International Journal of Internet, Broadcasting and Communication Vol.14 No.1 53-63 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.1.53

IJIBC 22-1-8

TSCH-Based Scheduling of IEEE 802.15.4e in Coexistence with Interference Network Cluster: A DNN Approach

Md. Niaz Morshedul Haque and Insoo Koo*

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, Korea
*niazhaquebd@gmail.com, leeyd1004@naver.com, and iskoo@ulsan.ac.kr**

Abstract

In the paper, we propose a TSCH-based scheduling scheme for IEEE 802.15.4e, which is able to perform the scheduling of its own network by avoiding collision from interference network cluster (INC). Firstly, we model a bipartite graph structure for presenting the slot-frame (channel-slot assignment) of TSCH. Then, based on the bipartite graph edge weight, we utilize the Hungarian assignment algorithm to implement a scheduling scheme. We have employed two features (maximization and minimization) of the Hungarian-based assignment algorithm, which can perform the assignment in terms of minimizing the throughput of INC and maximizing the throughput of own network. Further, in this work, we called the scheme "dual-stage Hungarian-based assignment algorithm". Furthermore, we also propose deep learning (DL) based deep neural network (DNN) scheme, where the data were generated by the dual-stage Hungarian-based assignment algorithm. The performance of the DNN scheme is evaluated by simulations. The simulation results prove that the proposed DNN scheme provides similar performance to the dual-stage Hungarian-based assignment algorithm while providing a low execution time.

Keywords: *TSCH, IEEE802.15.4e, Interference Network Cluster(INC), Hungarian Assignment Algorithm, Deep Learning*

1. Introduction

The scarcity of spectrum is a burning issue over the world. It has occurred with the increasing uses of the unlicensed industrial, scientific, and medical (ISM) radio bands. According to the expert report, mobile devices have been increased from 8 billion in 2016 to 11.5 billion in 2021. As a result, data flow from these devices has grown from 7.2 to 49.0 exabytes per month [1]. In the meantime, the internet of things (IoT) applications is also increasing. The industrial internet of things (IIoT) is a promising application of IoT. IIoT uses the ISM band to develop new technology, establish short-range communication, device-to-device communication, etc. [2].

According to IEEE 80215.4e, the TSCH (time-slotted channel hopping) is a useful resource in the industrial internet of things (IIoT). TSCH works with a slot-frame structure of numerous channel-offsets and numerous slot offsets [3],[4]. It is gaining popularity in industrial applications due to its simple architecture and low power consumption. The TSCH follows the media access control (MAC) mechanism, which covers the reconfiguration, enumeration, composition, and data transmission [5]. The slot frame is the central

Manuscript Received: November. 11, 2021 / Revised: November. 15, 2021 / Accepted: November. 18, 2021
Corresponding Author: iskoo@ulsan.ac.kr
Tel: +82-52-259-1249Fax: +82-52-259-1686
Professor, Department of Electrical, Electronic and Computer Engineering, University of Ulsan, South Korea

communication unit of TSCH; a pair of nodes is needed for data transmission. A slot frame is a series of time slots that are repeated continually. A different channel is assigned pseudo-randomly to each timeslot. The schedule defines which node to interact with the which neighbor node and what channel offset should be used for this transmission [6].

At present cross-technology interference is another vital issue due to the excessive use of the unlicensed band, such as wi-fi conflicts with Bluetooth. TSCH-based scheduling method utilizes the channel hopping method to solve these coexistence issues [7],[8]. Most of the TSCH-based scheduling schemes consider their own network's performance (e.g., optimal throughput, accurate channel state, latency, etc.) without considering interference from surrounding networks or noise from other sources [9]. However, channel hopping from one to the next is not an effective solution in a pragmatic scenario since all frequencies face different interference levels [10],[11]. Furthermore, in a practical context of the wireless network, channels follow random nature, which is defined by different mathematical algorithms considering interference [12].

In this paper, we propose a TSCH-based scheduling algorithm, which can observe the behavior of the surroundings interference networks. Here, we consider two networks: the first one is known called "own network," and the second one is unknown, called "interference network cluster (INC)", which may also include interferences (e.g., noise, microwaves) from other sources [13], [14]. Furthermore, the slot-frame structure of TSCH-based scheduling is modelled by a bipartite graph for establishing channel slot assignment of networks, where edge weight is considered both network's (own network and INC) throughput. By utilizing the bipartite edge weight, we have proposed dual-stage Hungarian-based assignment (one for INC and another for own network). The proposed method can perform the TSCH-based scheduling by avoiding collisions from INC's by maximizing the own network's throughput, minimizing the INC's throughput, and establishing a relation between them.

