A Cross-layer Method for Scheduling and Routing Real-time Traffic Flow in Industrial IoT

Nikta Azimian, Reza Mohammadi, and Mohammad Nassiri

Bu-Ali Sina University, Hamedan, Iran

https://doi.org/10.26636/jtit.2023.3.1362

Abstract — In the Industrial Internet of Things, a wide variety of sensors are distributed all over the environment to monitor data collection, thereby allowing industrial processes to be monitored more efficiently. One of the fundamental goals of IIoT is to provide the highest level of reliability while simultaneously increasing network lifetime, reducing power consumption, and preventing delays. 6TiSCH is a popular communication standard relied upon in IIoT. The aim of the present study is to propose an inter-layer method that simultaneously considers network scheduling and routing processes based on TSCH and RPL approaches in multi-sink environments. The proposed method is intended to address the limitations of IIoT and meet the requirements of field-specific applications.

***Keywords* — Internet of Things, Industrial IoT, multi sink, TSCH standard**

1. Introduction

The growing popularity of smart gadgets, industrial automation systems, and smart buildings has resulted in a tremendous growth of wireless sensor networks (WSNs) and the Internet of Things (IoT). The underlying aim of the Internet of Things is to connect all devices to the Internet. In the Internet of Things, a wide variety of sensors are distributed all over the environment to monitor and report various conditions and states, thereby allowing for smart decisions to be made based on the collected information [1]–[3]. In recent years, the Industrial Internet of Things (IIoT) has been proposed as a specialized IoT variety that is compatible with industrial environments and manufacturing sites. This idea is aimed at increasing flexibility and productivity while simultaneously decreasing production costs in industrial environments.

Although different devices, including entire systems, sensors, etc. were only connected to form clusters in the past, they are now connected to a single macro-structure that is accessed by the user. With the help of IIoT, industries may be connected to a single integrated network [2]–[5]. IIoT consists of a set of nodes, including a processor, a battery (power supply), a radio transceiver for communication and data exchange, and a number of sensors (monitoring temperature, pressure, humidity, etc.). These nodes exchange their data with one or more receiver stations that may be either of the stationary or mobile variety. Nodes working in IoT environments face various limitations related to their processing power, memory capacity, and energy consumptions [1]–[4]. In IIoT applications, the most important criteria include also reliability and delay (lag). In order to ensure optimum power consumption in IIoT settings, it is necessary to develop suitable sleep/wake mechanisms and to use the Carrier Sense Multiple Access (CSMA) approach in the Medium Access Control (MAC) sublayer [1], [2]. To achieve this goal, such standards as IEEE 802.15.4 [6] and IEEE 802.15.4e [7] define low power MAC and physical layers capable of meeting IIoT requirements. Focusing on the centralized coordination of nodes, the IEEE 802.15.4 standard controls the operation of CSMA among the nodes. It coordinates the transmissions by sending small packets called beacons. TSCH is one of the varieties of IEEE 802.15.4e. It is a combination of time synchronization, frequency division, and channel hopping that reduces the chance of a collision between different transmitters. The TSCH mechanism enables each node to synchronize its data transmission on a single channel and in a specific time slot. However, the standard does not determine how the channel/slot must be allocated.

In WSNs, addresses are allocated using IPv6. To activate IPv6 in TSCH, the IETF 6TiSCH work group runs a protocol stack based on IIoT-specific standards, such as RPL, 6LoWPAN, and CoAP routing protocols, using 6LoWPAN for address mapping. In addition to designing the MAC and physical layers, the current structure of the routing layer was modified in IEEE 802.15.4e by means of a routing protocol called RPL that is suitable for low-power networks. In order to use the multi-parent feature of this protocol, its DODAG structure can be utilized for convergent traffic in IIoT solutions [1]. As mentioned above, numerous scheduling methods

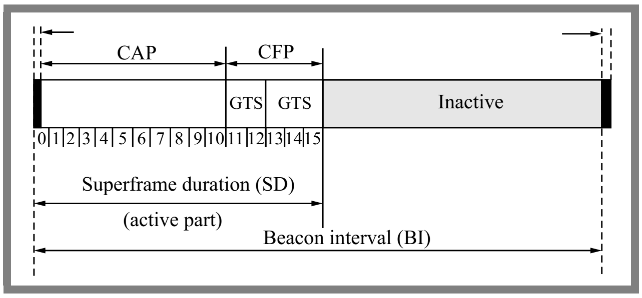


Fig. 1. The superframe structure in IEEE 802.15.4.

have so far been proposed for 6TiSCH networks, but few studies have addressed the problem of time scheduling in multi-sink networks. Therefore, the aim of this study is to meet the needs of industrial applications and remove some of the existing limitations by implementing several sinks in the network and utilizing a scheduling algorithm. In what follows, we shall investigate the possibility of creating multiple sinks. The reason for using multiple sinks in IIoT is to reduce power consumption and, ultimately, increase network lifetime. Using a 6TiSCH simulator, we evaluate the performance of the implemented method and compare it to other proposals.

Section 2 presents a review of the communication standards used in industrial IoT. In Section 3, we define the basic concepts and, in Section 4, we briefly review the existing literature. Section 5 describes the simulations performed and evaluates the results. Finally, Section 6 sums up the results and draws the conclusions.

