Reinforcement learning TDMA-based MAC scheduling in the Industrial Internet of Things: A survey

Mehdi Kherbache, Otabek Sobirov, Moufida Maimour, Eric Rondeau, Abderrezak Benyahia

Reinforcement Learning TDMA-Based MAC Scheduling in the Industrial Internet of Things: A Survey

Mehdi Kherbache * **Otabek Sobirov** * **Moufida Maimour** *
Eric Rondeau * **Abderrezak Benyahia** **

* *CRAN Laboratory, Université de Lorraine, CNRS, UMR 7039, Campus Sciences, BP 70239, F-54000 Nancy, France (mehdi.kherbache@univ-lorraine.fr, otabek.sobirov8@etu.univ-lorraine.fr, moufida.maimour@univ-lorraine.fr, eric.rondeau@univ-lorraine.fr).*
** *LASTIC laboratory, University of Batna 2, Batna, Algeria (a.benyahia@univ-batna2.dz)*

Abstract: As technology progresses, almost all Industrial Internet of Things applications are becoming more data-driven. Hence, more advanced ways of communication are required to both save energy and become efficient to collect huge amounts of data. Researchers have now agreed that the root improvement of IIoT communication can start from a lower layer which is Data Link. Time Division Medium Access-based protocols on this layer have drawn academia's attention to decrease the energy consumption of IIoT devices and achieve higher levels of throughput. Yet, a problem arises, that is, developing a scheduler has always been troublesome while designing a Medium Access Control protocol. Scheduling communication in TDMA-based MAC protocols is considered an NP-hard problem. It is one of the topics highly debated among researchers to make the scheduling adaptive to the huge variation of IIoT application requirements. Conventional scheduling algorithms may suffer from the rapid changes in sensor network topology, network throughput, and applications' delay targets. Therefore, Reinforcement Learning algorithms are being integrated into scheduling techniques because of their adaptability and efficiency. This paper surveys RL-based TDMA MAC protocols and compares them in terms of several unified features. Each protocol is explained and discussed with others regarding its advantages and drawbacks.

Keywords: Reinforcement Learning, Communication Scheduling, Wireless Sensor Networks, TDMA, IIoT.

1. INTRODUCTION

Industrial Internet of Things (IIoT) is mainly comprised of sensor devices sharing a common wireless medium whose optimal use is critical in terms of energy, throughput, delay, etc. Time Division Multiple Access (TDMA)-based medium sharing protocols have been considerably studied to meet the IIoT application requirements at the MAC layer as stated by Teles Hermeto et al. (2017). TDMA is one of the most prominent medium-sharing strategies that provide delay-deterministic communication. In TDMA, time is divided into small units called frames which will be further divided into slots. A slot is a base entity for communication between two devices. Scheduling timeslots among nodes should be efficient and optimal so that nodes can save energy and achieve the targeted throughput. TDMA-based communication requires nodes to have some form of duty-cycle management mechanism that periodically determines the sleeping and active timeslots to save nodes' energy, Ergen and Varaiya (2010).

Wireless Sensor Networks (WSN) tend to be multi-hop (ad-hoc) networks where a node can send its data to another node to reach the sink node, which makes energy-saving the most decisive factor while choosing a scheduling algorithm. However, IIoT application requirements necessitate high levels of throughput and low latency, and as invoked by Zibakalam (2012), this is where TDMA-based scheduling algorithms play an important role due to their inherent characteristics of providing intended delay and throughput. The preceding TDMA-based scheduling algorithms generally utilize a single channel. However, with the recent advancements, most protocols integrated the channel hopping technique, which resulted in the compatible usage of TDMA with FDMA, such as Time Slotted Channel Hopping (TSCH) standard. TSCH, standardized in 2015, is defined in Watteyne et al. (2015) as a medium-sharing MAC protocol that incorporates time slotting and frequency hopping as the foundations. However, TSCH standard did not specify any scheduling technique or a way to build it, which allows researchers to create custom application-specific schedulers.

Scheduling communication in TDMA-based MAC protocols is considered an NP-hard problem, according to Saifullah et al. (2010). It is one of the topics highly debated

among researchers to make the scheduling adaptive to the huge variation of IIoT application requirements. The rise in the number of IIoT devices has shown that conventional scheduling algorithms may suffer from the rapid changes in sensor network topology, network throughput, and applications' delay targets. Therefore, researchers started to integrate Reinforcement Learning (RL) algorithms into scheduling techniques because of their adaptability and efficiency. RL algorithms applied in schedulers usually gather information from nodes' medium usage and application performance, and then by processing it, they find the optimal time slot and/or channel offset schedules for communication.