In the last couple of years, deep learning (DL)-based algorithms have been utilized in the application and management of wireless communication. We got a complete outline regarding DL algorithms to the application of wireless resource allocation from the study in [15]. The study presents the limitations of traditional optimization methodologies as well as the potentialities of DL paradigms in wireless networks. Furthermore, we observed that the DL-based DNN scheme provided an outstanding performance in wireless power communication networks (WPCN). Where the authors formulated problems regarding power allocation to the communication networks [16] and energy beamforming vector in multiple antennas [17], both were solved by the iterative sequential parametric convex approximation (SPCA) algorithm. The supervised offline training of the DL-based DNN scheme was performed by SPCA-based training data. After completing the offline training, in the testing phase, the DNN performed similarly to the SPCA and contributed to the reduction of computing time.

According to the aforementioned discussion, the objective of this paper is to incorporate a DL method for the TSCH-based scheduling, which can provide the same performance to the dual-stage Hungarian-based scheduling scheme while reducing the execution time of scheduling.

The contributions of this paper can be summarized as follows:

- We model a TSCH network example coexistence with INC and formulate algorithms to produce their throughputs (own network and INC) as well as make an interconnection between them.
- We present a bipartite assignment (slot and channel assignment) graph model for the slot-frame structure of TSCH-based scheduling.
- Based on the bipartite graph edge weight information (throughput of own network and INC), we propose a dual-stage Hungarian-based assignment algorithm for executing TSCH-based scheduling, which is able to make data transmission scheduling and avoid collision between own network and INC. Furthermore,

this scheme can maximize the throughput of own network and minimize the throughput of INC.

- We propose a deep learning-based DNN scheme for TSCH-based scheduling to reduce the execution time. The proposed DNN scheme uses the training data obtained by the dual stage Hungarian-based assignment algorithm to avoid a collision from INC. Thus, the proposed DNN scheme correctly emulates the dual-stage Hungarian-based scheduling scheme and reduces the execution time of scheduling.
- The performance of the DNN scheme is verified by Simulation. The simulation results show that the proposed DNN scheme exhibits a similar performance to the dual-stage Hungarian-based scheduling algorithm as well as provides an additional contribution to reduce the computational time.

The remainder of the paper is arranged as follows. In Section 2, we demonstrate the system model and problem statement. In Section 3, we describe the proposed dual-stage Hungarian algorithm-based scheduling scheme. In Section 4, we delineate the training of the DNN scheme and the classification of the dataset. In Section 5, performance of the proposed scheme is illustrated and verified by simulations. Finally, Section 6 concludes this paper.

2. System Model

In this section, we describe the network model, state the mathematical equations for the problem formulation and channel model.

2.1 Network Model

In this paper, we consider the TSCH network consists of N nodes called the own network and a second unknown network called the interfering network cluster (INC), which may include noise from different sources (e.g., microwaves, noise generators, and jammers) [13],[14]. We presume that both networks will have similar priorities and rights to use the spectrum because we focus on unlicensed bands. However, as the INC is unknown, the own network has no idea about what technologies are being used in INC, traffic types, channel state information (CSI), etc. The TSCH network coexistence with INC is shown in Figure 1.

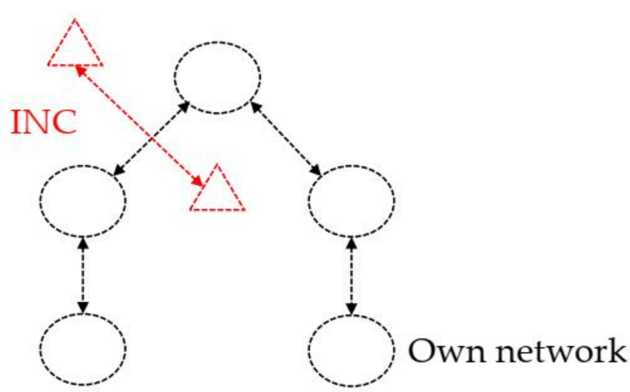

Figure 1. Example of TSCH network coexistence with INC

Furthermore, the TSCH uses time and frequency diversity. It's the combination of time division multiple access (TDMA) and frequency-division multiple access (FDMA) techniques and this is core concept of TSCH-based scheduling. Based on this concept, all N nodes of own TSCH network can communicate with one another on channel $c \in \mathbb{C}$. Each channel is separated into time slot $s \in S$. S time slots and $\mathbb{C}$ channels are available for each slot-frame, $f \in F$ in the TSCH schedule, as shown in Figure 2.