2. Communication Standards in IIoT

An implementation of an IIoT environment requires a variety of standards and protocols to be complied with. This section examines several standards for the data link layer and the network layer in IIoT applications.

A wireless sensor network (WSN) is usually used in IIoT for purposes that contribute to mitigating overload conditions caused by intra-network processing, data integration, and decision-making processes. The fundamental challenges in these networks are to decrease delay, reduce power consumption, ensure fault tolerance, and increase the lifetime of nodes.

IEEE 802.15.4 implements the physical layer and the media access control layer for low–power WSNs. It transmits packets via the data link layer, using a shared channel, in an integrated manner relying on the synchronization of nodes. The physical layer of this standard uses one of three bands: 868 to 868.6 MHz, 902 to 928 MHz, and 2400 to 2483 MHz. The majority of the implementations so far have relied upon the 2.4 GHz band with 16 separate channels and a transmission rate of 250 kbps. The data link layer is designed to reduce the probability of data collision and data transfer using this layer is either of the beaconless or beacon-based variety. In the beaconless mode, the classic CSMA-CA approach is implemented, in which the sender listens to the channel. If it finds the channel empty, it sends the data. Otherwise, it waits a random time interval before sending the data. The network coordinator must stay awake to receive the frame from the nodes, hence it cannot save energy. This increases the overload and greatly affects the end-to-end delay. Since the nodes keep consuming energy even when idle, this state is not appropriate for multi-hop modes which suffer from energy limitations [1].

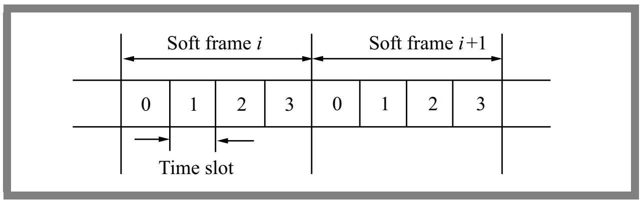


Fig. 2. A slot frame with four time slots in the TSCH mode of IEEE 802.15.4e.

In the beacon-based mode provided for in IEEE 802.15.4, both the sender and the receiver use a superframe structure (Fig. 1) for exchanging data. In this mode, a coordinator periodically sends beacons to determine the boundaries of its superframe. A superframe has an active and an inactive period. By determining the duration of each of the periods, the work cycle can be set dynamically. An active period begins with a beacon frame which is sent by the coordinator in regular time intervals. A superframe is composed of CAP and CFP. CAP is the part in which competition takes place and packets are transmitted by means of the CSMA/CA mechanism. In CFP, with no competition, data packets are sent and GTS cells are allocated to data transmission. In the beacon-based mode, nodes are awakened by the coordinator just before sending the beacon, in order to save energy, and they fall asleep again after receiving data. This allows more energy to be saved, because the nodes are not required to stay awake permanently.

The IEEE 802.15.4e standard, based on IEEE 802.15.4 used in low-power short-range wireless networks, has become popular for communicating in industrial environments [7]. The IEEE 802.15.4 standard has certain limitations, such as high power consumption of middle nodes that must always remain operable in order to transfer information to other nodes, or using a single channel for information transfers, which causes interference. IEEE 802.15.4 uses of some of the existing methods applied in WSNs, including time slots, specific and shared cells, multiple channels, as well as frequency hopping. It relies upon different modes, including LLDN, BLINK, DSME, AMCA, and TSCH. As the TSCH mechanism is a solution that reconciles all other modes and meets as many needs as possible, it is the most complex and appealing solution and is considered to be the core mechanism of IEEE 802.15.4e [1], [6]. TSCH is a combination of time synchronization, frequency division, and channel hopping. Due to its optimal use of time and frequency, this technique is a suitable choice for networks with multi-hop communications.

In TSCH, time is divided into specific spans called slot frames or superframes. Each slot frame is divided into a certain number of equal time slots and repeats periodically. The duration of a time slot is equal to the duration in which the sender transmits a packet to its single-hop neighbor and receives the verification of the packet's second layer from the neighbor. All nodes within the network share the same time structure. Its ability to support multiple channels by channel hopping is a characteristic feature of TSCH. TSCH has 16 different channels, each with the bandwidth of 5 MHz. In this mechanism, a direct communication link is created between sender and receiver nodes in a given time slot in a specific channel. Each link that is selected for communication between two nodes is specified by two indexes, i.e., time slot number and channel number. Figure 2 shows a slot frame with four time slots. In each time slot, a node can send the largest amount of data possible [1]. The RPL standard [8] has recently emerged as

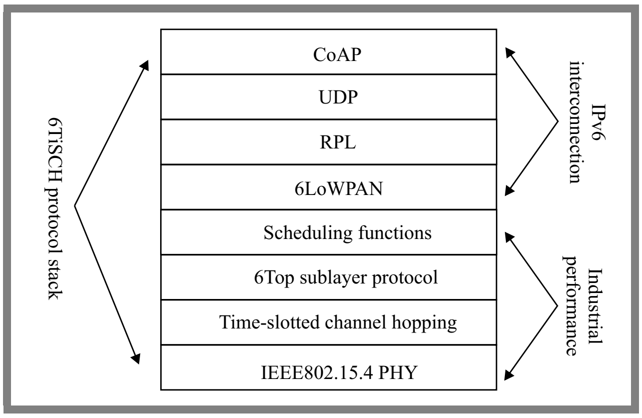


Fig. 3. 6TiSCH protocol stack.