Several surveys tried to collect and analyze TDMA-based schedulers. For example, Hammoudi et al. (2020) covered some RL-based schedulers for TSCH without being exclusively reserved to that. A more recent survey in Urke et al. (2022) covers more TSCH scheduling techniques and classifies them into five categories : centralized, collaborative, autonomous, hybrid, and static. However, it does not include machine learning-based schedulers. This paper surveys RL-based scheduling techniques for TDMA-based MAC protocols in the IIoT. To the best of our knowledge, this is the first survey dedicated to RL-based TDMA schedulers in the IIoT.

We collected TDMA-based RL schedulers to date from the most popular search engines, including Google Scholar, ResearchGate, and IEEEXplore library with the following keywords : TDMA-based, MAC protocol, RL, IIoT, WSN, Scheduling. The initial number of papers was relatively big (for instance, google scholar showed 42), but it was narrowed down to 9 due to the general requirements of this survey. This survey focuses on general TDMA-based MAC layer (single or multi-channel) that make use of RL that can be deployed in the IIoT for different types of applications.

This survey paper classifies RL-based schedulers into two groups: single-channel and multi-channel TDMA-based scheduling. And, it has the following structure. First, a general background on RL is provided in Section 2. Section 3 describes each paper individually and summarizes the advantages and disadvantages of a protocol in question. After that, a table is given to compare the chosen protocols based on several features. Finally, Section 4 concludes the survey.

2. REINFORCEMENT LEARNING

Reinforcement learning is defined in Sutton and Barto (2018) as a machine learning technique based on learning from interactions with the environment. This learning process is goal-directed where an agent must discover which actions yield the most reward by trying them. The interaction between a learning agent and its environment is defined in terms of states, actions and rewards using the formal framework of Markov Decision Processes. An agent in state s takes an action a that takes it to another state s' and then receives reward r from its environment. This learning process is repeated enough times for the algorithm to converge and results in an optimal policy π^*. This latter maps the states to the optimal actions to ensure that the design goal of the algorithm is achieved.

RL methods can be classified in three categories according to Heidrich-Meisner et al. (2007) :

- *Critic-only methods or learning based on value functions* : their idea is to first find the optimal value function and then derive an optimal policy.
- *Actor-only methods* : here the optimal policy is directly searched in the policy space.
- *Actor-critic methods* : which are a combination of the above approaches, the policy (actor) and the value functions (critic) are represented and improved separately. The critic measures the performance of the actor and decides when it should be improved.

The most known critic-only method is Q-learning which is a model-free reinforcement learning algorithm, detailed in Watkins and Dayan (1992). A QL agent takes an action to maximize the value function, $Q(s, a)$, of the state-action pair. $Q(s, a)$ represents the expected total discounted returns from the action taken in state s :

$$Q(s, a) \leftarrow \alpha\ Q(s, a) + (1 - \alpha)(r + \gamma\ max_a\ Q(s', a))$$

where α is the learning rate, γ is the discounting factor that sets the preference either towards immediate rewards or to long-term rewards, r is the received reward from executing action a in state s that leads to state s'. Further, $max_a Q(s', a)$ represents the largest Q-value that can be obtained from the actions that can be taken in the next state.

RL can be combined with deep learning to deal with the problem of high-dimensional state and action spaces, resulting in Deep Reinforcement Learning as stated in Arulkumaran et al. (2017). It is generally based on training Deep Neural Networks (DNN) to approximate the optimal policy and/or the optimal value functions. Furthermore, DNNs could be used to model the environment in which RL agents will be trained.

3. RL-BASED MAC SCHEDULERS

In this section we will detail the RL-based TDMA MAC schedulers by providing a description in terms of objectives, how RL is used in the proposed method and the performance results for each. Table 1 summarizes the discussed techniques in a comparative way, provides details of the experimentations done for each and also their advantages and drawbacks.