s=0 c=0	*s=1 c=0*	*s=2 c=0*	… …	*s=S c=0*
s=0 c=1	*s=1 c=1*	*s=2 c=1*	… …	*s=S c=1*
………………………			… …	……
s=0 c=ℂ	*s=1 c=ℂ*	*s=2 c=ℂ*	… …	*s=S c=ℂ*

Figure 2. Corresponding slot-frame of TSCH-based scheduling

2.2 Problem Formulation

In this paper, for TSCH-based scheduling, we consider all nodes of own network because the information of INC is entirely unknown such as any applied protocols, features, channel conditions, etc. We only know if any of the nodes of INC uses a channel at a given time of a slot frame. We assume that $P_{s,c}^{f} = 1$ only if the INC used channel *c,* at slot *s*, in frame *f,* at time *(f ,s)*.

In this work, a centralized scheduler is considered for TSCH-based scheduling, and the own network cannot schedule two transmissions (*TX)* actions between two nodes simultaneously using the same channel. We also assume that the INC is strong enough to collide with the transmission (node pair communication) of own network.

We also define that transmission of the own network between two nodes on a channel *c* at a time *(f, s),* is successful only if the own node is transmitting at the same time in that channel and no INC node is transmitting. Every node, $n \in N$ of own network executes a transmission action $TX_{s,c}^{f,n}$ uses the channel *c*, at slot *s*, for slot-frame *f.* Own node is transmitting at the time in that channel and no interfering node is transmitting. Here we use $\gamma_{s,c}^{f,n}$ to determine if an own packet could be successfully delivered or not.

$$\gamma_{s,c}^{f,n} = 1,\ if\ TX_{s,c}^{f,n} = 1 \wedge\ P_{s,c}^{f} = 0 \tag{1}$$

Similarly, the communication of INC is successful if $\gamma_{s,c}^{\prime f,n} = 1$. We can have $\gamma_{s,c}^{\prime f,n} = 1$ in the following case:

$$\gamma_{s,c}^{\prime f,n} = 1,\ if\ TX_{s,c}^{f,n} = 0 \wedge\ P_{s,c}^{f} = 1 \tag{2}$$

2.3 Channel Model

We design and develop a channel model to fulfill the condition of equation 1 and equation 2. As we discussed earlier, our goal is to propose a bipartite graph-based solution of the TSCH network. To establish the bipartite graph, we need to know the knowledge of throughput. Throughput is considered the main component for bipartite edge weight to solve the TSCH-based scheduling of channel and slot assignment [3],[4].

As INC is unknown, we do not have any knowledge about their channel state information (CSI) and other parameters [13],[14]. In this work, we have considered the throughput of INC as a random variable. We use the linear congruential generator for constituting INC throughput. A linear congruential generator (LCG) is an algorithm that generates a series of pseudorandom integers using a piecewise linear equation. We have found the most well-known pseudorandom number generator algorithm [18]. The throughput of INC will be executed by the following equation of linear congruential generator (LCG) such that we have

$$Y_i = (aY_{i-1} + q)(mod\ M) \tag{3}$$

where M is modular, $i = \{1, \dots, \mathbb{C}S\}$, multiplicator a and q are two suitable chosen integers. We can have the uniform random variable after scaling, U_i as following:

$$U_i = \frac{Y_i}{q} \tag{4}$$

Another goal is to minimize the INC throughput for enhancing the reliability of own network. The INC's throughput can be minimized by the following equation:

$$U^* = min \sum_{s \in S} \sum_{c \in \mathbb{C}} U_i \tag{5}$$

The own network channel state of each $c \in \mathbb{C}$ at slot $s \in S$ is defined by $x = X_{c,s} = |H_{c,s}|^2$ where $H_{c,s}$ is the channel gain, which is related to the fading phenomenon [4]. In this paper we consider that channel gain will follow Rayleigh fading and that it will be determined by the probability density function (pdf) below:

$$F_R(z) = \frac{2z}{\Omega} e^{\frac{-z^2}{\Omega}} \tag{6}$$

where $\Omega = E[R^2]$, and R is a random variable.