a solution for routing protocol in WSNs. It defines a structure for efficient and energy-saving transmission of packets. The RPL protocol creates a destination-oriented directed acyclic graph (DODAG). In a DODAG, the edges do not form a loop and are laid out so that they face towards the root node. The sink node is the coordinator that acts as the graph root and is the destination for all packets. To prevent cycles in DODAG, a target function is defined to determine the rank of the nodes. A node's rank is a numerical value that shows the place of the node within the DODAG. The closer a given node is to the sink, the lower its rank. In fact, the rank indicates the relative position of each node to the root. The nodes regularly inform each other about their ranks via periodic messages of DODAG information objects (DIO) [1].

3. 6TiSCH Network

6TiSCH brings IPv6 to low-power industrial wireless networks and establishes communication between time-slotted channel hopping (TSCH) networks and 6LoWPAN networks [9]. 6TiSCH combines industrial functions with IPv6 and provides a mechanism by which the link layer topology is adapted to the routing topology [10]. Figure 3 shows the protocol stack of 6TiSCH, using different protocols and standards in each layer [11], i.e., IEEE 802.15.4 in the physical layer, TSCH in the media access control (MAC) layer, SF as a scheduling algorithm and the 6Top protocol. In the network layer, the RPL protocol is used for routing and constructing the topology, and 6LoWPAN is used for IPv6 address mapping for the nodes and for compressing IPv6 packet headers. The UDP protocol is used in the transmission layer, when error detection and correction are not necessary. It is also used in running programs to avoid excessive process overheads. UDP is used for time sensitive applications. The CoAP protocol [12] is also used in the application layer as one of the most widely used protocols in the IoT. This protocol is standardized by targeting limited nodes in low-power networks. This protocol gives devices the ability to connect to a wider range of networks.

4. Centralized Scheduling Algorithm

In this type of scheduling, it is the central node that assumes the role of the host and is responsible for the formation and maintenance of network scheduling. All the network nodes periodically inform the host about the list of their neighbors and the data they generate. Using the gathered information, the host creates a network bonding graph and allocates an appropriate number of time slots to the bonding links based on the information concerning the amount of the data produced by each node. Once the scheduling has been completed, the admin node provides each node with its schedule. In real-world environments, networks have a main node called as a sink. In centralized scheduling, the host node is usually in charge of this task. As this node has knowledge about the entire network, the centralized solution offers an optimal schedule. In scenarios in which the topology is changing and there is a need for network reconstruction, the distributed solution is a more suitable choice [1].

5. Literature Review

Palattella *et al.* [13] proposed a basic method called TASA which is a centralized scheduling algorithm based on network topology and traffic load. Unfortunately, the TASA algorithm was less efficient than other algorithms. Jin *et al.* [14] proposed the adaptive multi-hop scheduling (AMUS) approach. This method executes three functions: scheduling, experimental cell allocation, and End-of-Q informing mechanism. Choi *et al.* in [15] proposed an algorithm for centralized scheduling, called CLS. Using the minimum number of control messages, CLS creates an efficient schedule multi-hop schedule, as it allocates and moves the slots without re-scheduling the entire application. Paper [16] used a routing structure based on multiple paths in which every router has several further hops ahead of it to increase fault tolerance. The authors also proposed a scheduling algorithm in which several senders that are connected to a single receiver are scheduled within a cell. This method is useful in multipath applications.

Sebastian *et al.* [17] developed a method for implementing multiple sinks based on DODAG. In this method, the topology contains multiple DODAGs which have separate sink nodes. There is no connection between these nodes, and such an approach reduces delay. This method can be suitable for considering the multi-sink approach in a network. Only few studies have been performed recently focusing on scheduling in networks with the possibility of creating multiple sinks. Some of the main reasons behind using multiple sinks in a network include the following: increasing network lifetime, reducing power consumption, and improving reliability. Therefore, the aim of this study is to implement multiple sinks in 6TiSCH networks as separate DODAGs and utilize a centralized scheduling algorithm, adopted from [15], to improve the use of multiple sinks in a 6TiSCH simulator [18], as well as to enhance power consumption, network lifetime, packet delivery rate, and delay.

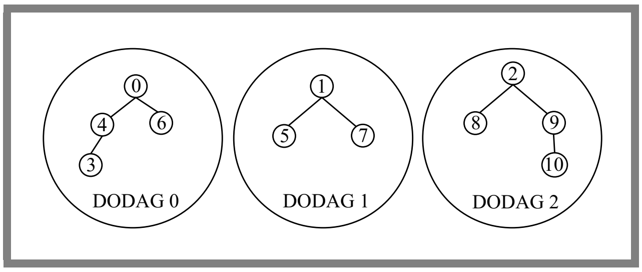


Fig. 4. Topology with three sinks.

6. The Proposed Multi-sink CLS

In the proposed method, several sinks are implemented in a 6TiSCH simulator [18], [19]. We increase the number of sinks to three (Fig. 4) and we select nodes 0, 1, and 2 as sinks. Given this, formation about network topology changes and the network initially contains three sinks that are connected to each other. Through these connections, the nodes can send packets to other sinks. This will increase the delay. According to [17], we use the idea of Multi-DODAGs and apply a limitation by removing the connection between the sinks. Therefore, the topology now contains three DODAGs with three separate sinks, and delay is improved.