3.1 Single-channel

Liu and Elhanany (2006) introduced RL-MAC as a reinforcement learning based MAC protocol for WSNs. It was designed to optimize the nodes' radio on-off function in order to minimize energy consumed by the sensor nodes. The main particularity of the proposed algorithm is that it adapts to the traffic generation pattern of the node and also of its neighboring nodes. The proposed learning scheme uses Q-learning with a reward function that tries to find a trade-off between minimizing energy consumption and ensuring an acceptable throughput. Precisely, the state s is represented by the number of packets queued for transmition at the beginning of the slotframe. The action is the reserved active time for the node. The reward function

is a combination of two components, the first one reflects on the internal state of a node at timeslot t which aims to maximise the energy efficiency. The second component reflects on the state of other nodes as perceived at timeslot $t+1$ which tries to minimize the number of missed packets. An ϵ-greedy method is followed in the learning algorithm to ensure a balance between exploitation and exploration. Performance evaluation of RL-MAC shows that it outperforms in terms of energy efficiency and data throughput S-MAC proposed in Ye et al. (2004), and T-MAC proposed in van Dam and Langendoen (2003), two MAC protocols designed to reduce energy waste caused by idle listening by managing the nodes' duty cycle. Also, latency is reduced considerably in RL-MAC compared to S-MAC especially when traffic load is heavy. Finally, we can say that despite the fact that the proposed protocol is one of the first works including RL for scheduling nodes' radio on-off, it is outdated (2006) and does not fit the strict requirements of nowadays IIoT applications.

Always in the context of energy-saving directed protocols, Mihaylov et al. (2011b) proposed a self-organizing reinforcement learning approach for scheduling the wake-up cycles of nodes in a wireless sensor network. It allows to adapt the use of sensor resources to the applications requirements in terms of latency, data rate and lifetime. Concretely, each node will learn to stay awake during the periods where it needs to communicate with its parents/children nodes (nodes that belong to the same *coalition*), this behaviour is called synchronization. At the same time, the node learns to stay asleep when neighboring nodes on the same hop are communicating (nodes in another coalition). In other words, the node desynchronizes with the neighboring nodes that are not in its coalition to avoid radio interferences and packet loss. The algorithm uses a value iteration approach similar to Q-learning with an implicit exploration strategy. The action space contains the timeslot numbers within the slotframe, an agent selects a slot when its radio will be switched on for the duration of the duty cycle which is fixed by the user. Each agent stores a "quality value" for each timeslot which is updated every time an event (overheard, sent or received packets, idle listening) occurs during that slot. The node will stay awake for those consecutive timeslots (of a length equal to the duty cycle) that have the highest sum of Q-values. Evaluating this protocol in different topologies has showed that it provides much lower end-to-end latency compared to S-MAC. However, various shortcomings can be addressed in the proposed protocol. For example, the duty cycle is fixed and equal for all nodes in the network which means that it is not traffic-adaptive. In addition, communications between active nodes on the same routing branch can collide since they are synchronized. These drawbacks has been solved in a subsequent extension of the algorithm called DESYDE, proposed in Mihaylov et al. (2011a).

Savaglio et al. (2019) proposed a Q-learning MAC protocol (QL-MAC) which self-adjusts the node's duty-cycle to minimize energy consumption without impacting the other network parameters. It optimizes the sleeping and active periods of the nodes based on traffic predictions and transmission state of neighboring nodes. This is ensured by applying a Q-learning scheme where each slot is assigned a Q-value that is updated based on the actions of a node or the state of neighbor nodes. A node decides whether it should stay active or in sleep mode during each time slot, so the action space depends on the number of timeslot in a frame. The reward function is crafted carefully to take into consideration the state of neighboring nodes in addition to node state. Performance results show that QL-MAC reduces considerably energy expenditure with a minimal negative impact on the PDR compared to CSMA/CA. However, the authors did not provide insights on the impact of the learning algorithm on latency which allows to say that there is no guaranties of low latency.

The previous works are protocols developed from scratch and do not rely on a standardized stable protocol, such as TSCH. Based on TSCH MAC protocol, Nguyen-Duy et al. (2019) presented RL-TSCH, a reinforcement learning solution for scheduling TSCH nodes. The algorithm schedules the process of turn on/off node's radio at each beginning of timeslot based on the current state and the previous state of the node. It takes into consideration the number of packets in the transmission buffer along with the remaining energy in every node. The learning agent determines the number of active and non-active timeslots at the beginning of each slotframe. Thus, the node behaves as in MSA (Minimal Scheduling Algorithm) during the active timeslots and turns off its radio during the non-active timeslots. The algorithm applies Q-learning with an action space consisting in choosing the number of active timeslots. The reward function is designed to minimize energy consumption and ensure a high throughput and reliability. Evaluating the proposed algorithm in different small-scale topologies showed that it reduces the energy consumed to about one third compared to MSA, achieves a similar PDR but the latency is much higher to that obtained with MSA. Based on the obtained results, it is logical to say that the proposed algorithm does not fit for low-latency applications but may be well suited for applications with strict energy requirements.