The transmission power p is assumed to be constant, and the throughput of own network is determined based on the minimization of INC throughput U^* with the Shannon's formula as shown below:

$$W_i(x) = \frac{\beta}{l} log(1 + \frac{xp}{\beta \eta_o + U^*}) \tag{7}$$

where β is the receiver signal's bandwidth, l denotes packet size, and η_o means noise variance.
Then, we maximize the own network throughput by following equation:

$$W^* = max \sum_{s \in S} \sum_{c \in \mathbb{C}} W_i \tag{8}$$

3. The Proposed Dual-Stage Hungarian Based Assignment Algorithm

In this section, we describe the modeling of a bipartite graph for executing slot channel assignment of TSCH-based scheduling. Based on the bipartite model of the TSCH slot frame structure, we utilize the dual-stage Hungarian-based assignment techniques to execute TSCH-based scheduling, which is able to perform the transmission schedule of own network node pairs without collision from INC.

According to graph theory-based mathematical structure, a bipartite graph has two independent sets of vertices: top and bottom. Every edge connects a vertex in the top to one on the bottom. It is used for modeling relationships between two different classes of objects [19]. In recent times, the bipartite graph is a promising application of the TSCH based scheduling for presenting an assignment of slots and channels for successful node pair transmission (TX) [3,4], where throughput is considered the edge's weight of a bipartite graph [4].

In this paper, we consider a bipartite graph $\mathcal{B} = (S, \mathbb{C}, E)$ correspond to the slot-frame matrix in Figure 2 The set of channels, $c \in \mathbb{C}$ is the top vertexes of bipartite graph, and the set of slots $s \in S$ is the bottom vertexes of bipartite graph $\mathcal{B}$. The set of edge weights is considered $E = \{e = (s, c) | s \in S, c \in \mathbb{C}\}$ as the throughputs of own network and INC.

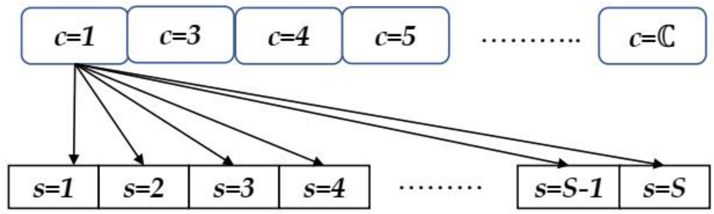

Figure 3. Bipartite graph corresponding to the Figure 2

Based on the knowledge of bipartite edge weight, we propose a dual-stage Hungarian-based assignment algorithm for executing TSCH-based scheduling to avoid collision between own network and INC. We consider bipartite edge weights are the throughput of own network and INC (section 2.3) as well as the negligible upper bounds 2^L (L is the number of node-to-node edge Figure:1)

The cost matrix of Hungarian algorithm is the total number of edges $S * \mathbb{C}$ of the bipartite graph. Firstly, the INC throughput U_i was considered as the edge weight of the bipartite graph. The Hungarian algorithm performed the assignment according to the minimization of edge weight of the bipartite graph and we obtained the $P_{s,c}^{f}$ and minimization of INC throughput, U^*. Secondly, the own network throughput W_i will be calculated based on the U^* by considering bipartite edge weight. The Hungarian again performed the scheduling task based on maximization of bipartite edge weight and we obtained the transmission of own network, $TX_{s,c}^{f,n}$ and obtained throughput of own network, W^*. Though most of the collisions between own network and INC were mitigated by this technique, every transmission of own network $TX_{s,c}^{f,n}$ was checked channel by channel and slot by slot. If INC is considered, $P_{s,c}^{f}$ the specifically assigned slot by own network, which means that duplicate schedule had found in that time and own network shifted their scheduling on the next slot according to the second highest edge weight value.

Algorithm 1: Dual-Stage Hungarian-Based Assignment Algorithm

Begin
//*INITIALIZATION*
for each $c \in \mathbb{C}$, $s \in S$ and $f \in F$:
    Run linear congruential generator (LCG) to obtain INC throughput U_i
    Run Hungarian algorithm based on minimization U_i of bipartite edge weight
      %where INC throughput used as bipartite edge weight
    Obtain $P_{s,c}^{f}$ and U^*
    Own network throughput, W_i will be calculated
    Run Hungarian algorithm according to maximization W_i of bipartite edge weight
    % where own network throughput used as bipartite edge weight
    Obtain $TX_{s,c}^{f,n}$ and W^*
    if $TX_{s,c}^{f,n}$=$P_{s,c}^{f}$,
        then $TX_{s,c}^{f,n}$ will happen in next slot
        %based on second maximized bipartite edge weight

```
        else TX_{s,c}^{f,n} happen in the assigned slot according to step 8
    end if
    ensure: TX_{s,c}^{f,n} = 1 ∧ P_{s,c}^{f} = 0 and TX_{s,c}^{f} = 0 ∧ P_{s,c}^{f} = 1
end for
//MAIN LOOP:
Generate 10,000 data frames for DNN training
```