We use the CLS algorithm [15] in a multi-sink mode. The proposed method is named MCLS. This algorithm has two main features. It is a multi-hop scheduling algorithm that considers the order of scheduling the links, simultaneously taking into account interference during the scheduling process. Thank to such an approach, the receiver node cannot receive data simultaneously from several sender nodes, because the signals may interfere with each other. Given these two features, the process of scheduling is as follows.

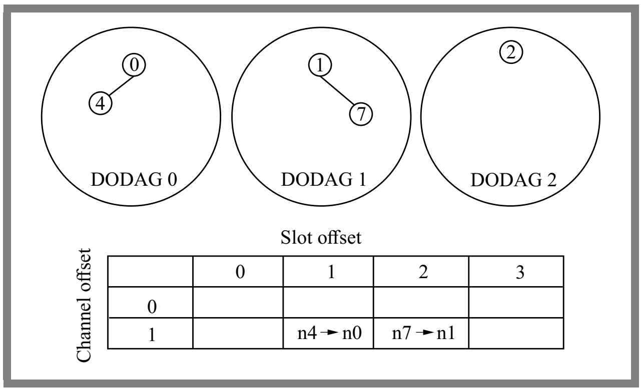
First, a scheduling matrix should be formed that is available to all nodes. The nodes can analyze the information in the matrix before scheduling and then cell allocation is performed. Cell allocation refers to the attempt to find the appropriate time and channel for the intended node. In TSCH, the number of the slot number is called slot offset and the channel number is called channel offset. Slot offset and channel offset are also determined by the number of matrix rows and columns, respectively. The strategy of scheduling and allocating suitable times and channels is based on interference level. Scheduling is performed for a given node using the distance of each node from its parent, i.e., hop distance or node rank. Thus, for nodes which are at a distance of one hop from the sink node, the slot offset from the first time slot is set, equaling 1 [15]. The first time slot for scheduling is set as 1, because time slot 0 is used to send initial packets, such as beacon and RPLDIO. The channel offset value is calculated as [15]:

$$ChannelOffset = \text{rank}(hopdistance) - 1 \ , \qquad (1)$$

where rank is the node's distance from its parent.

When the nodes are at a distance of several hops from the sink node, the slot offset value can be calculated according to [15]:

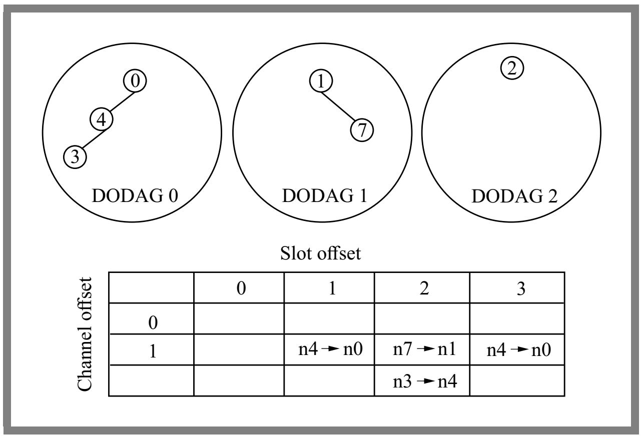
$$SlotOffset = SlotOffset(parent) + 1 \ . \qquad (2)$$



	0	1	2	3
0				
1		n4 → n0	n7 → n1	

Fig. 5. Scheduling for a one-hop scenario.

In Eq. (2), the value of slot offset (parent) is calculated with reference to the slot offset of the parent which has connected to the network beforehand and for which cell allocation has already been completed. The value of channel offset can be calculated by Eq. (1). Scheduling is performed based on a dynamic strategy. This means that when the simulator is run, scheduling and cell allocation are performed for a node at the moment when it connects to the network. Figures 5–6 show the scheduling and cell allocation processes for one-hop and multi-hop scenarios. Figure 7 shows the pseudocode of the MCLS allocation scheduling algorithm for cell allocation. This algorithm is executed for each node that joins DODAG.



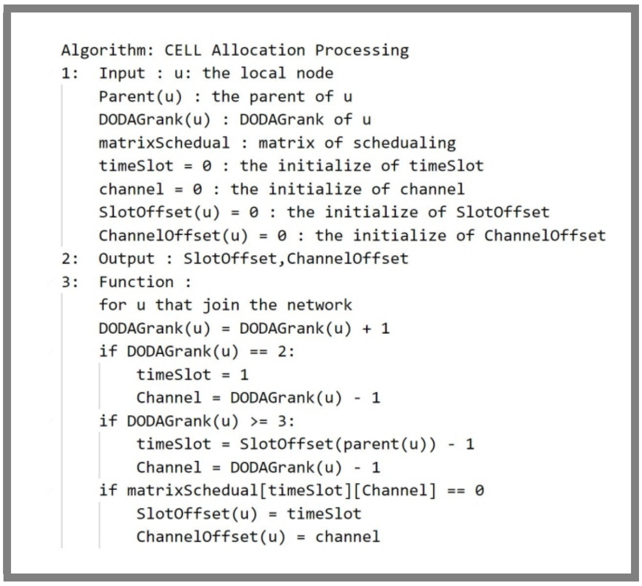
	0	1	2	3
0				
1		n4 → n0	n7 → n1	n4 → n0
			n3 → n4	

Fig. 6. Scheduling for a multi-hop scenario.