Another work based on TSCH protocol is the one presented by Park et al. (2020). The authors introduced a multi-agent reinforcement learning based scheduler for TSCH, called QL-TSCH. The proposed algorithm reduces collisions while allowing contention, so multiple links can be scheduled in the same timeslot. This has the effect of increasing network throughput while allowing low energy consumption, thus the algorithm fits for high-density and high-traffic applications. In fact, each node acts as a Q-learning agent that learns the transmission slot with the lowest transmission failure rate and transmits only in that slot. During the learning phase, an agent performs a transmission with a probability $P_{exploration}$ in a slot chosen by an action peeking mechanism that selects the least active timeslot. Otherwise, the action (the timeslot) with the highest Q-value is selected. The chosen timeslot is scheduled as a transmission slot and the remaining slots are scheduled as listening slots. Rewards are assigned based on the success/failure of the performed action, a positive value when a transmission succeeds and a negative value is assigned if it fails. This algorithm addresses the non-stationarity problem of the multi-agent system that may interrupt the convergence by including an action peeking mechanism. This latter consists in a node observing the activity of other neighboring nodes during its listening

slots, the node concludes that a timeslot is already reserved whenever a nearby communication is detected during that slot. Performance evaluation of QL-TSCH has been done in a large-scale network of 99 nodes by considering the PDR and the end-to-end packet delay in three industrial scenarios, compared with Orchestra scheduler proposed in Duquennoy et al. (2015) and FTA scheduler proposed in Park et al. (2019). Results show that QL-TSCH outperforms both Orchestra and FTA in terms of PDR, a better end-to-end packet delay compared to Orchestra but FTA has the best performance regarding this latter metric. In spite of all the good results that QL-TSCH achieved, its energy expenditure has not been evaluated. This leaves an open future research perspective.

3.2 Multi-channel

Phung et al. (2013) proposed a multichannel protocol for data gathering WSNs with a reinforcement learning based scheduling algorithm. The aim of the algorithm is to minimize energy consumption caused by collision, idle listening and deafness problem in WSNs. It addresses the joint problem of route selection and transmission scheduling and solves it in a fully distributed manner without a need for a coordination between the nodes. In other words, it makes nodes learn not only to which parent but also on which channel they should forward their data. Concretely, in each slot a node executes an action from the set of available actions (listen on its own channel for reception or transmit to one of its parents default channels) and keeps track of the probability of successfully performing each action in that slot. Such a probability is updated for the selected action in a given slot and will be the basis for choosing the best actions in the scheduling phase. The trade-off between exploitation and exploration is ensured by applying a "*win-stay lose-shift*" policy. In fact, a successful action will be repeated in the same slot in the next frame while a failed action leads to choose randomly another action from the action space in the next frame. The algorithm is traffic-adaptive because a node only contends for channel access when it has packets in its queue. Evaluation results of the proposed protocol show that it outperforms a frequency-hopping protocol called McMAC, proposed in Hoi-Sheung et al. (2007), in terms of PDR, end-to-end latency and it provides 9 times better energy efficiency.

Phung et al. (2018) introduced a scheduler for TSCH networks supporting multiple QoS (Quality of Service) objectives. It is based on a trial-and-error process where each node acts as an autonomous agent learning from feedback from its environment and the interaction with the other nodes. In fact, two RPL (Routing Protocol for Low-Power and Lossy Networks) instances are considered in the design of the learning scheme, one is reserved for delay-sensitive data and the other one for regular data. A node either listens for data coming from the children nodes, transmits data to the parent of delay-sensitive instance or transmits to the parent of regular instance. The reward function is a combination of rewards for the objective of reliability and energy and rewards for the objective of low latency. Based on the received reward signal, the action to execute in the next slotframe would be the same as in the current one if the reward equals 1 or another action is selected if the reward equals 0. The learning algorithm keeps track of the probabilities of successfully performing each action in each slot, the same way as the precedently described algorithm. The probabilities are updated for each executed action and the allocation process exploits the final obtained probabilities to choose the best actions for each node during the slotframe. The scheduler has been evaluated with two RPL instances where one is for delay-sensitive data (bounded latency setup at 2 timeslots) and the other is for regular data. Results show that the proposed scheduler provides much lower data delivery latency than Orchestra (one order of magnitude lower) while keeping a similar energy consumption. This proves that the scheduler fits for delay-sensitive applications requiring low latency.