4. The Proposed Deep Learning Based DNN Scheme

In this section, we propose a deep learning-based DNN scheme that uses the training data from a dual-stage Hungarian-based assignment algorithm shown on Figure 4. We generated a total of 10,000 data frames by using the dual-stage Hungarian-based assignment algorithm (Algorithm 1) and divided the dataset into three parts 60% for training, 20% for validation and 20% for testing. Validation is the unbiased method. That is, assessment procedure deals with as to how well our model learns the knowledge during training. Testing is the actual procedure for analyzing neural network performance. After finishing supervised offline training of DNN, the DNN scheme performs the scheduling by learning the kinship between the input and output. We utilize the scheme for every new value of the throughput of own network W_i to find the optimal solution of $TX_{s,c}^{f,n}$.

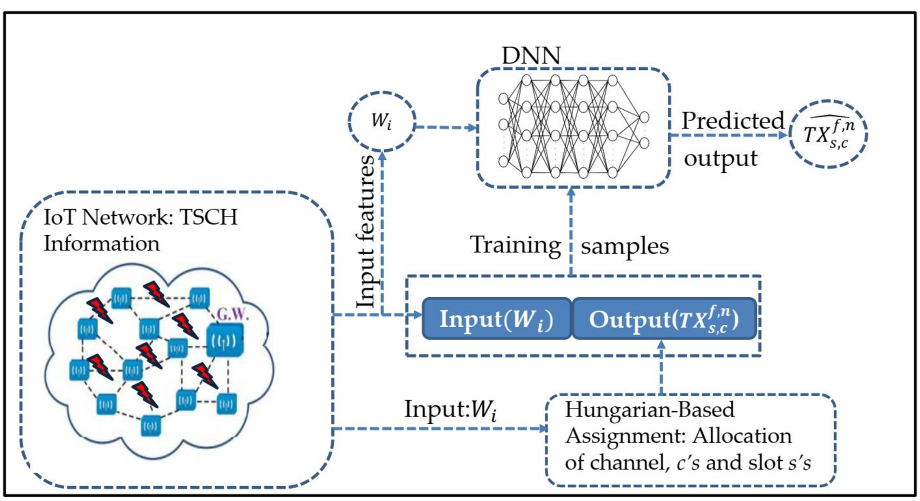

Figure 4. Overall structure of the proposed DNN-based scheme

5. Performance Evaluation

We compared the outcomes of DNN with the Hungarian-based assignment algorithm (HG) to demonstrate the efficiency of the proposed DNN scheme. The simulations were performed in two steps: first, we executed algorithm 1 for generating enough samples data; second, after obtaining data samples, the DNN training was performed. Algorithm 1 for data set generation was carried out in MATLAB on a PC with an Intel Core i7 processor and 8 GB RAM. After getting data samples, the supervised offline training of DNN was executed in Python by utilizing the Keras and NumPy libraries.

2.1 Execution of TSCH-Based Scheduling

The primary goal of this paper is to implement TSCH-based scheduling to avoid Collison from INC. The details of the model are described in Algorithm 1. The DNN training data was generated from this model to train the DNN. The following parameters are considered in Table 1 for establishing Algorithm 1.

Table 1. Parameters for algorithm 1

Parameters	Specification
Number of Channels, $\mathbb{C}$	**12**
Number of Slots, S	**16**
Number of Data Frames, F	**10,000**
Bandwidth, β	**1 MHz**
Bits in Each Packet l	**1000**
Transmission Power, p	**10 mw**

2.2 Building a DNN Scheme

We have considered a DNN a model with one input layer, multiple hidden layers, and one output layer. The number of input features of DNN is the total number of bipartite edge weight, which means the total number of bipartite edge weights is the multiplication of the total number of channels and the total number of slots, $\mathbb{C} * S$. The output considers how many times the own network uses the channels, which means that the total number output of DNN is the total number of channels, $\mathbb{C}$. We have used the rectified linear unit (ReLU) activation function for every hidden layer to obtain minimal min square error (MSE) with little difficulty. The specification of the proposed DNN scheme is given in Table 2.