7. Simulation and Evaluation

The proposed MCLS method was simulated using 6TiSCH [18], [19]. It is a Phyton based open-source tool which has been developed by the IETF 6TiSCH work group. This simulator measures different criteria, such as end-to-end delay, end-to-end reliability, power consumption, and network lifetime. Using this simulator, we compare the results of MCLS with the following methods from the literature:

- the CLS [15] centralized scheduling algorithm that assumes a single sink node in the network,
- Multi-DODAG from [17] which implemented multiple sinks in separate DODAGs in a network, lacking a scheduling algorithm for the nodes.



```
Algorithm: CELL Allocation Processing
    Input : u: the local node
    Parent(u) : the parent of u
    DODAGrank(u) : DODAGrank of u
    matrixSchedual : matrix of scheduling
    timeSlot = 0 : the initialize of timeSlot
    channel = 0 : the initialize of channel
    SlotOffset(u) = 0 : the initialize of SlotOffset
    ChannelOffset(u) = 0 : the initialize of ChannelOffset
    Output : SlotOffset,ChannelOffset
    Function :
    for u that join the network
    DODAGrank(u) = DODAGrank(u) + 1
    if DODAGrank(u) == 2:
        timeSlot = 1
        Channel = DODAGrank(u) - 1
    if DODAGrank(u) >= 3:
        timeSlot = SlotOffset(parent(u)) - 1
        Channel = DODAGrank(u) - 1
    if matrixSchedual[timeSlot][Channel] == 0
        SlotOffset(u) = timeSlot
        ChannelOffset(u) = channel
```

Fig. 7. MCLS scheduling algoritm.

Tab. 1. Simulation parameters for the first scenario.

Parameter	Value
Number of nodes	10, 12, 14, 16, 18, 20
Duration of simulation	260 s
Routing algorithm	RPL
Packet size	90 bytes
Packet transmission period	60 s

Tab. 2. Simulation parameters for the second scenario.

Parameter	Value
Number of nodes	20
Duration of simulation	330 s
Routing algorithm	RPL
Packet size	90 bytes
Packet transmission period	20, 30, 40, 50, 60 s

The criteria for evaluation in each scenario include network lifetime, reliability, packet transmission rate, and delay. In order to make more accurate interpretations of the results, we will keep to a confidence level of 99%. Moreover, the location of each node is randomly arranged around the sink and in a 2 × 2 square space.

In the first scenario, the performance of MCLS is examined in terms of an increase in the number of nodes as well as scalability. The simulation parameters of the first scenario are listed in Tab. 1. Figure 8a shows that the average lifetime of the network is greater in the proposed method. The reason is that reduced power consumption and traffic loads enhance network lifetime. Figure 8b shows the average end-to-end delay. As one may notice, in some cases the delay in MCLS is lower than in CLS, especially where the number of nodes is greater. The reason is that, in multiple sinks, the distance between the nodes and the sinks is reduced and packets arrive at the sinks with a shorter delay. It may be noticed that MCLS has significantly decreased the delay of the Multi-DODAG method. As shown in Fig. 8c, the packet delivery rate (expressed in percentages) in MCLS and CLS is better than in Multi-DODAG, because, as mentioned earlier, the Multi-DODAG method does not perform scheduling for nodes. Moreover, as MCLS uses multiple sinks and reduces the network's traffic load, it decreases the probability of packet loss and increases the packet delivery rate.

Tab. 3. Simulation parameters for the third scenario.

Parameter	Value
Number of nodes	20
Duration of simulation	254 s
Routing algorithm	RPL
Packet size	90 bytes
Packet transmission period	20, 30, 40
Number of sink nodes	1, 3, 5, 7, 9 s

Tab. 4. Simulation parameters for the fourth scenario.

Parameter	Value
Number of nodes	20
Duration of simulation	254 s
Routing algorithm	RPL
Packet size	90 bytes
Packet transmission period	1, 5, 7, 10 s

In the second simulation scenario, the performance of MCLS is examined in terms of the number of packets sent and traffic load. The simulation parameters are listed in Tab. 2. As shown in Fig. 9a, MCLS performs better than the two other methods at higher traffic rates. If the nodes send a packet every 20 s, the nodes' lifetime will still remain longer in MCLS. This is explained by the presence of the sink nodes in the network, as these can manage large traffic loads. Figure 9b shows the changes in the average end-to-end delay for different packet transmission durations. The average delay in MCLS has decreased in comparison with the two remaining methods, due to the smaller number of hops required to arrive at the destination. Figure 9c shows the changes in the average packet delivery rate for different packet transmission durations. In MCLS, due to the existence of multiple sinks in the network, the network's traffic load for different packet delivery durations is distributed among the sink nodes. Therefore, the packet delivery rate of the sink nodes is greater than in the two other methods, even when the packets are sent every 20 s and the network is very crowded.

In the third scenario, the performance of MCLS is examined based on the number of sink nodes (Tab. 3).