In order to consider the more general problem of satisfying packets' deadlines in delay-sensitive environments, Chilukuri et al. (2021) proposed RLSchedule as a reinforcement learning based TDMA slot scheduler for networks with strict time constraints. It has the objective of finding a schedule where the least possible number of packets miss the deadlines by the least amount of time. The paper identifies a set of node features that will be the basis for network state representation. This enables the RL scheme to take scheduling decisions based on an up-to-date dynamic network status and not on the same static criterion every time. The proposed framework follows a centralized approach where a controller gathers information about the most frequently seen network scenarios and sends this knowledge to a server. Deep Reinforcement Learning with PPO (Proximal Policy Optimization) is applied to learn the optimal policy which will be sent to the centralized controller. This latter exploits the received policy to build schedules for any scenario it sees. The action space consists in choosing the first M (the set of available channels) non-conflicting transmissions based on a proven baseline heuristic. The used heuristics are the following five : a) Deadline Monotonic (DM), b) Earliest Deadline First (EDF), c) Proportional Deadline (PD), d) Earliest Proportional Deadline (EPD) and e) Least Laxity First (LLF). A sixth possible action consists in choosing the top M non-conflicting transmissions with the minimum set of features. The reward function is designed in such a way to minimize the number of packets missing their deadlines and even if they exist the time by which they miss their deadlines is minimized. Performance evaluation of RLSchedule shows that it provides much lower packet delay and it has the least percentage of missed packets compared to the other scheduling heuristics (DM, EDF, PD, EPD, LLF). The obtained results confirm that RLSchedule is well suited for time-constrained networks.

Chilukuri and Pesch (2021) presented RECCE, a scheduler that follows the same principles of RLSchedule but considers a more general problem by including routing in the learning scheme. In fact, RLSchedule considers the best (shortest) routes following Dijsktra algorithm while RECCE explores and learns multiple routes and schedules to deliver more packets within the deadline. Experimentation results showed that by exploring different routing paths and choosing the one (not necessarily the shortest) with the minimum delay allows RECCE to meet the scheduling goal better than RLSchedule.

Table 1. General Comparison of RL-based Schedulers

Scheduler	Main Objective	Class	RL method	Simulation	Testbed	Compared with	Advantages	Limitations
Liu and Elhanany (2006)	Energy saving	Single-channel	Q-Learning	NS-2 simulation	No	S-MAC and T-MAC (star with 5 nodes, linear with 10 nodes, mesh topologies with 100 nodes)	increased throughput and energy saving.	- synchronization problems - no real test-bed experiment - old proposal and compared with outdated protocols.
Mihaylov et al. (2011b)	Low latency	Single-channel	Q-Learning	OMNet++ simulation	No	S-MAC (Topology: Line with 4 nodes, Single-hop mesh with 6 nodes, Grid with 4x4 nodes)	- low end-to-end latency. - reduced energy consumption	- not well suited for irregular data traffic. - fixed duty cycle for all nodes - coordination problem.
Savaglio et al. (2019)	Energy saving	Single-channel	Q-Learning	Cooja simulation	Implemented on TelosB motes with TinyOS	CSMA/CA (100% and 60% duty cycles), Topology: mesh with 7 nodes, grid with 4x4, 7x7, 10x10 nodes	- adaptable to topology changes . - reduced energy consumption	- no insurance of low latency - tested against a basic protocol (CSMA/CA)
Nguyen-Duy et al. (2019)	Energy saving	Single-channel	Q-Learning	Cooja simulation (Zoletia Z1 sensor nodes)	Nodes running Contiki OS	TSCH Minimal Scheduling Function (In simulation : 4-8 nodes in star, linear and mixed topologies. In real test-bed : 3 nodes)	- reduced energy consumption - traffic-adaptive	- no extensive experimentations - not suited for low latency applications. - benchmarked against the most basic TSCH scheduling function (MSA)
Park et al. (2020)	High throughput	Single-channel	Q-Learning	No	ARM Cortex M4 MCU-based IoT platform	Orchestra and FTA scheduler (Mesh topology with 99 nodes)	- fit for high-traffic, high-density large scale networks. - low latency and collision rate.	- no garanties of recuced energy consumption.
Phung et al. (2013)	Energy saving	Multi-channel	General RL	Matlab simulation	No	Mc-MAC (grid and random topologies of 25 nodes)	-traffic-adaptive - low energy consumption	- synchronization problems. - experimentations done in a matlab simulation.
Phung et al. (2018)	Low latency	Multi-channel	Trial-and-error process	Matlab simulation	No	Orchestra (grid topology of 16 nodes)	- traffic prioritization included - low data delivery latency - fit for delay-sensitive applications	- support of only two types of traffic (delay-sensitive and regular). - no garanties of reduced energy consumption. - experimentations done in a matlab simulation.
Chilukuri et al. (2021)	Satisfying packets deadlines requirements	Multi-channel	Deep RL - Proximal Policy Optimization (PPO)	Home-made simulator	No	Scheduling heuristics (DM, EDF, PD, EPO, LLF,CFLLF, Random)	- almost garanteed satisfaction of deadline requirements. - suitable for time-constrained applications - scheduling decisions based on different preselected nodes features - adaptability to topology changes.	- no real test-bed experimentations - home-made custom simulator - need for a centralized controller
Chilukuri and Pesch (2021)	Satisfying packets deadlines requirements	Multi-channel	Deep RL - Proximal Policy Optimization (PPO)	Home-made simulator	No	Scheduling heuristics (DM, EDF, PD, EPO, LLF,CFLLF, Random)	- routing included in the learning scheme → increased performance relatively to Chilukuri et al. (2021)	similar to Chilukuri et al. (2021)