Table 2. DNN specification

Parameters	Specification
Number of Input Features	**192**
Number of Hidden Layers	**4**
Number of Neurons in Each Layer	**800,1600,1200,800**
Number of Outputs	**12**
Learning Rate	**0.001**
Batch Size	**100**
Number of Epochs	**1000**

2.3 Assignment Method: TSCH-Based Scheduling

Figure 3.5 illustrates the example of transmission, $TX_{s,c}^{f,n}$ assignment of own network that can avoid a collision from INC. Here green color indicates the transmission assignment of own network executed based on the maximization of own network throughput. The red color indicates that INC uses the slots $P_{s,c}^{f}$; it was executed based on the minimization of INC throughput. Lastly, the own network checked channel by channel and slot by slot. For this frame, we observed that the duplicate assignment was executed on channel 3. Both own network and INC used slot 3 for the channel 3 while using the above checking method when own network found INC in the same slot; then it shifted its transmission schedule to the next slot 2 based on the second highest of own network throughput.

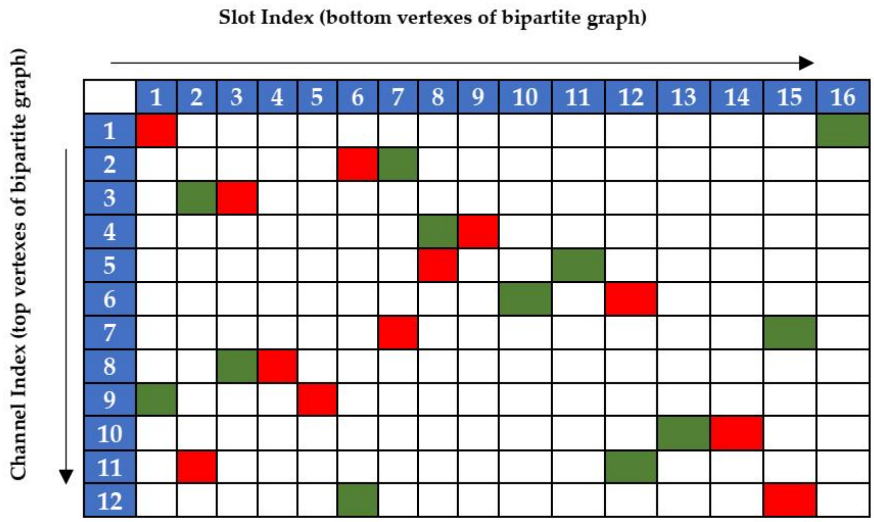

Figure 5. Example of cell assignment

2.4 Throughput Measurement

Figure 6 shows the average maximized throughput of own network and average minimized throughput of INC for different channels. Our proposed dual-stage Hungarian-based assignment is able to maximize the own network throughput based on the minimization of INC throughput. It increases network reliability for data transmission.

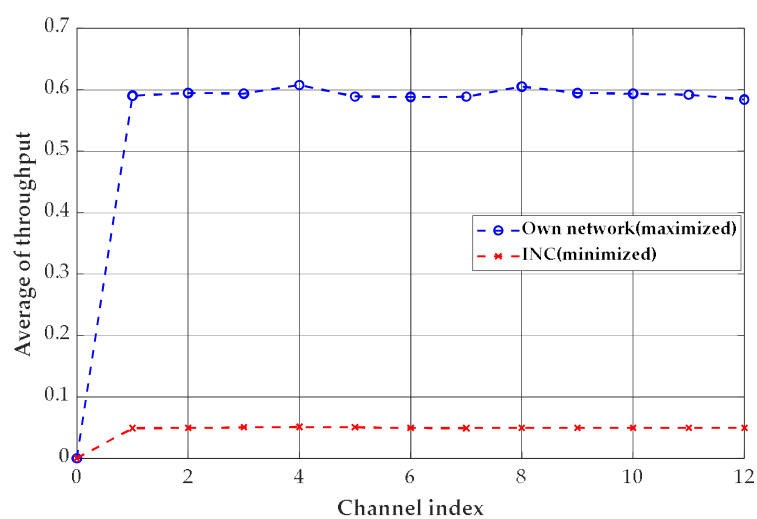

Figure 6. Throughput of own network and INC

2.5 DNN model accuracy

Accuracy is the vital part of the deep learning-based DNN scheme. It defines that how much the proposed learning-based scheme learns the original method. We considered an accuracy metric based on the transmission of own network as given in equation 8. To do this, we firstly determined the accurate parameters based on the following method. The transmission assignment of own network $TX_{s,c}^{f}$ is the output of dual-stage Hungarian-based scheme is an integer but predicted output of DNN, $\widehat{TX_{s,c}^{f}}$ is not an integer value. For alleviating this problem, we used the round function as follows: If $TX_{s,c}^{f} - round(\widehat{TX_{s,c}^{f}}) = 0$, it indicates as accurate, and $TX_{s,c}^{f} - round(\widehat{TX_{s,c}^{f}}) \neq 0$ indicates it is not accurate.