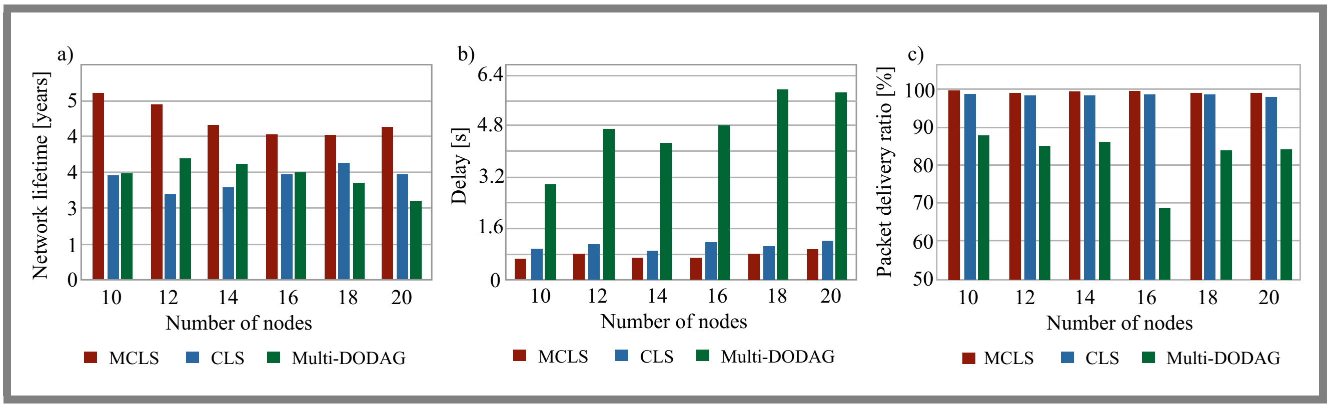


Fig. 8. Simulation results for scenario 1: a) average network lifetime, b) average delay, and c) average packet delivery rate.

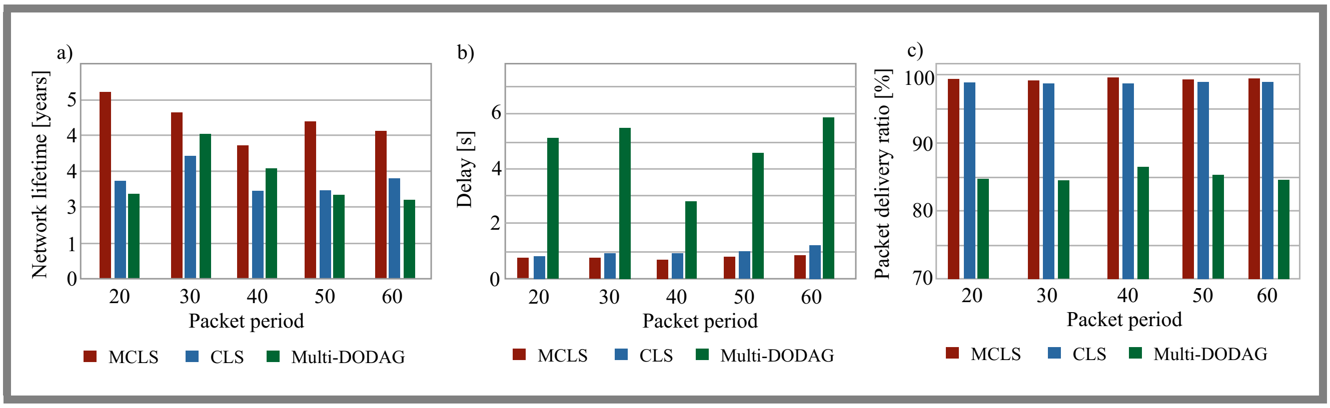


Fig. 9. Comparison of: a) average lifetime, b) average end-to-end delay, and c) average packet delivery ratio.

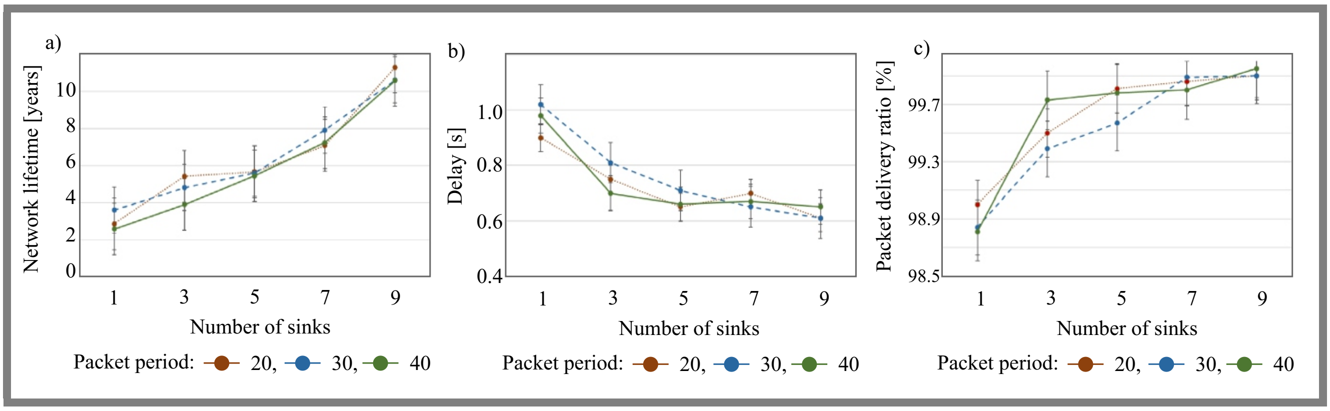


Fig. 10. Comparison of average: a) lifetime of networks, b) delay, and c) packet delivery ratio with different number of sink nodes.