4. CONCLUSION

In this paper we surveyed the most prominent RL-based TDMA MAC schedulers for the IIoT. We provided a summary table that can be the basis for comparing the different surveyed schedulers, enumerating their advantages and drawbacks. From this table, we can notice that 44% of the papers focus mainly on energy objectives while the rest are more *QoS* directed. But there are no papers that consider both of the issues at the same time. In addition, Quality of Experience (*QoE*) is not really exploited in these works and applications' needs should be better considered. Moreover, 44% of papers use network simulation tools for performance evaluation and 44% of papers use either matlab simulations or custom simulators. Further, only 33% of the papers evaluated their works on a real test-bed with real sensors. This means that the results truthfulness can be discussed regarding two angles : a) non network simulation tools do not integrate all the complexity of the WSN protocols and their environment, b) real platforms should be required for a complete validation.

As future work, it is intended to extend the survey to cover machine-learning based schedulers in general and not only RL-based ones as in this paper.

REFERENCES

Arulkumaran, K., Deisenroth, M., Brundage, M., and Bharath, A. (2017). A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine*, 34. doi:10.1109/MSP.2017.2743240.

Chilukuri, S. and Pesch, D. (2021). Recce: Deep reinforcement learning for joint routing and scheduling in time-constrained wireless networks. *IEEE Access*, 9, 132053–132063. doi: 10.1109/ACCESS.2021.3114967.

Chilukuri, S., Piao, G., Lugones, D., and Pesch, D. (2021). Deadline-aware tdma scheduling for multihop networks using reinforcement learning. In *2021 IFIP Networking Conference (IFIP Networking)*, 1–9. doi:10.23919/IFIPNetworking52078.2021.9472801.

Duquennoy, S., Al Nahas, B., Landsiedel, O., and Watteyne, T. (2015). Orchestra: Robust mesh networks through autonomously scheduled tsch. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, 337–350. Association for Computing Machinery, New York, NY, USA. doi:10.1145/2809695.2809714. URL `https://doi.org/10.1145/2809695.2809714`.

Ergen, S.C. and Varaiya, P. (2010). TDMA scheduling algorithms for wireless sensor networks. *Wireless Networks*, 16(4), 985–997. doi:10.1007/s11276-009-0183-0.

Hammoudi, S., Bentaleb, A., Harous, S., and Aliouat, Z. (2020). Scheduling in ieee 802.15.4e time slotted channel hopping: A survey. 0331–0336. doi:10.1109/UEMCON51285.2020.9298043.

Heidrich-Meisner, V., Lauer, M., Igel, C., and Riedmiller, M. (2007). Reinforcement learning in a nutshell. 277–288.