If test samples are $\mathbb{N}_t$, and non-zero or not accurate parameters are indicated by $\mathbb{P}$, accuracy of DNN can be determined with the following equation:

$$Accuracy = \frac{\mathbb{N}_t * \mathbb{C} - \mathbb{P}}{\mathbb{N}_t * \mathbb{C}} * 100\% \quad (9)$$

2.5 DNN performance

We measured the accuracy for 1000 test samples and achieved around **90%** accuracy of the proposed deep learning-based DNN scheme. The over all performance of proposed DL-based DNN scheme is summarized in Table 3.

Table 3. DNN performance

Parameters	Specification
Accuracy	**90%**
HG Execution Time Per Sample	**250 ms**
DNN Execution Time Per Sample	**30 ms**
MSE	**0.023**

Furthermore, the figure 7 shows the performance of the proposed DNN scheme based on measuring the throughput for own network and INC. It reveals that the proposed deep learning-based DNN scheme correctly emulates the proposed dual-stage Hungarian-based assignment scheme. DNN is a learning-based scheme and learns the training data from a dual-stage Hungarian (HG) based scheduling scheme; that is a reason why HG shows the optimal and DNN shows the suboptimal for the throughput of both networks.

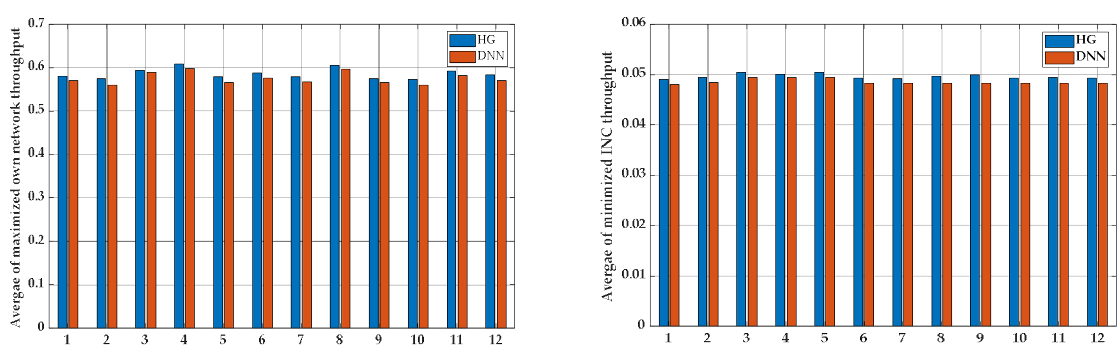

Figure 7. Performance between DNN and HG in terms of throughput

6. Conclusion

In the paper, we proposed a dual-stage Hungarian-based assignment algorithm that can smartly perform the TSCH-based scheduling of own network without collision from INC. We have also utilized a deep learning-based DNN scheme, which provided similar performance to those of TSCH-based scheduling, and it contributed to reducing the execution time of scheduling. The training data set was generated based on a dual-stage Hungarian-based scheduling scheme which can maximize the throughput of own network and minimize the INC throughput. The DNN was trained by this dataset; it achieved almost 90% accuracy and reduced 80% execution time of the original scheme. Moreover, the deep learning-based DNN scheme showed an efficient performance for the TSCH-based scheduling.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea through the Korean Government Ministry of Science and ICT (MSIT) under Grant NRF-2021R1A2B5B01001721. In addition, these results were supported by "Regional Innovation Strategy (RIS)"through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

[1] The Cicso Company, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2016 – 2021 white paper," Cisco:San Jose, CA, USA, 2017.

[2] R. Natarajan, P. Zand, and M. Nabi, "Analysis of coexistence between IEEE 802.15.4, BLE and IEEE 802.11 in the 2.4 GHz ISM band," *IECON Proc. (Industrial Electron. Conf.*, pp. 6025–6032, 2016, doi: 10.1109/IECON.2016.7793984.