As one may see in Fig. 10, as the number of sink nodes in the network increases and for different packet transmission durations, such parameters as network lifetime, delay, and packet delivery rate improve as well. The reason is that the nodes consume less power for packet transmission and, therefore, have a longer lifetime.

On the other hand, packets are delivered with smaller delays due to the decrease in the number of hops from the source to the destination node. Also, due to the existence of multiple sinks, the network's traffic load for different packet delivery durations is distributed among the individual sink nodes. This causes an increase in packet delivery rate in the sink nodes.

As can be seen in the comparison with Multi-DODAG and CLS, MCLS has improved network lifetime and packet delivery rate, simultaneously decreasing delay.

In the fourth scenario, the network's congestion control capabilities are examined. For this purpose, by adding the number of packets sent per second, network lifetime parameters, packet reception and delay are analyzed (Tab. 4). As shown in Fig. 11, along with the growing network congestion, the proposed method works better and the parameters of lifetime, latency and packet delivery are improved.

In the proposed method, due to the presence of several sink nodes, when the number of packets sent from the nodes increases, the network can manage the congestion more effectively. Therefore, the packet loss rate is reduced and the

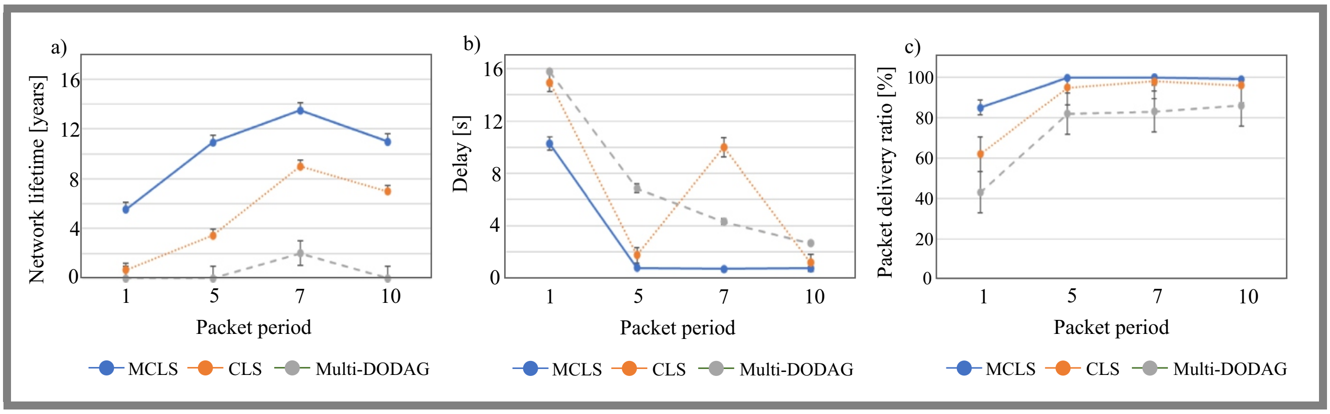


Fig. 11. Comparison of average: a) delay, b) packet delivery ratio, and c) network lifetime for different packet period.

nodes do not need to resend them. They consume less energy and, finally, the overall lifetime of the network increases compared to the two other methods.

On the other hand, due to the presence of several sink nodes and the reduction of the number of steps between the source and the destination, the packet sending delay is reduced.

8. Conclusion

In this paper, we implemented multiple sinks in the 6TiSCH simulator and use a centralized scheduling algorithm. To implement multiple sinks, we used the Multi-DODAG method which lacks a scheduling component. For the purpose of scheduling the nodes, we used the CLS method, i.e. a centralized scheduling algorithm that contains one sink node only. The proposed method is called MCLS and uses both multiple sinks, just like Multi-DODAG, and a centralized scheduling method, just like the CLS method. According to the simulation results, MCLS has overcome the limitation of using one sink in CLS and has outperformed Multi-DODAG by adopting the centralized scheduling algorithm of CLS. As discussed in the evaluation section, the criteria of network lifetime, end-to-end delay, and packet delivery rate in different scenarios were improved for each of the methods.

References

[1] S. Scanzio *et al.*, "Wireless Sensor Networks and TSCH: A Compromise Between Reliability, Power Consumption, and Latency", *IEEE Access*, vol. 8, pp. 167042–167058, 2020 (`https://doi.org/10.1109/ACCESS.2020.3022434`).

[2] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The Industrial Internet of Things (IIoT): An Analysis Framework", *Computers in Industry*, vol. 101, pp. 1–12, 2018 (`https://doi.org/10.1016/j.compind.2018.04.015`).

[3] K. Gulati *et al.*, "A Review Paper on Wireless Sensor Network Techniques in Internet of Things (IoT)", *Materials Today: Proceedings*, vol. 51, no. 1, pp. 161–165, 2022 (`https://doi.org/10.1016/j.matpr.2021.05.067`).