Hoi-Sheung, So, W., Walrand, J., and Mo, J. (2007). Mcmac: A parallel rendezvous multi-channel mac protocol. In *2007 IEEE Wireless Communications and Networking Conference*, 334–339. doi:10.1109/WCNC.2007.67.

Liu, Z. and Elhanany, I. (2006). Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks. In *2006 IEEE International Conference on Networking, Sensing and Control*, 768–773. doi:10.1109/ICNSC.2006.1673243.

Mihaylov, M., Le Borgne, Y.A., Tuyls, K., and Nowé, A. (2011a). Distributed cooperation in wireless sensor networks. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, 249–256. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC.

Mihaylov, M., Le Borgne, Y.A., Tuyls, K., and Nowe, A. (2011b). Self-organizing synchronicity and desynchronicity using reinforcement learning. In J. Filipe and A. Fred (eds.), *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*, volume 2 of *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*, 94–103. Joaquim Filipe and Ana Fred.

Nguyen-Duy, H., Ngo-Quynh, T., KOJIMA, F., Pham-Van, T., Nguyen-Duc, T., and Luongoudon, S. (2019). Rl-tsch: A reinforcement learning algorithm for radio scheduling in tsch 802.15.4e. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 227–231. doi: 10.1109/ICTC46691.2019.8939833.

Park, H., Kim, H., Kim, K.T., Kim, S.T., and Mah, P. (2019). Frame-type-aware static time slotted channel hopping scheduling scheme for large-scale smart metering networks. *IEEE Access*, 7, 2200–2209. doi:10.1109/ACCESS.2018.2886375.

Park, H., Kim, H., Kim, S.T., and Mah, P. (2020). Multi-agent reinforcement-learning-based time-slotted channel hopping medium access control scheduling scheme. *IEEE Access*, 8, 139727–139736. doi:10.1109/ACCESS.2020.3010575.

Phung, K.H., Huong, T.T., Khanh Dung, D., Tuong, V.X., Pham, T., Nguyen, T., and Steenhaut, K. (2018). A scheduler for time slotted channel hopping networks supporting qos differentiated services. In *2018 International Conference on Advanced Technologies for Communications (ATC)*, 232–236. doi: 10.1109/ATC.2018.8587569.

Phung, K.H., Lemmens, B., Mihaylov, M., Tran, L., and Steenhaut, K. (2013). Adaptive learning based scheduling in multichannel protocol for energy-efficient data-gathering wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2013. doi:10.1155/2013/345821.

Saifullah, A., Xu, Y., Lu, C., and Chen, Y. (2010). Real-time scheduling for wirelesshart networks. In *2010 31st IEEE Real-Time Systems Symposium*, 150–159. doi:10.1109/RTSS.2010.41.

Savaglio, C., Pace, P., Aloi, G., Liotta, A., and Fortino, G. (2019). Lightweight reinforcement learning for energy efficient communications in wireless sensor networks. *IEEE Access*, 7, 29355–29364. doi:10.1109/ACCESS.2019.2902371.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.

Teles Hermeto, R., Gallais, A., and Theoleyre, F. (2017). Scheduling for ieee802.15.4-tsch and slow channel hopping mac in low power industrial wireless networks: A survey. *Computer Communications*, 114, 84–105. doi: https://doi.org/10.1016/j.comcom.2017.10.004.

Urke, A.R., Kure, , and Øvsthus, K. (2022). A survey of 802.15.4 tsch schedulers for a standardized industrial internet of things. *Sensors*, 22(1). doi:10.3390/s22010015. URL `https://www.mdpi.com/1424-8220/22/1/15`.

van Dam, T. and Langendoen, K. (2003). An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, 171–180. Association for Computing Machinery, New York, NY, USA. doi:10.1145/958491.958512. URL `https://doi.org/10.1145/958491.958512`.

Watkins, C. and Dayan, P. (1992). Technical note: Q-learning. *Machine Learning*, 8, 279–292. doi:10.1007/BF00992698.

Watteyne, T., Palattella, M.R., and Grieco, L.A. (2015). Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. RFC 7554. doi:10.17487/RFC7554. URL `https://www.rfc-editor.org/info/rfc7554`.

Ye, W., Heidemann, J., and Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3), 493–506. doi: 10.1109/TNET.2004.828953.

Zibakalam, V. (2012). A New TDMA Scheduling Algorithm for Data Collection over Tree-Based Routing in Wireless Sensor Networks. *ISRN Sensor Networks*, 2012, 1–7. doi:10.5402/2012/864694.