[3] M. Ojo and S. Giordano, "An efficient centralized scheduling algorithm in IEEE 802.15.4e TSCH networks," *2016 IEEE Conf. Stand. Commun. Networking, CSCN 2016*, 2016, doi: 10.1109/CSCN.2016.7785164.

[4] N. Taheri Javan, M. Sabaei, and V. Hakami, "IEEE 802.15.4.e TSCH-Based Scheduling for Throughput Optimization: A Combinatorial Multi-Armed Bandit Approach," *IEEE Sens. J.*, vol. 20, no. 1, pp. 525–537, 2020, doi: 10.1109/JSEN.2019.2941012.

[5] D. De Guglielmo, S. Brienza, and G. Anastasi, "IEEE 802.15.4e: A survey," *Comput. Commun.*, vol. 88, pp. 1–24, 2016, doi: 10.1016/j.comcom.2016.05.004.

[6] S. Kharb and A. Singhrova, "A survey on network formation and scheduling algorithms for time slotted channel hopping in industrial networks," *J. Netw. Comput. Appl.*, vol. 126, no. October 2018, pp. 59–87, 2019, doi: 10.1016/j.jnca.2018.11.004.

[7] L. Alkama and L. Bouallouche-Medjkoune, "IEEE 802.15.4 historical revolution versions: A survey," *Computing*, vol. 103, no. 1, pp. 99–131, 2021, doi: 10.1007/s00607-020-00844-3.

[8] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: Why channel hopping makes sense," *PE-WASUN'09 - Proc. 6th ACM Int. Symp. Perform. Eval. Wirel. Ad-Hoc, Sensor, Ubiquitous Networks*, no. September 2014, pp. 116–123, 2009, doi: 10.1145/1641876.1641898.

[9] S. Hammoudi, S. Harous, and Z. Aliouat, "External Interference Free Channel Access Strategy Dedicated to TSCH," *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2018-May, pp. 350–355, 2018, doi: 10.1109/EIT.2018.8500259.

[10] L. P. Sachs, "Performance evaluation.," *NLN Publ.*, vol. 20, no. 17–1807, pp. 61–64, 1980.

[11] D. D. G. authorGiuseppe A. A. Seghetti, "From IEEE 802.15.4 to IEEE 802.15.4e: A Step Towards the Internet of Things," *Adv. onto Internet Things*, pp. 135–152, 2014.

[12] J. Ansari and P. Mähönen, "Channel selection in spectrum agile and cognitive MAC protocols for wireless sensor networks," *MobiWac'10 - Proc. 8th ACM Int. Symp. Mobil. Manag. Wirel. Access, Co-located with MSWiM'10*, pp. 83–90, 2010, doi: 10.1145/1868497.1868511.

[13] R. Mennes, M. Claeys, F. A. P. De Figueiredo, I. Jabandzic, I. Moerman, and S. Latre, "Deep Learning-Based Spectrum Prediction Collision Avoidance for Hybrid Wireless Environments," *IEEE Access*, vol. 7, pp. 45818–45830, 2019, doi: 10.1109/ACCESS.2019.2909398.

[14] R. Mennes, F. A. P. De Figueiredo, and S. Latré, "Multi-Agent Deep Learning for Multi-Channel Access in Slotted Wireless Networks," *IEEE Access*, vol. 8, no. ii, pp. 95032–95045, 2020, doi: 10.1109/ACCESS.2020.2995456.

[15] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-Learning-Based Wireless Resource Allocation with Application to Vehicular Networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, 2020, doi: 10.1109/JPROC.2019.2957798.

[16] I. Hameed, P. V. Tuan, and I. Koo, "Exploiting a deep neural network for efficient transmit power minimization in a wireless powered communication network," *Appl. Sci.*, vol. 10, no. 13, 2020, doi: 10.3390/app10134622.

[17] I. Hameed and P. V. and I. K. Tuan, "Deep Learning – Based Energy Beamforming With Transmit Power Control in Wireless Powered Communication Networks," *IEEE Access*, pp. 142795–142803, 2021.

[18] Q. Tong, X. Zou, and H. Tong, "A RFID authentication protocol based on infinite dimension pseudo random number generator," *Proc. 2009 Int. Jt. Conf. Comput. Sci. Optim. CSO 2009*, vol. 1, pp. 292–294, 2009, doi: 10.1109/CSO.2009.436.

[19] J. Ma *et al.*, "Social account linking via weighted bipartite graph matching," *Int. J. Commun. Syst.*, vol. 31, no. 7, pp. 1–14, 2018, doi: 10.1002/dac.3471.