment parameters can be easily set in the configuration file. The num_Motes parameter is set to 25. The app_pkPeriod parameter is set to 6 s, 12 s and 60 s respectively, which represent the traffic rate is 10 packets/min, 5 packets/min and 1 packet/min. And we repeat 30 runs for each scheduling function to obtain the experiment results.

3.2 Experimental Results

In order to verify the effectiveness of our distributed scheduling function, we implemented our scheme on 6TiSCH simulator. End to end delay is an important metrics in distributed resource scheduling algorithm. Here we mainly consider two metrics: average end-to-end latency and maximum end-to-end latency.

First, we compared our proposed scheme to SF0 and LLSF in terms of average end-to-end delay. Average end-to-end delay is the main metrics of performance evaluation in the resource scheduling algorithm. As shown in Fig. 4 (a), the average end-to-end delay of our proposed scheme is much lower than that of SF0 and LLSF in the different traffic rate cases. When the network traffic rate is 1 packets/min, the average end-to-end delay of our proposed scheme is only about 0.75 s. At the same time, the average end-to-end delay of SF0 and LLSF reached 2.89 s and 1.81 s. Compare to SF0 and LLSF, the average end-to-end delay of our proposed scheme has dropped significantly. Fig. 4 (b) shows that the maximum end-to-end latency of our proposed scheme is much lower than SF0 and LLSF.



（a） The average end-to-end latency （b）The maximum end-to-end latency

Fig. 4. The average and maximum end-to-end latency of SF0, LLSF and Our Proposed Function in different cases.

4 Conclusion

In this paper, we presented an efficient decentralized scheduling function for TSCH-based wireless networks. Our paper has three novel contributions. First, we proposed a cell selection method which fully consider the use probability and distance of cell. It can solve the problem of high latency in the current scheduling functions. Second, we proposed a cell deletion method which fully consider packet transmission probability. It can reduce network energy consumption. Finally, we implemented our proposed scheme and verified its performance through experimentation on the 6TiSCH simulator. The experimental results show that our proposed scheme can reach lower average and maximum end-to-end latency than the SF0 and LLSF scheduling functions. In our future work, we will deploy our proposed scheme in a real-world 6TiSCH wireless networks to further verify its performance.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No. 62002143 and the Natural Science Foundation of Jiangxi Province under Grant No. 20192BAB217007.

References

1. Khan, W., Rehman, M., Zangoti, H., et al.: Industrial internet of things: recent advances, enabling technologies and open challenges. Comput. Electr. Eng. **81**(11), 1065–1087 (2020)

2. Huang, X.: Intelligent remote monitoring and manufacturing system of production line based on industrial Internet of Things. Comput. Commun. **150**(1), 421–428 (2020)
3. Pister, K., Doherty, L.: TSMP: time synchronized mesh protocol. In: International Symposium on Distributed Sensor Networks (DSN), pp. 391–398 (2008)
4. WirelessHART Specification 75: TDMA Data-Link Layer, HART Communication Foundation Std., Rev. 1.1, 2008, hCF SPEC-75
5. ISA-100.11a-2011: Wireless Systems for Industrial Automation: Process Control and Related Applications, International Society of Automation (ISA) Std., May 2011
6. IEEE Standard for Local and Metropolitan Area Networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer, IEEE Standard 802.15.4e-2012, April 2012
7. Stanislowski, D., Vilajosana, X., Wang, Q., et al.: Adaptive synchronization in IEEE802.15.4e networks. IEEE Trans. Ind. Inform. **10**(1), 795–802 (2014)
8. Watteyne, T., Tuset-Peiro, P., Vilajosana, X., et al.: Teaching communication technologies and standards for the industrial IoT? Use 6TiSCH! IEEE Commun. Mag. **55**(5), 132–137 (2017)
9. Accettura, N., Vogli, E., Palattella, M.R., et al.: Decentralized traffic aware scheduling in 6TiSCH networks: design and experimental evaluation. IEEE Internet Things J. **2**(6), 455–470 (2015)
10. Jin, Y., Kulkarni, P., Wilcox, J., et al.: A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks. In: 2016 International Conference on Wireless Communications & Networking Conference, pp. 1725–1730. IEEE (2016)
11. Palattella, M.R., Watteyne, T., Wang, Q., et al.: On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks. IEEE Sens. J. **16**(2), 550–560 (2016)
12. Dujovne, D., Grieco, L., Palattella, M., Accettura, N.: 6TiSCH 6top scheduling function zero/experimental (SFX), Internet-Draft draft-ietf-6tisch-6top-sfx-00, Internet Engineering Task Force, 22 September 2017
13. Chang, T., Watteyne, T., Wang, Q., Vilajosana, X.: LLSF: low latency scheduling function for 6TiSCH networks. In: 2016 International Conference on Distributed Computing in Sensor Systems Conference, pp. 93–95 (2016)
14. Daneels, G., Spinnewyn, B., Latre, S., et al.: ReSF: recurrent low-latency scheduling in IEEE 802.15.4e TSCH networks. Ad hoc Netw. **6**(9), 100–114 (2018)
15. Esteban, M., Glenn, D., Mališa, V., et al.: Simulating 6TiSCH networks. Trans. Emerg. Telecommun. Technol. **30**(3), 1–16 (2019)