[4] C.-Y. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges", *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003 (`https://doi.org/10.1109/JPROC.2003.814918`).

[5] A. Sari, A. Lekidis, and I. Butun, "Industrial Networks and IIoT: Now and Future Trends", *in Industrial IoT, Challenges, Design Principles, Applications, and Security*, pp. 3–55, Springer, 2020 (`https://doi.org/10.1007/978-3-030-42500-5_1`).

[6] A. Gezer and S. Okdem, "Improving IEEE 802.15.4 channel access performance for IoT and WSN devices", *Computers & Electrical Engineering*, vol. 87, art. no. 106745, 2020 (`https://doi.org/10.1016/j.compeleceng.2020.106745`).

[7] D. De Guglielmo, S. Brienza, and G. Anastasi, "IEEE 802.15.4e: A survey", *Computer Communications*, vol. 88, pp. 1–24, 2016 (`https://doi.org/10.1016/j.comcom.2016.05.004`).

[8] H. Kharrufa, H.A.A. Al-Kashoash, and A.H. Kemp, "RPL-based Routing Protocols in IoT Applications: a Review", *IEEE Sensors Journal*, vol. 19, no. 15, pp. 5952-5967, 2019 (`https://doi.org/10.1109/JSEN.2019.2910881`).

[9] M.R. Palattella *et al.*, "Standardized Protocol Stack for the Internet of (Important) Things", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2012 (`https://doi.org/10.1109/SURV.2012.111412.00158`).

[10] X. Vilajosana *et al.*, "IETF 6TiSCH: A Tutorial", *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 595–615, 2019 (`https://doi.org/10.1109/COMST.2019.2939407`).

[11] T. Chang *et al.*, "6TiSCH on SCμM: Running a Synchronized Protocol Stack without Crystals", *Sensors*, vol. 20, no. 7, art. no. 1912, 2020 (`https://doi.org/10.3390/s20071912`).

[12] A. Bhattacharjya, X. Zhong, J. Wang, and X. Li, "CoAP-Application Layer Connection-less Lightweight Protocol for the Internet of Things (IoT) and CoAP-IPSEC Security with DTLS Supporting CoAP", *in: M. Farsi, A. Daneshkhah, A. Hosseinian-Far, H. Jahankhani (Eds.) Digital Twin Technologies and Smart Cities. Internet of Things*, pp. 151–175, Springer Cham, 2020 (`https://doi.org/10.1007/978-3-030-18732-3_9`).

[13] M.R. Palattella, N. Accettura, M. Dohler, L.A. Grieco, and G. Boggia, "Traffic Aware Scheduling Algorithm for Reliable Low-power Multi-hop IEEE 802.15. 4e Networks", *in 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 327–332, 2012 (`https://doi.org/10.1109/PIMRC.2012.6362805`).

[14] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A Centralized Scheduling Algorithm for IEEE 802.15.4e TSCH Based Industrial Low Power Wireless Networks", *in 2016 IEEE Wireless Communications and Networking Conference*, pp. 1–6, 2016, (`https://doi.org/10.1109/WCNC.2016.7565002`).

[15] K.-H. Choi and S.-H. Chung, "A New Centralized Link Scheduling for 6TiSCH Wireless Industrial Networks", *in Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, Springer, 2016, pp. 360–371 (`https://doi.org/10.1007/978-3-319-46301-8_30`).

[16] E.M. Ahrar, M. Nassiri, and F. Theoleyre, "Multipath Aware Scheduling for High Reliability and Fault Tolerance in Low Power Industrial Networks", *J. of Network and Comp. Applications*, vol. 142, pp. 25–36, 2019 (`https://doi.org/10.1016/j.jnca.2019.05.013`).

[17] A. Sebastian and S. Sivagurunathan, “Multi DODAGs in RPL for Reliable Smart City IoT”, *Journal of Cyber Security and Mobility*, vol. 7, no. 1, pp. 69–86, 2018 (`https://doi.org/10.13052/jcsm2245-1439.716`).
[18] E. Municio *et al.*, “Simulating 6TiSCH Networks”, *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 3, art. no. e3494, 2019 (`https://doi.org/10.1002/ett.3494`).
[19] T. Chang, M. Vučinić, X.V. Guillén, D. Dujovne, and T. Watteyne, “6TiSCH Minimal Scheduling Function: Performance Evaluation”, *Internet Technology Letters*, vol. 3, no. 4, art. no. e170, 2020 (`https://doi.org/10.1002/itl2.170`).

Nikta Azimian, M.Sc. Student
Department of Computer Engineering
E-mail: nikta76azimian@gmail.com
Bu-Ali Sina University, Hamedan, Iran
`https://basu.ac.ir`

Reza Mohammadi, Ph.D., Assistant Professor
Department of Computer Engineering
`https://orcid.org/0000-0002-2139-4518`
E-mail: r.mohammadi@basu.ac.ir
Bu-Ali Sina University, Hamedan, Iran
`https://basu.ac.ir`

Mohammad Nassiri, Ph.D., Associate Professor
Department of Computer Engineering
`https://orcid.org/0000-0002-0703-5949`
E-mail: m.nassiri@basu.ac.ir
Bu-Ali Sina University, Hamedan, Iran
`https://basu.ac.ir`